Thank you for the introduction. I'd like to talk to you about My Years at NASA.

Betty Jo Armstead, January 21, 2015

At the time I didn't realize there was anything exceptional about my story. I grew up on a farm in northeastern Missouri, and I enjoyed my experiences as a student. I had an outstanding math teacher in high school that really challenged me, and I loved it. Once I graduated, I only applied to one college because I could stay with my aunt and uncle in Illinois.

I graduated from Eastern Illinois Teachers College in 1953 with a major in chemistry and a minor in mathematics—I was the only woman in that program at the time. Going into college I actually preferred mathematics, but I didn't see any jobs in mathematics other than teaching, which I did not want to do. I applied to Iowa State in Ames for graduate school, and worked half time as a teaching assistant in the Chemistry department. Most of the students were locals right off the farm. Because I was the only teaching assistant from the Midwest and much more understandable to the local students, many students sought me out. My tutoring classes ended up with about 20 students each rather than the usual two or three. The excessive tutoring load was taking too much of my time. Also, I was not happy with the Chemistry Professor who seemed to be teaching Chemistry from the 18th century, which was such a contrast to the terrific professors I had at Eastern Illinois. So I dropped out of the program.

Now what to do? I picked up the general Federal Government application forms and applied for a job. Because of my years of studying math, I was offered a position at NACA, the National Advisory Committee for Aeronautics located in Cleveland Ohio. NACA had a number of jet engine test cells, as well as two supersonic wind tunnels. The power required by the tunnels was so great that when they ran them, they had to notify the Tennessee Valley Authority or TVA.  One time there was an emergency shutdown of one of the wind tunnels, and the turbines at TVA actually began to turn backwards.  These facilities were used in the design and testing of jet engines.  Two of the main issues being dealt with were trying to increase engine efficiency and solving the problem of birds and other objects flying into the engines. The local residents were upset when they found out that some of the engine tests involved dead chickens being thrown into a running engine.

When I walked into the computer area where I was assigned at NACA, what I saw was a room full of women seated at either a Frieden or Monroe calculator.  These were fancy adding machines that could add, subtract, multiply, divide and take the square root.  Next to each machine were two pieces of paper (an input page and an output page).  The input page contained engine test data such as temperatures, pressures, etc.    The data was recorded during tests by taking pictures of tubes that looked like thermometers.  The women read the data from the negatives of these pictures.  My job was to provide the instruction for processing the data. This meant working with the test engineers to provide these instructions.  Based on their instructions, I would create appropriate column headings on the input and output sheets. The end results would tell the engineers how well it worked.

Because I was working with the engineers, I soon found more interesting areas that could use my talents.  The engineers were generally lacking in math and computer skills, so anyone knowledgeable and willing to work was welcome, male or female.  NACA had acquired an IBM 650 computer, which I volunteered along with a few others to learn to program; the small group included my supervisor, Marge Jereb.  The IBM 650 had 60 36-bit words of electronic memory and read its instructions from a rotating drum—compare that to the capacity of today's computers.  I transferred to the Aeronautics division of NACA to write programs for the IBM650, supporting the engineers studying engine designs.

Soon after this, I went to work for Aaron Boxenbom an engineer in the Controls Division, well known for thinking outside the box and someone I really enjoyed working with.  He was interested in calculating possible trajectories to the moon.  I did write a program for the IBM 650 calculating a trajectory to the moon.  Although it was interesting, we did not have a practical use for the program at that time since NACA was primarily designing, building and testing jet engines. NACA did not become involved in the space industry until it became NASA in 1958.

One of the funny stories about that time in my career was that Aaron and I had written a "Technical Note" report, proving that rotating the thrust of the nozzle of a rocket engine perpendicular to the path of the rocket would effectively stop the forward momentum. When we sent the report to NACA headquarters, they rejected the report, saying they didn't understand it.  It was far more mathematical than what they were accustomed to so it got shelved.  I believe Aaron was busy with something else and just let it drop, which was very much his style.

Another project I really enjoyed was working on Monte Carlo programs.  This was a technique of generating and using random numbers.  I wrote a "Monte Carlo" program for NACA's nuclear research division to determine the shield thickness for a nuclear engine.   In this case the random numbers generated were utilized to map out probable particle pathways, an approximation that served to determine the thickness of the shield design.  I had to find some experts (outside of NACA) that knew something about Monte Carlo methods of computing. It was a fun and challenging project, and I had to make calls all over the U.S. to find people who could guide my work with the Monte Carlo programs—it was a real learning process. However, NACA lost interest in nuclear engines for space flight at that time.

I finally transferred back to the computer division as it became more prestigious. Instead of a room full of women working on calculators, the computers were electronic. At that time, our primary computer was the IBM 650. The users, who were NACA engineers and scientists, were allowed to place their card decks in the card reader and wait for their output to be printed.

The demand for computing became such that we were running the IBM 650 twenty-four hours a day. This meant we needed operators especially for the second and third shift. The jobs were run on a first come first run basis. It also meant we needed more computing power, so we decided to test an IBM 704 as our computing needs were expanding.

My supervisor, Marge Jereb and I had gone to Washington DC to test some of our jobs on the 704. We had been up all night getting all of our test jobs to run. By then we were very tired and running around in our stocking feet. The 704 was sitting in a large picture window that looked out on Pennsylvania Avenue in Washington DC. We both looked like we hadn’t slept for two days. That morning, a group of people had gathered and was looking in the window at us, but it wasn’t because of how we looked. The date was October 4, 1957, and that was the morning Sputnik went up. They came and took the computer from us in order to track the Sputnik since it was one of only a few computers capable of the task. Needless to say we got no more testing that day. Within a very short time NACA became NASA. At that time there were only three centers—NASA Langley Field Virginia, NASA Ames in California, and NASA Lewis in Cleveland where it was, later renamed NASA Glenn after the astronaut John Glenn.

We did acquire the IBM 704, which required a raised floor and **lots** of air conditioning—buildings at that time were not constructed to account for the noise, heat, and electricity demand of large computers. The new computer ran **very hot**. The NASA engineers and scientists who were the users were not happy at first, because they no longer had direct access to the computer. They had to leave their card decks for the operators to feed into the computer and wait for the output to be delivered back to them. Initially their jobs were read directly into the computer, and the output went to an attached printer and or a card reader punch. and it was first come first served. Eventually when the backlog got larger, we would copy their jobs to tape, which we hoped to complete over night. We had enough backlog of jobs so that a user might have to wait several days to get their results back.

Up until this time, we had a pretty simple system of scheduling jobs. Jobs requiring more than 30 minutes were run after hours, and jobs under 30 minutes were scheduled first come, first served. Also, we had not limited or charged for the amount of printed output. After seeing lots of engineers pick up a **full box** of paper, look at the last page, and throw the rest out, we decided we had to do something. The first thing we tried was charging for paper. Since this was really a fictional charge it didn't work. Then it became clear we needed a more sophisticated method of scheduling jobs. We began to require users to estimate run times and the number of pages of output. Since jobs with run times longer than 30 minutes and/or fifty pages were run at night, many people purposely underestimated run times and number of pages of output. It soon became that in the daytime these estimates had to be strictly enforced, and jobs exceeding their page and time estimates were simply terminated. Users quickly learned to be more precise with their estimates!

Around 1964 we acquired the IBM 7094/7044 Direct Couple System, which was a hybrid platform with a shared memory connection between two existing systems.  It allowed newer* "human" interface devices, at the time that was the card reader/punch and the high-speed impact printer, to be connected to IBM's primary scientific computer.  Besides the printer and card reader access, this union also made another 32K words (36 bit) available for input/output buffering and job spooling support, and **doubled** the amount of memory normally available on the 7094.   By today's standards, this computer had an unbelievably stingy amount of memory, the equivalent of 393,000 skinny (6 bit) bytes.  Other input/output devices attached to 7044 were a disk the size of a car that provided only 55 Mega Bytes of storage and a terminal interface built for the IBM Selectric typewriter—which didn't print much faster than a good typist.  For the NASA-Glenn scientists and engineers, this provided a computer environment with sophisticated batch job processing and some limited user available disk storage.

So while there was clearly lots of serious work going on, NASA offered a really unique environment, and we had a lot of fun too.  We had a speaker system set that allowed the operators to make announcements to the users waiting outside the air conditioned computer room.  One night my husband Al and I ran a small batch of the jobs waiting to be processed and disposed of the output.   We then recorded a tape that started with "Good morning Linda, surprised I can talk?" We followed this with silly comments on each job.  It took us about four hours to set up the joke.  The next morning, everyone knew something was up, and Al's whole branch was in the room to see the fun. When the operator (Linda) started the jobs, we started the tape recorder. The first words were, "Good morning Linda, surprised I can talk?" She came flying out of the computer and said, "That computer is talking to me!" As luck would have it, Al's boss Dr. Albers was showing some visitors the computer at that time. He asked Al to explain that it was only a tape recorder. Al's answer was, "Dr. Albers, you know its language has been a little spicy lately!" Dr. Albers tried to explain to the visitors that it was only a tape recording. One visitor said, "I know, but how does that computer talk?" I am sure she told people she had heard a talking computer at NASA in the 1960s.

During all of this time I had been an active member of the IBM scientific users group called SHARE, a group that included representatives from all the NASA sites, several universities, and companies like Bell Labs. I also served on the FORTRAN committee; FORTRAN is a programming language developed by IBM in the 50's, so at this time it was still relatively simple. There were two areas that I was interested in – handling errors in FORTRAN runtime and developing more **user-friendly** input/output methods. As a member of SHARE and a NASA employee, I designed and wrote a set of routines called the FORTRAN error package. This package handled such programming errors as division by zero, floating point and fixed point errors that were detected by the hardware. If such errors were ignored, your outcomes would be garbage. The routine I developed would activate when these errors caused an interrupt. I also modified FORTRAN library routines to activate the subroutine for illegal arguments, such as taking the square root of negative numbers. My program provided the user with a back trace to the main program, and the job would terminate upon detection of the first error. The first day the error package was installed, we started with a backlog estimated to take about 48 hours. That backlog only took about two hours to complete, since so many of the programs had errors—about 98%.

Users were not pleased to hear this. The users rebelled, and I was given 15 minutes notice to attend a meeting with my branch chief and a group of about 30 angry users—it was a full room! I explained to the group that these types of errors were giving them wrong answers and must be corrected. Apparently I must have made a good argument, because my Branch Chief said the software would stay. I did agree to allow the **user** to specify the number of errors that were allowed before their program was terminated; of course they soon realized they had to fix all the errors regardless, and terminating the program at the first error was far more efficient. Our office volunteered to provide help in correcting the errors.

It had been helpful that IBM was willing to modify the FORTRAN compiler so that I could develop the back trace. They added markers to the coding so that my program could identify and locate problems. This meant that users knew not just that there was an error, but in which line of code the error appeared. At the next IBM SHARE meeting, I described my FORTRAN Error Package to about 150 users and developers of FORTRAN, and got a standing ovation when I finished. People were enthusiastic about the program, and IBM agreed to incorporate the subroutines into the standard FORTRAN library.

1968 marked the beginning of interactive access to support computer modeling, program development and real-time facility test management with the installation of IBM's Time Sharing System or TSS. TSS ran on the IBM 360 Model 67, one of the first machines built on virtual memory architecture; this meant that the user was able to use a larger address space that was mapped onto disk. This was known as virtual memory. Although we at NASA Glenn worked with TSS for several years, it became apparent that IBM was not going to go down this road. Even though **our** users liked TSS, IBM decided to discontinue it at that time. IBM did support a system called VM 370 which used virtual memory, but it did not support the terminal access like TSS. Many viewed that decision as a misstep by IBM since NASA and others continued to use TSS, and contemporary PCs use the virtual memory concept. The next shift occurred when NASA Glenn purchased a Cray computer that was at least ten times faster than previous computers. Seymour Cray was an extremely interesting individual who believed orderly wiring slowed a computer down. The inside of Cray computers looked like a bird's nest if you opened them up. And no two Cray computers looked alike once they went into use in the field since they were designed to be repaired by Cray support engineers. The Cray software was primarily a FORTRAN compiler and a very **simple** batch operating system. One of my jobs was to provide an interface to the Cray, so I chose a Sun mini-computer, mainly because we had one available and I needed something small to feed the Cray. I wrote the Sun program that fed jobs to the Cray and processed the output. The output could include the binary card deck, and output that was sent to a printer.

From May 1954 until July of 1982, I was a NACA/NASA employee, and after I retired from NASA I worked for a NASA contractor called ANALEX as a system programmer. Robert Perry was someone I had mentored when he started at NASA as a computer operator; over his career he learned enough to be a system programmer. I left ANALEX and joined Robert in a company he started. Our job was to provide technical support for the time-sharing, Cray, and virtual memory systems. We also provided technical input for future hardware and software purchases.

In 1982 I joined Sverdrup Technology as a system analyst. I became a member of the Centralized Mass Storage team, which included both Sverdrup and NASA employees. The archiving software chosen for evaluation was a system called UNITREE.  I worked with a young NASA employee and we designed a wonderful interface to run UNITREE on NASA computers. This system was not only to supply users with a reliable mass storage system where they could safely store information, but also backup user workstations. Although we had a lot of disk space, we actually backed up the system to tape, **and** sent two copies to the NASA site Plumbrook,  which sat on the shore of Lake Erie.  Each week we replaced the oldest 0f the two copies.  I don't think we ever had to use them, but they were there if we ever needed them.  I worked for Sverdrup until I actually retired in the late 90s.

As you may know, engineers hate to throw anything away, but the ever-expanding volume of tapes we had to store got to be a real problem.  We had visions of a much more elaborate backup storage system.  We have that now, in what's known as the cloud.  There's no way to know for sure, but I suspect much of the work we did through the SHARE group was the beginning of the cloud.

And that's my story of my years at NASA, during an era of truly groundbreaking work in computing and technology.  It was the first time computing was being integrated to such a degree for **problem solving** in the sciences.  This represented a real paradigm shift away from previous methods of simple trial and error.  With that I'd like to conclude my talk, and I'd be happy to answer any questions you may have.  Thank you.