Computing Memories

Nancy Welty Clark

The following is derived from my autobiography – *Blessed with Too Many Talents*. There are many omissions and a few additions to help the partial book stand alone.

To Work in Cincinnati

Luckily during my last semester at Marquette University a female math professor got me, Nancy Welty Clark, an interview for a job, so I was ready for life after graduation. I had been worried that I might be stuck teaching, which I didn’t think I would be good at. I had a Major in Mathematics and in Philosophy and was only one credit hour short of a major in Physics (I remedied that later) with minors in Education and Latin. So the Monday following graduation in January 1951, I began working for General Electric Aircraft Gas Turbines in Evendale, Ohio, just north of Cincinnati, with a group of thirteen introverted, compulsive working girls, mostly graduates in mathematics at small colleges.

The Computations Group

Two girls were from Lynn, Massachusetts; two were from Pennsylvania; Ruth was from Wisconsin; Mae from Kentucky; two from Tennessee; and rest were locals. We carpooled to work every day from Cincinnati to Evendale, alternately closing and opening car vents and windows as we passed the Proctor and Gamble plant, the Gilbey’s Gin distillery and an asbestos shingle plant, which left a layer of asbestos dust on the car.

Four of us rented the second floor of a duplex. Four young engineers rented nearby and we played Ping-Pong and went places together–Modern Art Museum, Night Court, Ohio Riverboat trip.

Extra Curricular

During the time I spent in Cincinnati, I tried to improve my non-academic skills by attending Patricia Stevens modeling classes and Arthur Murray dance classes.

I especially enjoyed tangoing to “Kiss of Fire,” “Blue Tango,” and “Fernando’s Hideaway.” Tango was a slower and more graceful dance then without all the jerking around that characterizes it now. I could follow almost any dancers lead. I joined Friars Club, where I swam and square danced.

Patricia Stevens' classes helped me gain confidence and to realize that, although I was different from most people around me, I was okay. They told each student which colors they should wear and it turned out those were the choices I was already making. At the hair styling session, the instructor took each girl and told her how to cut her hair, style it and perhaps add color. She told me to comb my hair a little flatter; otherwise it was beautiful and shaped fine.

Then the makeup instructor gave some students elaborate instructions in foundations, eye makeup and lipstick. She got to me she just said that, if she had my coloring, she would not wear any makeup at all. Having people whose opinions I respected tell I was already doing was good; really boosted my self-confidence.

General Electric Aircraft Gas Turbines

I had just moved in and gotten a little comfortable with Cincinnati when I was offered a chance to go to the Watson Lab Class in Massachusetts. I didn't know what the Watson Lab was all about and I had just survived the stress of moving to Cincinnati, so I turned down the opportunity and did not attend that class until later.

I had been working at GE only four days when on Friday I was told to meet with an engineer named Eric Hattendorf. My first assignment working for him was a study of the shears, bending moments, stresses, boundary layers, and flow patterns from the turbine flange through the afterburner to the clamshell of the J47D engine designed for the F86D fighter plane.

Since this was completely outside my area of knowledge, I took notes on what he wanted and I borrowed Mark's Engineering Manual and scrambled like fury over the weekend to bone up on subjects Eric believed I already knew. The formulae contained Mach numbers and Reynolds numbers. On Monday Eric was not at his desk when I returned the book. Eric's officemate asked me how to find total pressure. I replied something like (I don't remember the exact formula), "The ratio of total to static pressure is equal to the mach number plus one raised to the Reynolds's number over the Reynolds's number minus one." He laughed and said that I must have really read a lot over the weekend.

The J47D axial jet engine had a 13-stage axial compressor, and twelve cylindrical combustion chambers arranged around the turbine shaft. In the combustion chambers kerosene was added and the mixture ignited. At full military speed, 7950 rpm, the gas reached 1650 degrees when it exited to the centrifugal turbine. Aft of the turbine, more fuel could be added to raise the afterburner temperature to 3200 degrees.

Luckily my work on architecture with my father had made me familiar with blue prints and so I took the blue prints up to the assembly line and with a flashlight got a three dimensional prospective on the engine. I determined the weight and attachment of each part so that I could compute the moments and shears. I made the results of my study into a small book titled "Estimation of loading, deflection, stress, flow characteristics, pressures and temperatures on a jet engine afterburner" with source material, graphs, weights, moments, shears, velocities, boundary layers, etc. When an important supervisor saw my results, he raised my salary by 60 percent. Then my work was classified secret at a higher level than my clearance, which at that time was for confidential material, so I could not see my own work.

My first adventure into the design of nomograph (alignment chart) began in 1951 with a relatively simple chart on which one could place a ruler from wheel diameter through RPM and across to get surface speed. There are no rules for creating nomographs; it takes ingenuity. Later I created some fancier ones. I wrote a paper: "The construction and use of nomographs to estimate turbine wheel rotational stresses." This one (for turbine wheels) took centrifugal force across to a series of lines indicating choices of a thickness to chord ratio (gamma) then vertically to a series of lines for the chord length and horizontally to centrifugal stress. The most complex one I ever created had thirteen series of curves transversed alternately horizontally and vertically.

We each had a calculator on our desk – Friden, Marchant or Monroe. Each had an advantage. The Friden could hold a multiplier for repeated use, but it was noisy and jammed frequently. The Marchant had an excellent trouble free record and sounded nice. My choice was the Monroe, the only one in which the result of a computation could be used in the next computation without re-entering it. All, especially Friden, jammed frequently and Margie was the one in our group who knew exactly how high to lift and drop them to un-jam them.

At that time many of our computations were done using a slide rule. I had acquired a circular slide rule (which I still have) that avoided having to change ends, but had two guidelines instead of one as in straight slide rules. It is still easier to do some computations that way, but no one seems to use them anymore

Our group of thirteen girls also worked on another engine. This new engine, the XJ something, was secret, so they assumed we didn't know much about it. They assumed that the girls in the computations group were too stupid to figure it out from the calculations we were asked to make. We easily discovered that it had a double-staged compressor, a torus-shaped combustion chamber, two turbine shafts, one inside the other, and two turbine wheels. We also knew that it flew at mach 3.5 at ninety thousand feet, enough information for us to figure that it was not an ordinary airplane engine. We kept the information to ourselves for fear we would somehow be thought to be outside out clearance level.

We also did some work on a ramjet, a jet engine without a compressor. It just uses the air coming in, slightly compressed by the shape of the entrance, adds fuel, ignites it, and blows it out the back. The main problem with a ramjet is that it doesn't work until the aircraft reaches a very fast speed.

Our office in the basement of the building had a concrete wall that divided the office section from a testing area. One day we heard a loud noise. Soon an engineer appeared and asked it we would like to see the secret engine we had been working on. He took us around to the other side of the wall where there were pieces of aircraft engine all over; it had exploded due to an improper weld. One piece of the turbine had gone through the ceiling and landed on a roof half a mile away.

Power Outages

There were frequent power outages in the building and no emergency lighting in our basement so we kept flashlights in our desks. When the power went out some of us played bridge and some played battleships by flashlight.

Sabotaging the Disastrous Boss

Our group of thirteen compulsive workers, the computations pool, initially had no supervisor. Jeannie, the one with a few days' more seniority, did whatever was necessary and we got along fine. Whenever people had a job for us to do, they just came to our door as asked for help.

After about five months, however, someone decided we needed a boss. Mr Boss decided to assign jobs in a strict rotation, which obviously couldn't work because different jobs required different amounts of time and different talents. I undermined his scheme quickly when someone came in with a job requiring spherical trigonometry, involutes and evolutes, and I was the only one who had the background to do the complex trig. He tried to assign it to about eight different girls in his order, but none could do it. When he asked me, I said I would be happy to do it when my turn came up, but that I had just started a task and my turn would not come up for several months. Then I went to the engineer who needed the job done and told him I could get it done in my lunch hour. It was easy if one knew how. My agreement with that engineer was that he would keep bugging the boss about it until the boss changed his job assignment policy. It worked.

I had a gift for design of involute-spline gear trains. There are no calculable solutions. Attempting to calculate a solution would be likely to result in something that could not be constructed, like a gear with 16 1/3 teeth. One can only guess and then test the guess. I was very good at guessing. The paper I wrote on this subject was "The computation of practical involute spline test measures."

On breaks, I walked along the new design assembly line, where partially assembled new engines were mounted on mini railroad cars. There assembly people teased me a lot, but I also made many friends who occasionally needed some help interpreting their blueprints. A paper I wrote to help out was "A remedy for a welding access problem on retractable variable jet intake screens."

One day we plotted to give the boss a scary tour. I asked him if he would like me to show him around the plant. As we walked around, friends caused loud noises and lit off their arc welders. We even timed his tour for an engine test where a jet engine went up to 7950 military rpm, ignited fuel to 3200 degrees, and glowed purple with the heating of the titanium alloy. We were perhaps sixteen feet away, insulated by heavy concrete in which thick glass blocks and periscopes provided visibility, so it wasn't really dangerous, but the noise and the purple glow really scared him.

Because the boss was ruining our group's morale, we were not getting as much work done as we had without him. He made us all on edge. He could not keep his hands off the girls, he asked them to help him look for things in the supply closet, and he interrupted concentration by asking for ridiculous help, like stopping whatever we were doing to staple some of his papers together. He moved things on our desks and looked over our shoulders to read whatever we were writing, generally managing to touch us as he reached across to point out something, dropping his cigar ashes on our desks and clothes and stinking up the office.

One day I decided to write a letter to one of the big bosses, relating these things. I put it in an envelope with a cover note and distribution list asking if anyone disagreed with my comments. Shortly, my letter was returned to me with twelve signatures below mine. The boss was gone before noon of the following day.

Job Related Education

The Air Force was educating hundreds of their pilots and repair personnel by having Jet Engine Repair and Maintenance classes at GE. The GE engineers were invited to attend, so I asked to attend too. They said I could if I got another female to attend also. They didn't think we would really do it. I crammed information into the other girl and we enjoyed the classes. We thought it was fun when the Air Force people sympathized with us, because they thought we were required to attend. A dinner for all the attendees followed the final exam. I had won the award for the highest scores on the tests. The other girl won the third place award and the attitude toward us at GE changed. Female employees got a better chance at the education fringe benefits.

General Electric had a continuing education program arranged with the University of Cincinnati.

To attend the program GE chose engineers by examination. My boss had bet another boss his week's salary that I could pass the test and break into the male-dominated program, so I agreed to take the test.

On test day I noticed that one of the questions was a trick question. It gave away that the answer was zero by starting the question with "in a closed system." Then I looked and found another trick question asking how high a fluid would rise in a three centimeter inside diameter capillary tube. If it really were a capillary tube, the diameter would have been in millimeters. I decided that all the questions were trick and all the answers were zero.

One of the fifteen who passed the test, I ended up taking thermodynamics, aerodynamics, electronics, etc., finishing up my degree in Physics and earning a degree in Aeronautical Engineering.

Performance group

I was transferred from the general computations pool to the Engine Performance Group. In this group we studied fuel regulation servomechanisms, stall characteristics, and behavior of the jets at different altitudes, temperatures, speeds and how the different functions related to each other in order to design fuel control servomechanisms that would prevent stall. One of my jobs was to produce performance maps. These were also called G & A Curves because one could use them to quickly get estimates to answer generals' and admirals' questions. On these maps families of curves represented the relationship between different flight parameters: Altitude, ground speed, mach number, fuel flow, air flow, drag, thrust, engine rpm, and stall line. Some charts had several families of curves. One large chart had six families of curves that I had color-coded and I could not figure out why my boss was not impressed until

I found out that he was colorblind. My paper was "Correlation and mapping of turbine performance data."

The Whistling Experiment

The engine group occupied one large room containing over one hundred desks in rows so close that one could hardly get through. Every day someone started whistling, and soon the whole group was whistling whatever was started. One day we had grown tired of the popular tunes, so three of us started whistling a tune from *Scherezade*. From then on, everyday someone would start it, and it was worse than any popular tune when one heard it over and over.

The GE plant was in several buildings and one frequently needed to go between them. Tunnels connected the buildings and, at each end of these tunnels, there was a bicycle rack where one could just take a bike, ride through and park at the other end - a good solution.

Designing Turbine Blades

After a few months I was offered a job in the preliminary mechanical design section in another building. My task was to assist in determining how to get more thrust out of an engine. One way is to burn to higher temperatures. Since titanium steel turbine blades couldn't take the higher temperatures, we were attempting to find out how to make a turbine with ceramic turbine blades. The ceramics could handle much higher temperatures but tended to fracture near their root with vibration. Once a single blade fractured, the rest broke in rapid succession. The turbine blades were fitted into fir-tree slots in the rim of an exponentially tapered turbine wheel. I was allowed a lot of freedom in brainstorming ideas to solve this problem. I was limited by needing the blades to fit into the fir-tree slots in the turbine wheel and obviously restricted by the necessary airfoil shape of the blades. I could change the shape of the base of the blade, change how the blade was faired onto the base, drill holes to change the heat and air flow, and change the rigidity of the base fitting. When I had an idea, three blades of my design were made and mounted onto a turbine wheel that was rotated and shook to determine what stresses the design could take. It was fun to let imagination go wild and see if an idea worked.

I wrote a couple papers during this project: " Estimation of natural vibration frequencies of airfoils" and "Node contour mapping for vibrating disks and airfoils?

My boss, Clarence Danforth, gave me a paper he wrote that proposed to offer a solution to the differential equations describing the vibration and stress on the airfoil shaped turbine blades. I could not make his solution work; but I noticed that it did seem to work if I changed the sign of one term. It was a while before I realized that the sign change, which Clarence had missed, was due to the fact that the equations contained second differentials of trigonometric functions. Because the angle of interest was 90 degrees, he had substituted one for the sine term not realizing the first derivative of sine is cosine, but the second derivative is negative sine.

The stresses could be estimated by solving the four simultaneous second-order, partial differential initial boundary value equations that describe the vibration of an airfoil. Since these equations have no known exact solution, we had to invent numerical solution techniques. A try using simple extrapolation techniques resulted in my writing “The accumulation of error in the solution of a fourth order partial differential equation boundary value problem using simple extrapolation methods.” To reduce this error I invented a technique that alternated forward and half-backward extrapolation, like taking two steps forward and one step back. We knew displacement and slope to be zero at the root of the airfoil and moment and shear to be zero at the tip of the airfoil. To solve the equations, we started with our two known values at the root and guessed at the other values. Then, extrapolating outward from the root, we would come to some values at the tip—wrong, of course, missing the two values known to be zero at the tip. Then we would start over with a different and hopefully better guess and repeat the process. The fourth time we did this we would be acceptably close to the perfect result. This required four solid days on a desk calculator.

I’m have no idea how, but I could guess the frequency of the primary and secondary modes of vibration of an airfoil accurately enough that I got telephone calls from people in other aircraft plants describing their foils and asking for my guesses. I was usually quite close. I got a reputation for this since it wastes a lot of time and money if you start testing a foil on a shaker and you are not close to its natural mode.

Starting to Work with Computers

One day I mentioned to Clarence that the keypunch operators in the computer group made ten dollars more per week than I did. He said that the $76.50 per week I was making was as high as he was allowed to go. With the advantage that I was already knowledgeable about a problem that needed to be programmed, I went to the computer section to work under Dr Herbert Grosch.

I programmed my solution of the simultaneous partial differential equations on an IBM-CPC, a 10-word memory wired-program computer. This machine and those that followed were descendants of the steam powered difference engine designed by Babbage in 1822. This machine, Card Programmed Calculator, was one of the earliest programmable computers. It sold for $50,000, consisted of four cabinets – a card reader, a cardpunch, a calculator and a memory cabinet with a 16 10-digit number capacity. It weighed 5458 pounds, required 43.8 amperes of 230-volt electricity (10 kilo watts). As it ran it generated 22930 BTU of heat per hour. Logic paths were supplied via plug boards (control panels). Because the program resided on the punched cards, it was not a stored program computer; there was no theoretical limit to the size of the program. The CPC looked like a combination of an old-fashioned telephone switchboard and a large card sorter. Intermediate results were punched out on IBM cards into one of the sorter slots and later feed back into the card reader. . The room was so crowded with equipment that, when I became pregnant (that was not supposed to happen – two different gynecologists had said I was sterile), Dr Grosch, half kidding, offered to move equipment so I would have room to work.

An IBM card was about 3 by 8 inches and had a format of 12 rows and 80 columns. Hollerith had first developed punched cards for use in the US Census in 1890. The first row was the "+", second the "-" and rows three to twelve "0" through "9". Usually the last eight columns were used for ID and sequence numbers so that if you dropped the cards or the reader spit them out you could get them back in order.

Each program had its own large wired board. Mine got so heavy that I was offered help to lift it into the computer. Each run to determine the deflection, slope, moment and shear on a vibrating airfoil took 45 minutes. This was a big improvement on the previous thirty hours using a desk calculator.

A Stored Program Computer – the IBM701

Later that year General Electric purchased an IBM701 computer. It had stored program and conditional control transfer capability. Its electronic switches were based on diodes and transistors. The programming language was an assembler called SP2. It was a machine language operation language that allowed symbolic data location. The IBM 701 computer had an 8K electrostatic storage, (huge at the time), intermediate magnetic drum storage, with two tape drives, a card reader, and a slow huge printer. They advertised a multiplication speed of 456 microseconds. The 701 broke down frequently, so one never pressed a button or fed in cards unless the maintenance man was present. The computer had huge vacuum tubes and the wind blew like a gale in the room trying to keep them cool. I believe this was the first computer using binary floating-point numbers i.e. each number was expressed as a fraction (mantissa) and a power of two (exponent). To do the floating-point arithmetic (which was not hardware), the programs used interpretive routines (later similar to macros) so that each time one wanted an arithmetic operation, the assembler imbedded the necessary code.

I don't know what serial number our 701 was but IBM's famous Watson said "With this dedication of the fourth IBM701 computer we have installed enough capacity to meet all the computing needs of the United States for the next one hundred years".

On it I programmed the solution of the same differential equations. It was both easier and faster. I was probably the first to run into a new computer problem – called underflow, a computation resulting in a number too small to be expressed. I was also the first to use an overlay technique wherein part of the program or data is stored on a mass storage device (at that time magnetic tape) and later retrieved to perform additional computation. This is so common now that we don't even know it is happening when information from a disk is entered into the random access memory. The tape units were about six feet tall and three feet wide and at the top held two twelve inch diameter reels with a read-write mechanism between them. Under each tape reel was a vacuum column with a sensor to even out the speed and prevent stretching of the tape. The bit density (200 bytes per inch) on the 2400 feet of magnetic tape per reel was so low that one could sprinkle the tape with iron filing and read it. Each byte on the tape was seven bits, a six bit ebcdic character with a check bit.

I worked there until shortly before my son, Tom, was born at the end of June 1954.

Cincinnati Symphony

Also during that period I played cello in the Cincinnati Symphony. I have LP (33 rpm) records of our rendition of Mozart's "Ave Verum Corpus," Schubert's *Unfinished (B-minor) Symphony*, and Franck's *D-minor.*

Leaving Cincinnati

When because of serious pregnancy problems I could not work any longer, we drove from Cincinnati to Park Ridge. We didn't pack anything; so we lost everything that we had in the Cincinnati apartment. We lost my Marquette mementoes, all the papers I had written while at GE and all the graduate school records of the aeronautical engineering classes – aerodynamics, thermodynamic, electronics, jet engine fundamentals, etc. They could not be replaced because a fire destroyed some University of Cincinnati records. They offered to reconstruct the records if I could give them course numbers, titles, instructors names, dates, proof of attendance and grade. I, of course, had nothing but my memory. I couldn't even get a copy of the papers I had written because in the meantime they had been classified secret.

Hardware Store

For a while we lived in a small town. Carney's was a hardware, paint, glass, garden, boat and gift store. It was bigger than the average Ace Hardware, but smaller than a Home Base. When I applied for the job, Mr. Carney said he was looking for a strong male employee, but since he also sold gifts he would give me a chance. Little did he know that I knew more about tools, saws, screws and bolts than about gifts. Luckily most of the customers wanting gifts wanted to look for themselves and Mr. Carney discovered that I was good at all the other stuff. I was particularly good at helping people pick paint colors and wallpaper. I sold tools, mixed paint, and loaded sackcrete and fertilizer onto trucks and learned a whole lot.

Helipot – Beckman Instruments

We moved to Costa Mesa, where in August 1956 I secured a job with the Helipot Division of Beckman Instruments. The job entailed programming to design non-linear rheostats on a poorly designed, drum-driven 16K computer called the ALWAC.

I had a desk between two men. Hans was a brilliant hard working refugee who did not have good command of English. Phil was a lazy, sloppy man who wore sweaty dark suits and acted like a big shot. Because many workers assumed he was a boss, he got away with telling others to do the work assigned to him.

My first task was to create a symbolic language and write an assembler for the ALWAC. The main challenge was its weird addressing scheme and the contortions one had to go through to write any efficient program on a drum driven machine.

A paper I wrote there was “Slope discontinuity considerations in fitting non-linear potentiometer designs with straight line segments.”

When I got a job at Convair and gave notice at Helipot, my exit interview at Helipot was fun. I complimented each of the people who had really done the work credited to Phil until the boss caught on. I left without saying anything bad about Phil and with the satisfaction of a job well done.

San Diego

In the fall of 1956 we moved into a small house east of San Diego. In the back we had an area christened the weed garden. My son Tom was so concerned about living things that he couldn’t stand the idea of killing weeds; instead, we transplanted them. He couldn’t stand plants that were alive being discarded. In fact we even had to have funerals and pretend burials for burnt out light bulbs.

I had I collected over fifty different succulent plants. One day when one bloomed with a single, unusual star-shaped flower, I sent it with Tom to kindergarten. It had an awful odor, and I was amazed that the teacher kept it on her desk all day

In 1959 we bought our first television – a 13 inch black and white that was usually tuned to the test pattern; it had ghosts and wasn’t vertically stable. When it worked it seemed to be always on boxing or football. It requires real enthusiasm to enjoy watching those sports on a small black and white poorly focused TV with poor camera work and the image rolling over.

Convair – General Dynamics

I was a programmer in the Convair General Dynamics, San Diego, computer section from the fall of 1956 through June 1959. I started at four hundred dollars a month.

When I first started at Convair the computer group was overflowing their allotted floor space. To accommodate about twelve additional employees they put desks for us in the crawl space under the wind tunnel. The ceiling height of our room was about 66 inches striped yellow and black to remind us not to stand up. I had to stoop slightly to walk around. When wind tunnel tests were being performed it was very noisy and vibrated. To get to our workplace one had to walk down a steep ramp and duck under the beams at the entrance. It may seem strange, but we liked it there. When they reorganized space we were moved into the main computer group area.

Our Cubicle Group

Charles, Robert, Libby and I, four of the five who worked in the same cubicle, played bridge together during our noon hour. A couple of us would go down a fire escape ladder, climb a cyclone fence, walk through a huge drain pipe and cross a corner of the San Diego Airport to get to a lunch wagon to bring back food for our foursome

Four-foot panels separated our work cubicles. Each was just large enough to allow three standard desks across the side opposite the doorway, a desk on each side of the doorway, and one five-drawer filing cabinet with a drawer for each of us. These drawers had security bars for classified material. We made precarious stacks of work: cards, resource material and printouts on, under, and in the kneehole of our desks, leaving just enough room in the center to lay out current work.

Because there was not room enough for extra people in the cubicles, the usual method of communication was yelling. It was very efficient.

Our stress level was pretty high, so by about three on Friday afternoons, no one was accomplishing anything. We called it our slaphappy time. We would conduct silly surveys, such as how many people had aspirin or Tums in their middle desk drawers. One Friday we had a contest to see who could produce the longest computer program that, without repeating itself, did nothing. Obvious entries included adding zero, multiplying by one, moving to self, and Boolean ANDs and ORs.

Convair Computers

Besides the difficult to program IBM650, the unreliable Sperry Rand 1103 and the limited capability IBM407 accounting machine, Convair had an IBM704. It had similar hardware and capability to the IBM701 I had programmed at GE.

The first computer I worked on at Convair was an IBM650 bi-quinary 4k drum driven machine. I could be very inefficient if not well programmed, wasting a drum revolution each time it had to retrieve something off the drum. We also had an IBM407, a very basic accounting type computer with wired programs.

Convair acquired an 8K IBM704. Initially al programs were written in assembly language that translated each statement one for one into machine code. Since it was still quite slow and the operator could tell what was happening by watching the lights blink, we were heavily dependent on the operator's judgment. The operator was supposed to stop our program if he judged that nothing useful was happening. Sometimes he misjudged and we lost our results. Therefore on it I wrote the first software that allowed the computer to talk to the operator while the program was running. The program could say things such as, “”Expect this program to be in a tight loop for the next few minutes”. “Stop this program if another message does not appear within four minutes,”” or “Please mount a tape for a dump-restart and press sensor 3 when the tape is ready.”

This made the bosses angry: Computers were not supposed to talk to the operators, whom it might distract. However, when I asked the bosses to consult with the operators before requiring that I remove the software, the operators voted unanimously in my favor. They liked the timely instructions on the computer, not on the usual scribbled piece of paper supplied by the programmer so soon other programmers were using my routine to talk to the operator.

Time on this computer was measured in centi-hours (36 second chunks) with each centi-hour costing about four dollars.

Five Projects

During my time at Convair I was assigned to five projects (1) Heap; (2) Air-loads, (3) Area Rule Drag, a program to estimate the drag on a supersonic plane, (4) Lift-off orbit and re-entry project, (5) Tukey. (which ended up at UCLA) It seems like most of the projects I get involved in involve calculus and differential equations.

1- Heap

Heap (I don't remember the reason for the name) was a program to intake test flight information, integrate it and obtain the stability derivatives of the F-102A airplane. These derivatives are coefficients of a set of differential equations. Data utilized by least squares averaging gives rise to a matrix that is solved using triangulation.

2- Air-loads

Shear, moment, and torque air-loads, on a wing or fin were computed by chord-wise and span-wise integration over their surface.

3- Supersonic Area Rule Drag

Another time the bosses asked if I could find an error in a program intended to estimate the drag on an airplane at selected supersonic speeds and attack angles. The program, as I received it, required four hours and twenty minutes on an IBM704 to compute results for a single speed, costing $1,730 per run. Also, because of an error, it frequently gave back obviously impossible results. Even when results seemed reasonable, they could not be trusted when there was a known error.

The drag on a supersonic airplane is estimated by doing a little numerical calculus on areas of the plane intersected by a cone as it transverses theairplane. The mach number of the speed being investigated determines the "sharpness" of the cone. Many many coordinates of the airplane surface following buttock-, waterline- or cross-section described each component (body, wing, engine, etc) of the airplane to the computer. If one takes a cone and tries to determine the area of the plane it intersects, the job of finding all the intersections is difficult and time- consuming.

On the other hand, if one reverses the thought process and realizes that every point on the surface of the plane is an intersection of some mach cone, determining the correct one is easy. Thus the solution to doing this integration rapidly is to compute all the cone areas at one pass, incrementing the appropriate cone area for each point on the plane's surface.

Having figured that out, I asked permission to reprogram; the result produced a reliable estimate of drag that, on the same computer, took less than two minutes per mach velocity. This allowed the airplane designers to study models at many speeds since the cost was less than nine dollars (1/200 of the cost of running the previous program). Just to blow away my readers, here is the formula for drag.

$$\text{Drag} = -1/(4*\pi^2*s) \int_0^{2\pi i}(d\theta) \int_0^{f(\theta} \int_0^{l(\theta)} s'(x,\theta)*s'(\delta,\theta)*\ln |x-\theta| dx d\delta$$

.

Where s' is a Fourier series and the Fourier coefficients are evaluated using Tchebichev polynomials

4- Lift-off Orbit and Reentry

This space flight guidance program was used early in NASA's trials at a time when no ship had yet made it into orbit. I had only worked on this project for a short time before responsibility for it was transferred to Astronautics. I did not choose to transfer to that installation because it would have been difficult to travel there.

Layoffs

By the beginning of 1959 Convair was suffering some major cutbacks. One of the perverse things we did then was to bet on who would be laid off next. At first it was easy - new hires goof-offs and those deemed overpaid. Then criteria grew more complex. We wrote a program that weighed in salary, productivity, seniority, background experience, age, sex, number of dependents, etc. We made a game of it hacking into the plant's accounting system for employee information..

After about one quarter of our staff had been laid off, I realized that, at my salary, then $635 per month, I was getting close to the top of the lay-off list. After a scout from Remington Rand approached me to offer a higher salary ($725), I asked the boss where I stood. He encouraged me to transfer. I don't know whether that meant that I was at the top of the list or whether it had just saved someone else from lay-off.

I was asked to give three-week notice and during that time I became used to a new experience – I spent hours in a conference room teaching other programmers my techniques. Until then I had not realized how many new and ingenuous things I had created.

Scripps Institute of Oceanography

At the same time as my work at Convair I was working on contract about sixteen hours per week at three dollars per hour for Dr Walter Munk at Scripps Institute of Oceanography in La Jolla. On weekends a UCLA student would pick me up at my house and drive me to UCLA, where he would study as I worked with their IBM7094 computer.

My job was to write a spectrum analysis program to analyze underwater disturbances using Fourier transforms. These disturbances could be caused by volcanic eruptions, earthquakes, tsunamis, storms, depth charges, explosions, meteorites, landslides (slumps), movements of the tectonic plates, or movement of the earth magnetic poles (did you think they stayed put?)

and, of course, the moon, sun and planets. I realized that, with tides eliminated, the sum of the wave heights and troughs over time must be zero. This realization allowed me to integrate from whichever beginning or end of the series was closer and thereby cut the time and cost in half.

If one eliminates surface chop by placing the recording instruments at sixteen fathoms, and then mathematically eliminates the known tidal waves, the other waves can be examined. Ordinary ocean waves or trapped waves have a period of five to twenty seconds and crash on land at speeds of ten to twenty miles per hour. Tsunamis have a much longer wavelength typically between ten minutes and two hours and can travel at a speed of six hundred miles per hour.

The program allowed for up to five hundred points on the frequency scale and any number of recording data points. The number of frequencies was restricted to less that one-fifthiest of the number of recording points for reasonable result reliability. There are four equations to analyze the spectrum plus equations for coherence, beam width and phase lead. For example this is the (slightly simplified) third, the one that gives the qua-spectrum:

$$C(j)=<X_{i*}Y_{i-j}>_i \;\;|\;\; Z(k) = [(\sum_{j=1}^{n-1}(1+\cos(\pi_*j/m))_*\cos(\pi_*k_*j/m)]\,/m$$

Using any one stream of pressure recordings, the Tukey program could determine the length and amplitude (energy) of the waves represented.

Using the pressure recordings from a simultaneous pair of these transponders, the program could produce two correlations: the in phase (co-spectrum) and the out of phase (qua-spectrum), along with the coherence and the phase lag between the two records. With this triangulation can determine the probable source and magnitude of the disturbance.

Pressure transponders were placed sixteen fathoms deep off La Jolla, the Baja coast, Santa Teresa Island, Hawaii, etc. One wave we recorded on the La Jolla channel and the St Teresa sensors was traced to the waters south of New Zealand. By the time it reached the California coast, it was only one millimeter high. It was distinguished from local waves because it had an unusually long wavelength and persisted for several days. It was propagated thousands of miles and was the result of a severe storm.

At the time I was working on this program the data was transmitted by radio to Scripps institute, punched onto IBM cards that were then fed into a computer that was also being used for other things. Obviously there was a considerable time delay before the results were available. Some years later a computer was devoted to receiving and analyzing the waves in pseudo-real time. .

We also used sensors mounted on iridium rods deep within abandoned mines to study the movement of the earth's crust. Another project involved monitoring the migration of the earth's magnetic poles.

One disappointment was that I wasn't able to go out on the Scripps' ship, but I had mixed feelings about that because they had put so much research equipment on the deck that the waves really rocked the top-heavy boat, and seasickness was common.

Naval Tactical Data System

From mid 1959 until June 1960, I worked for Sperry Remington Rand on a Naval Tactical Defense System contract on Point Loma, a steep, narrow peninsula 500 feet high, sticking out into the ocean a little north of the center of San Diego. It had a lighthouse at its end, Rosecrans military cemetery, some military emplacements, old, mostly deserted barracks buildings, Navy Office Buildings and not much else. One passed through a military inspection station on the only road out onto the peninsula.

In one of the deserted barracks buildings one boss, one secretary and about fifty military and civilian workers set up our offices. My office was on the second floor on the ocean side with a good view of the military funerals going by and with whale migration close off shore.

Security was tight and I ended up with secret, Q, and government security clearance (so I could carry secret documents). I was offered a gun, permit and training and generally made to feel paranoid. Noone seemed to realize that the information in my head was more valuable than whatever paper I carried with me..

When I arrived the countess had no software and machine code instructions were entered on a Teletype machine. That is a gross mismatching in speed to enter information by hand onto a very fast machine. Typical behavior at the time was to spend an hour entering your program, a second executing the program and another half hour printing the results back on the Teletype.

The countess computer consisted of a pair of semi-independent 65K word computers, each with seven input output channels and each capable of doing four things at once if properly programmed. The computer had a millisecond internal clock and a speed dial. One could slow the process down to look at, or record, the console lights.

My first self-assigned task was to write an assembler for their Siamese twin "Countess" computers. It seemed unbelievable to me that one did not already exist – assemblers for other computers had been around for at least six years. Writing an assembler is an interesting process since when you start out there is nothing except the hardware. One has to start by understanding what each series of ones and zeros means to the computer and then entering the ones and zeros to form a basic start up program – a bootstrap.

I also wrote a ship-radar-to-ship-radar communication program, an anti-collision program, and a pursuit guidance program.

Radar to Radar

Using the radar-to-radar communication program, a radar operator on one ship could click a blip, and a circle would appear within two milliseconds around the corresponding blip on another radar up to 256 miles away. To perform this coordinate transformation between the ships two 3 x 3 matrices were formed. One contained terms derived from the position of the two ships and the position and altitude of the target blip; the other contained terms that were spherical trigonometric expressions. The product of these translated down to the center of the earth and back out to the other ship's position. Then having determined the target position and altitude with respect to the receiving radar, it placed a circle around the corresponding position located on the receiver's radar screen. To do this task required multiplying two three by three matrices in which each term contained at least one trigonometric function. Because one of the angles, the one at the center of the earth was always small (20 degrees or less) I wrote special faster subroutines to evaluate trig functions of small angles

My paper on this project had a ridiculously long subtitle: "Long distance inter-ship communication radar-to-radar point correlation study – A study of the effects of radar errors, ship instrumentation errors, and error due to ambient conditions in combination with time lags and approximation errors and the effect these errors have on the accuracy with which radar operators on distant ships can point to blips on each other's screens by pointing to the corresponding blip on their own screen".

My first adventure on the Navy's two countess computers, to try the radar communication program, was scheduled from midnight to three a.m. Only a gooseneck lamp lit one of the computers and the glow from fifteen radar screens barely lit the football-field-sized, windowless room. Cables formed a web across the floor and catwalks loomed overhead. It was the first time I had been in the room, and I had no idea that I would be alone. It was spooky, especially when I went outside and found nobody except the guard at the gate. I was alone and scared. Though I did not even know how to turn the computers on, I did know that time on a computer was difficult to secure and expensive, so I was wasting it if I didn't get something accomplished. With all the military stuff around, I was afraid I would send up a missile or something if I pressed the wrong button. I found a manual in a drawer, read as fast as I could, turned the computer on, fed my program in on a teletype machine--its only input device--set timers, and ran my program. The clock that read in milliseconds registered zero. It took me some time to realize that I should loop my program one hundred times in order to time it. The radar-to-radar communication program took only two-tenths of a millisecond.

Our computer time was usually scheduled between midnight and dawn because the administration assumed that as staff professionals, we were willing to work a forty-hour dayshift and return at midnight. Encanto, where we lived, was about twenty-seven miles from Point Loma, so it was a lot of work and travel time, and the midnight runs were always alone. The Navy used the computers during the day.

We were encouraged to treat the computers poorly, to leave jackets draped across air inlets, to smoke if we wanted and to drink coffee or colas while operating the computers to test if they could take the kind of abuse they might receive on ships at sea.

Pursuit and Avoidance

One problem with designing defense systems is to determine which targets (or unknowns) to watch. I designed a method for scanning targets and attaching priorities. Obviously the closest and the fastest targets need to be monitored often, but one cannot ignore the others or they can sneak up. My design allowed less threatening targets to come to attention with a frequency lower than the more threatening targets, but with a frequency proportional to their potential threat.

The anti-collision program was designed to extrapolate ahead from all the targets visible on the radar. Then it was to warn the lower of any two "friendlies" that were expected to get too close to descend immediately. Of course, if either aircraft was not friendly, we warned ours and suggested an avoidance maneuver.

We designed the pursuit program to allow three pursuit options. A collision course is the fastest and best if the target stays on its course. A pursuit course, aiming at wherever the target is at the moment, is good if one has a speed advantage. The third option, called a lead pursuit course, aiming at a point in front of the target, but not as far ahead as the collision course would select, was a good compromise most of the time; it was close to the course most experienced pilots would take instinctively.

I had top secret, "Q" and government courier clearances, complete with the briefcase with handcuffs, and I knew much about our defenses on the west coast and at sea. That scared me because I knew how easy it would be to coerce information out of me. I wondered why they were so worried about the information I carried, when there was a great deal of secret information in my head.

NTDS Work-mates

I was the only female staff member, so I greatly enjoyed being included in the noon hour horseshoes and ball games.

My office was on the ocean side of an old barracks building five hundred feet above the ocean on a slope too steep to walk. From the window one could watch the whale migration. From the opposite window one could see the naval air base with its runway pointed right toward us.

Someone would yell when the blue angels appeared on the runway and we watched them practice. One-day a plane's engines failed and the pilot nosed over into the bay to avoid hitting our installation. We scrambled down the hill –the bay side was not quite so steep-- and watched as a Scripps underwater exploration ship, which just happened to be nearby, went unsuccessfully to the rescue.

Stress Test

The navy was doing what they called neuro-psychological stress testing and I volunteered to be a subject for one test. It was lots of fun, but I don't think that they intended anyone to enjoy it. One sat in front of a console with a circle of nine numbered buttons, each with an associated light, and one light in the middle. The lights in the room flashed on (day) for four seconds and then off (night) for four seconds. Some buttons were day-active, some night-active. Each turned lights on or turned lights off, some actions were immediate, some delayed and some sequential. In each of six tests, a puzzle program was fed in. The goal was to determine the order in which one could press only buttons 4,5,and 6 to get the center light to come on. The other buttons, 1-3 and 7-9, cold be used to help in figuring out the effect each light had on the others The solution might require pressing series of buttons over several days and nights.

Each puzzle was more complex. After I had worked on the sixth for a short while, the operator told me to start the puzzle over. When I finished, I asked the operator why he had told me to restart number six. He said that I was the first person to solve number five and get to the sixth puzzle. He didn't want an early mistake to stump me, since he really wanted to know whether I could solve it. He said he usually had to be ready to duck because many of the volunteers threw something at him and walked out before the end.

My Escape

My husband was becoming violent. So when I decided to leave him, I wanted my plans to be secret. I had been looking for a job in a remote scientific or military research base maybe in Alaska or New Mexico, -but had not succeeded. Most of the employers wanted men. Because I had secret clearance it would be particularly difficult for me to just disappear.

In that final week at work I quietly wrote progress reports and enlisted some other employee's help with each of my projects, making sure that they could carry them when I left. On a Friday morning in May of 1960 I went to work as usual. I took my progress reports and information about colleagues who would carry them to the Remington Rand division president. When I explained what I was going to do and why, he said he was sorry to lose me but that he understood. He handed me a termination request form dated two weeks earlier.

For seven hours I drove eastward on major highways as quickly as safety and the law would allow. Then I figured I had better find a motel off the major roads. 1958 Volvo's was rare, round and high, and easy to spot from the highway. The rest of the way to Illinois we stayed off Route 66, not an easy thing to do in some areas.

Argonne National Laboratory

Shortly after coming home to Park Ridge I applied for a job at Argonne National Laboratory. On Tuesday I made the appointment for a Friday interview. Early on Wednesday I was in an accident. I showed up for the interview with a separated shoulder, stitches across my head and a badly bruised leg. I don't know how I successfully made it through the interview. I was scared that they would question all the scrapes and bruises. I had been earning $740 so they were apologetic when offering me the $675 that I immediately accepted.

Argonne is one of at National Atomic Energy Commission Labs devoted to research into peaceful uses of atomic (nuclear) energy. It is located thirty-two miles from my house on a

3700-acre wooded site just south of interstate route 55 (the famous old route 66) near Lemont, Illinois. Except for a strip bordering the perimeter cyclone fence, parking lots, a small area around the buildings, a small old cemetery and a prairie reclamation field, the whole site is heavily wooded and is home to an inbred group of over 200 Himalayan miniature white deer. The story is that someone brought back a pregnant deer as a pet; it then mated with its own offspring and started the whole inbred herd.

Within the fence about 5000 employees work in over 50 scattered buildings. The east area is mostly business and administration. The west research buildings are named for the main research going on in them – Biology, Bio-medical, Chemistry, Chemical engineering, Solid state science, Applied mathematics, Reactor engineering, and Zero gradient synchrotron. Most of my work involved only the Math, Physics and Bio-med buildings. The physics building and the biomedical building each also house extensive libraries where at any given time 90% of the books were out on loan and they would not know what to do with the books if many were returned at once. The biomedical building also houses a plant and animal section, with snail, grasshoppers, crickets, eggs and various plants. Some of the plants were supported in gimbaled rigs to cancel out the effects of gravity. The physics building also has a "clean" area deep underground.

The Argonne Physics Building

My office as applied mathematician was initially in the physics building and latter in the applied mathematics building. My job there extended from mid 1960 through fall 1978.

My first group leader, Dick King, was a pretty good supervisor except for one habit – he cornered people and got up uncomfortably close, so you could not get away. We devised a system for handling this habit – Whenever anyone noticed someone feeling trapped, they made up an excuse to break it up – usually a question or an important telephone call the trapee had to come answer.

Our group also had an interesting secretary. Mae liked to munch on popcorn so she brought in a popper. Since there was no spare outlet near her desk, she put the popper on my desk and I listened to the popping and smelled the popcorn every day. She was very good at following directions – If you wrote a letter and circled a phrase and wrote "underline this" in the margin, it would come back just that way – with the phrase circled and "underline this" typed in the margin.

Norber (my office mate)t and I became the Argonne representatives to the IBM SHARE group where scientific computer users got together for a week twice a year and traded information and programs.

Argonne – 1960 Computers

In 1960 the computer room was in the basement of the physics building. The room, about 30 by 50 feet, was crowded with an 8K 32-bit word IBM704 and a 4K-byte IBM1401 Peripheral processor. Most of the programs to be run on the 704 went from cards onto magnetic tape on the 1401 at 400 cards per minute. Then the tape was rewound and transferred to a

704-tape unit. The program was run and the results written onto another 704 tape unit which was then rewound and remounted on the 1401 for printing at 600 lines per minute or card punching at 150 cards per minute. The 1401 rates were about four times as fast as the reader/printer/punch within the 704's own hardware. The program I wrote to handle this was called PEST (Peripheral Equipment System Tasking).

The 80 column 12 row (+, -, 0-9) standard IBM cards could be punched in three formats. The first format, Hollerith, used a column for each letter, number or symbol, e.g. "A" was indicated by a punch in the "+" and the "1" rows. In Binary format each location corresponded to "zero" if full and "one" if punched. The third format, Condensed, was my invention. In it codes were used to indicate repeated symbols and half columns indicated letters or symbols (in a format called EBIDIC coding). My program, an ancestor of ZIP, was called DEW and its reversal, an ancestor of UNZIP, was called UNDEW. This reduced the number of cards or size of the file by at least half and frequently to as little as one-tenth. Computer decks with over two thousand cards were not unusual. Since the cards were heavy to carry and reading them limited the speed that the computer could run, my condensed format and programs were an advantage adopted by many companies within the SHARE group.

The IBM 1401 Computer

We thought of the IBM1401 as a very small computer. At 4K it bytes was small in capacity, but now would be considered physically huge. It consisted of five cabinets. Three of these: the CPU, the reader/punch, and the printer were each about three feet by six feet by five feet high. The two tape unit cabinets were about two by three by seven feet tall. Each contained (left to right) an input reel, a vacuum column, a reading station, another vacuum column and a take-up reel. The tapes in use then were on twelve-inch reels. Reels contained 2400 feet of half-inch magnetic mylar tape and recorded at 200 bytes (characters) per inch. Many of our current computer files would be too big for one tape with capacity less than half a megabyte. One could take a piece of this magnetic tape and scatter iron-filings on it and read the EBIDIC or binary code visually with an ordinary magnifying glass.

Shortly after Argonne acquired the IBM1401, I wrotee some demonstration programs to show off its capabilities. As you probably know, computers don't look very impressive while doing real mathematical work so I taught the IBM1401 to solve mazes, to present and grade multiple choice quizzes and to play music, melody only. It was fun and impressed people, but the music was severely limited since the computer was not fast enough to produce high notes.

The IBM 704 Computer

The IBM704 was still a tube type of computer, no transistors, and no core memory or miniaturized circuits yet. Tubes generated lots of heat so the room was air-conditioned and it was like working in a very noisy cold wind tunnel. This computer was still slow enough that one could watch the lights and get an idea of what was happening. The console had light displays of the address, arithmetic registers, and the indices. It also had six sense switches available to the operator to cause changes in the program logic. One of these might be assigned by the programmer to initiate a trace, *i.e.* cause the computer to print out each step it was taking. Another might cause a dump onto magnetic tape, a post mortem print out of the condition of the registers and the contents of the memory. Even with only 8K memory, the dump, when printed from the tape on the 1401, was quite a volume of paper to look through, but it was usually the only was to find an error.

Fortran Preprocessor

One interesting thing about computer languages is that, in contrast to spoken languages, in computer languages one learns to write before learning to read. At this time the popular language on the IBM-704, Fortran, did not have good diagnostics. One frequently got back a result that simply said, "compilation failed" and gave no clue as to why. The idea of a Fortran Preprocessor was formed at a Share discussion when someone proposed writing a Fortran II Compiler to run on the IBM 1401. This idea was opposed, but my boss, John Denes, thought that some sort of Fortran checking program might be possible. Although I seven years of experience in computer programming, I had never written a program in the Fortran language and I was completely unfamiliar with the IBM 1401. We did not even have that computer yet. I began in June of 1960. By October I had the program written, but it far exceeded the capacity of a 4K 1401. By December I had split the program into seven records that could be overlaid in a 4K memory. To debug I had to mail the box of IBM cards to Bartlesville Oklahoma with a one-week turnaround. In February 1961 Argonne's 1401 was installed and things got faster. The finished 2038 instruction program "DDT" (get rid of bugs) looked for errors in Fortran programs and printed helpful diagnostics and statistics. Since it ran on a 4K IBM-1401, it was a cheaper and generally easily available help to debugging Fortran programs. After check out at over 50 Share installations, we started distributing the DDT program through SHARE in April but the program became so popular worldwide that IBM took over its distribution in 1962.

Share

By 1962 the scientific computer user SHARE group was sharing so many programs that it was getting difficult to manage. My office mate and I undertook to organize all the SHARE programs. We asked each author to provide an abstract of ten lines or less for their program A Category Code, Serial number, Program name, Author and Abstract were then punched on IBM cards and a manual of over eight thousand was printed and distributed. Among my contributions to this share library were the 704 systems functions: sine, cosine, tangent, cotangent, square root, modulus, logarithm, and exponentiation, as well as the 1401 programs: DEW, UNDEW, PEST and DDT.

IBM built safeguards into their hardware to protect the user. Frequently these safeguards prevented a knowledgeable user from doing something useful. One useful, but forbidden, trick was to put in an address too large for the computer. If allowed, the computer would then respond by indicating its largest valid address, so one could find out how large the computer was and scale tables accordingly. IBM called the safeguard “invalid address prevention;” we called it the “valid address inhibitor”. We wrote programs to be submitted to IBM’s SHARE group. Since members had computers of differing sizes, it was helpful for a program to know how big the computer was and to modify itself to take advantage of the size. For example some data files could be kept in core memory on a large core computer, but written onto tape on a smaller one. Another thing forbidden was negative numbers in address fields. Sometimes it is handy to look for something backwards. At the SHARE meeting we prevailed on IBM and they took the inhibiting hardware out.

.

New ANL Applied Math Building

After a few years the Applied Mathematics Division obtained its own building. Its computer room was about 100 by 150 feet entered only by key card. It had a raised floor with the cables and freon cooling beneath. As one entered the area there was a storage room on the left, an input/output organizing room containing the card-reader, cardpunch and printer in the center and a keypunch room on the right. The computer room initially housed a 256K-byte IBM 360-50. Soon a 512K-byte IBM360-75 was added and the –50 became a controller for input output and for twenty-four Teletype stations.

The hallways and three walls of each office were painted light green. We could choose a contrast color for our door and for one wall. I chose a darker green. Later when we played musical offices, I ended up in and office with the contrast color gray. I liked that too. This office faced north and had a large window which looked out at the parking lot and across to the forest and to an area where the deer grazed. Some of the employees who were afraid of the deer came to look out my window before going out to their cars.

The previous office had faced east, the window wall of the building. One day I was sitting with my back to the window and felt a breeze. I stood on a chair and discovered the glass did not reach the upper frame. When I reported this that whole side of the building was evacuated and we shared offices with someone in another hall until the window wall was disassembled and reinforced.

Early Graphics Equipment

Connected to the IBM360-75 were two new graphic devices. The IBM 2250 Cathode Ray Tube programmable graphic display had a maximum speed of forty refreshes per second. It could only produce line segments in black and white – no color, no shades of gray, and no filled-in areas. Theoretically it could show motion. Practically, if the content of the screen was non-trivial, (more than a few line segments) the refresh rate dropped to ten to thirteen refreshes per second and any attempt to produce motion was jerky and tended to produce nausea if watched for more than a few seconds.

The other device was an IBM 2280 Film Recorder. It was supposed to produce 35 mm film. It was supposed to record and develop pictures them as they passed between two vacuum columns. It was an impractical device – registration of pictures was poor and developing uneven. In spite of all the imperfections I spent time getting the system to produce usable images. Our only notable success was producing a stereoptic view of rotating p-toloene sulfonic acid crystal.

Plotting

The ability to output graphic information was just beginning and was difficult, so I wrote a program called QPLT (quick plot). With it one could easily produce a graph in linear or log scale as a group of points, a curve fitted to some points or a histogram. Scaling and coordinate labeling were automatic for both linear and logarithmic scales. I produced a group of examples including some neat lisague figures.

Control Data Corporation 3600

Soon computer use demand exceeded the capacity of the current computers even running 24-7. Argonne contacted Control Data Corporation and contracted to spend five million dollars on the combination of CDC3600 and CDC160A provided they passed a benchmark test. Ron Krupp and I were assigned to write a benchmark program to test the functions, speed, and intercommunication between the main computer, the 65K 48bit word CDC3600, and the peripheral CDC160A. I wrote the 3600 part while Ron wrote the part for the 160A. We were sent to CDC's home in Minneapolis to debug and test our programs. We were left alone with full use of the computers from 11PM until 7AM. At that time the computers, in the process of being built, consisted of circuits mounted on sheets of plywood with toggle switches and Christmas tree type indicator lights and cables all over the floor. Every night the computer had changed and we had a difficult time debugging our programs. The room contained a printer that we could use, however, the printer contained a poltergeist and would start printing at random times when neither of us was using it and noone else was in the building. That was scary.

A 65K 48-bit word CDC3600 with the transistors and with both tape and disk storage passed its benchmark test. The disks were in stacks permanently mounted and were about 18 inches in diameter.

About this time Fed Laccabue wrote a Fortran manual for the CDC3600. It was quite possibly the best manual ever. He had managed to incorporate so much humor and clever use of Latin, French and German in examples that we read it carefully, afraid to miss the joke. Unfortunately after enough people had proofed and edited it, the published sanitized version had lost the charm.

Computer Room Layout

With the CDC3600 satisfactorily benchmarked and about to be delivered and the plan for acquiring another IBM computer, the plan for a computer layout became critical. A room in the Applied Math building was designed for computers with cabling space and freon cooling under raised flooring. It was huge - maybe 150 by 100 feet. Each computer consisted of a main (console) cabinet and several other cabinets. The computer operator needed to be able to reach the console, the tape units, the printers and the readers and sometimes a teletype machine. The maintenance people need to be able to open and work in all cabinets. Cabinets were generally about seven feet high, thirty inches deep and varied in width from about three feet for tape readers to about six feet. At least the wide sides had access doors swinging out about 30 inches. Between the IBM360-50, IBM360-75, CDC3600, CDC160A, Digital Equipment computer, a homemade parallel processing computer (Neurotron) and the planned IBM360-192J there would be more than 50 cabinets to be arranged so that there was access space, power and cabling. The center manager seemed overwhelmed by the task and drew several non-workable plans. I made cardboard shaped modkup of each cabinet with its access doors open and cabling requirements written in. I had fun designing a good plan with cables as short as I could get. Long cables slowed the computers and lessened reliability. The noise level in the room was dangerous so we had big sound protector earmuffs we could wear. When we added some graphics equipment, it had to be installed in a room across the hall. The finished rooms was like a maze with cabinets tall and close together. One could get confused, lost and claustrophobic when between all the tall cabinets. To have a little fun and help that situation, I made cartoon floor map drawings with "you are here" arrows like you might see in a zoo. I taped them to a number of the cabinets around the area.

Analog computing

George, our large analog computer had a separate room. The most interesting project I remember on it was by Authur Roberts. He programmed it to produce music. He included overtones, attack sounds, hesitations, acceleration, crescendo, fading and vibrato. Instruments such as French horn clarinet oboe, cornet tympani piano, violin and cello each had their characteristic sound. He composed fugues in quarter tone that only took a short time to get used to. He created infinite ascending and descending scales. He experimented with the effect of various non standard rhythms and scales. His results were magnificent.

Another IBM Computer Added

Then came a 2048K 32-bit word IBM370-192J. Called a multiprocessor, it could handle more than one program at a time. We set it up for a maximum of seven programs simultaneously sharing the resources. To make the sharing more efficient we asked everyone to class their work as compute bound, input/output bound, or mixed We also asked each for their core requirement so were could combine programs to share space and resources. The most frequent sizes were 140K, 256K, and 800K (which usurped the computer)

Because we had people who were greedy and people who were uninformed, I was asked to write a spy program to find out what the users were really using. If I found something excessive and I judged that the user was an uninformed, I fixed their program for them – usually this was very easy. If they were among the greedy, I left them a note. Another reason for this spying was to try to determine how the cost of the computer should be shared. The sharing algorithm would be designed reward those that play together well with tight or well-defined requirements. The information I gleaned from spying was used to create a charging algorithm that we hoped was fair and rewarded the efficient users.

.

Success / Failure

It wasn't long before this computer, running all week except for maintenance, was hopelessly backlogged, so I was asked, using a representative job stream, to simulate the effect of three possible fixes – increasing computer speed, making the core capacity larger or increasing the input/output resources – and estimate the cost/benefit of each

San Francisco Meeting

I had been coerced into writing an abstract for a speech to be presented May 3-4, 1973 in San Francisco at the Atomic Energy Commission Scientific Computer Information Exchange annual meeting. After a struggle because I didn't have any appropriate project completed, I submitted the following abstract on the project that I was just beginning.

ABSTRACT

A mathematical probability model is constructed to predict the behavior of a multiprogramming computer having several time independent resources under the demands of concurrent jobs for those resources. Parameters for a particular application of this model were obtained from the analysis of Argonne National Laboratory's job stream on the IBM 360/75. Prediction of the behavior of that job stream on the IBM 360/195 from the model gives estimates of concurrent CPU utilization and channel usage as a function of job CPU boundedness and as a function of job size. The overall resource use and production time are predicted. The success of the prediction is evaluated by comparison to the behavior of the actual job stream on the object computer. Permutations of the parameters of the model show how cost appraisal of various changes in hardware can be performed

.

. Once the abstract was submitted, I was under pressure to produce the program and results to report. I wrote a simulation program that could incorporate these three fixes (increased speed, size, or I/O) separately or in combination. I produced some nice predictions and graphs. When the predetermined job stream was run under a selected new fixes, my simulations proved to be fairly accurate.

Next we told the authors of the job stream programs to submit their tasks for testing. We quickly discovered that although my predictions were theoretically sound they were not practically realistic. When we told users that they would have twice as much room to share, instead of sharing better, they made their arrays bigger and used up the added space. If given more compute power, they decided that six digits of accuracy were needed where before four would do. With more output power, they printed out more details and added graphic output. The net result was that none of these fixes would work. What would work was getting more computers

I reached this conclusion shortly before I was to give the AEC presentation. I panicked – how could I give a presentation on failure. Luckily a friend statistician Sylvanus Tyler, said "why not, most research experiments fail." I set out to write a presentation that would show my assumptions, simulations and conclusions. The prepared speech was full of probability equations, utilization graphs, turn around time graphs for various types of tasks and cost of this time-decrease. I planned a beautiful, erudite presentation.

I was scheduled to talk on the second day, so on the first day I handed in my slides and just listened. From the presentation given and the questions asked, I realized that my presentation, as prepared, would go above the heads of most of the audience. That night I got a bunch of four by six cards and rewrote my lecture. I had to make it match the slides I had already handed in. My revision concentrated more on the reasons for the simulation, what we hoped to accomplish, how well we did and how we would vary the experiment if we were to do it over.

I got up in front of the group put my cards down on the lectern and when they lowered the lighting to show the slides, I could not read my cards at all. I guess adrenaline took over and I managed. At the end of the program the master of ceremonies got up and commented on the fact that there was only one speech that truly represented a research project and that most of the others were just bragging about their accomplishments. Mine was that one.

Data Processing Management Association

George Robinson and I were among those who took the first CDP (Certified Data Processor) examination. I figured I had a good chance of scoring well since I knew many different computer languages and those included all the popular ones at the time with the exception of COBOL (common business oriented language). George and I scored high enough that, when the DPMA group wanted to revise and update their examination in 1966, we were asked to take and review the new exam. It seemed a good exam with four subtests: Hardware, Software, Systems, and Quantitative methods. My percentile ranking on the revised test was at the 96th percentile. A total of 1005 candidates sat for that test and 408 of them passed.

Cyclical Components of Respiratory Flow

In 1964 John Henry of Hines Hospital recorded the flow from patients – normal, anginal and obese. I used the Tukey Fourier Series spectrum analysis program that I had written back in 1958 while working at Convair to compute the power spectral density of flow in each of the subjects to study the low frequency components of Cheyne Stokes respiration compared to normal (eupneic) breathers. The results showed precise regulation of flow in normals and an almost random variation in flow in Cheyne Stokes subjects.

IBM versus CDC

The attitude of the IBM and Control Data companies was quite different. Control Data did not try to control the user. If one thought of a reason for using the computer in a way not described in the manual, the computer would allow you to do it and possibly to get into trouble.

Minnesota Aptitude/Interest Testing

The Minnesota company that does all the neat testing of interest and aptitude had a little note in some scientific magazine indicating that they were trying to get their tests to include better criterion on female scientists. That intrigued me so I called the company and offered to get a group of female scientists from Argonne and a parallel group of male scientists to participate, provided they would share what they learned with us. Within the Math group I recruited five female and five male programmers. When the results came back all five females were categorized as having very masculine interests. It was interesting for us to learn how similar our female interests were and how different they were from the male subjects

.

Argonne - Working for Jim Cody

Jim Cody was an ideal leader for a group of self-motivated workers. He hung a sign up sheet on the window of his office door. On it were listed the tasks that needed to be done. Next to the tasks were columns for initials. The first was column was for initials of anyone who was interested in the task. Column two was for the initials of whoever started that task. Then there were columns for various stages of progress and finally for completion. We were only allowed to put our initials on three of the tasks, so we worked to get tasks completed before someone swiped the task we wanted to do next.

System Function Subroutines

For scientific work on computers one needs a set of subroutines to evaluate functions. These were needed both for real and for complex (real + imaginary) values and for numbers in single and double precision. Each of these can be evaluated as the sum of an infinite polynomial series. Obviously infinite is not practicable, so my group leader, Jim Cody, and I determined rational fractions to evaluate each and programmed a set of function subroutines. It took me several months to produce this set of assembly language function subroutines.

The subroutines included the trigonometric functions (sine, cosine, tangent, cotangent, arcsine, arccosine, arctangent, arccotangent), absolute value, square root, cube root, logarithm and exponentials (x**y), pseudorandom number, Bessel Function, gamma function, and elliptic integral.

My power function (x**y) that was calculated as 2**y*$\log_2$(x) (computers are fond of the number 2) was more accurate and faster than any previously devised, so I was asked to present it at the SHARE convention in Los Vegas. We created fifteen slides and Jim made me rehearse until I was fairly comfortable. Still it was quite an adventure speaking in front of a large group (maybe 1000 people) in the Las Vegas MGM Grand with a speaker system and slides repeated on seven screens.

One of the demonstrations of the improved accuracy of my power function was a three dimensional pin map where the logarithm of the error determined the height of the pin for random x and y. The map in the original routine looked like two orthogonal valleys intersecting at zero x and y. The pins showing the errors resulting from my subroutine were much lower and had no definite pattern.

Inductive Proof

During 1963 I was the first to write a computer program to prove an inductive theorem, i.e. if something is true for ***i,*** it is true for ***i+j***. This particular theorem was: there exists a closed table with any given number of associative laws (a(bc)=(ab)c) true. The proof by induction is unusual in that the inductive step is easier than establishing the result for the initial values of ***i***. For this particular theorem the ***j*** was 4 and the initial ***i*** was 3. For ***i***=3,4,5,6,and 7 proof by example was needed. For the seventh there are about 100,000,000,000,000 possible tables. Even if one could define and examine one table per millisecond (roughly 100,000,000 per day) it would take 10,000,000 days (about 3000 years) of error free computer time to examine them all. What would one do with all those results! The number of possible tables is so large that to examine all of them would be prohibitive. I began by programming to examine each table in sequence, eliminating reflections and rotations so there would not be quite so many. After examining several thousand, I printed out the frequency in which each number of associative laws was true and looked at the unusual ones. Then I biased my selection of which tables to examine and eventually found an example of tables with zero up to 343 associative laws true. At that point the inductive proof took off to prove the existence of tables from 344 to infinity.

The theorem was published by Dr Herbert Wilf of the University of Pennsylvania. In his acknowledgement of my help he wrote, “ Nancy Clark shattered all records for efficiency, as well as having some bright ideas for doing the calculations better”.

Circulants

The title for this project is *General Root-Powering Methods For Finding The Zeros Of Polyomials.* For this one had to multiply synthetic polynomials in seven variables in every combination up to the power seven, sum several of these and then divide the result by the product of another pair of polynomials of the same size to show that the result was always an integer. To do this every coefficient had to be stored as an integer or rational fraction, never as a decimal fraction. It was probably one of the hairiest and sneakiest programs I ever wrote. The equation to be solved was:

$$A_{ij}^{(k)} = (A_{kk}^{(k-1)} A_{ij}^{(k-1)} - A_{kj}^{(k-1)} A_{ki}^{(k-1)}) / A_{k-1,k-1}^{(k-2)}$$

Where A is seven by seven matrix where each element is a polynomial in seven variables, with the powers i, j, and k taking on all the numbers from zero to seven. Solving it took seven hours on a CDC3600 with the products and sums occupying much of some huge hard disks. It was included in a publication by Bareiss in 1967.

Integers

In real life, not just in quantum physics, many things have to occur in integers. The most frequently used example is an egg, where a fraction of an egg is not an egg. Many things happen only in integers. A gear will not work unless it has an integral number of teeth. Many sporting events and games are scored as integral numbers of some event: a strike, a base hit, a goal, a touchdown, three no-trump, etc. Is half of a ball still a ball? Several of the problems which I worked on had to have integer solutions. Computers with their fraction and exponent notation are not good at integer arithmetic. The trouble with computers and decimal fraction is illustrated by the observation that (1/3)*3 usually is evaluated as .999999, not as one. I wrote software to aid in handling this type of problem.

Uranium and Osteoporosis

I was walking down the hall in the Physics building one day when I saw several women sitting in wheelchairs. They had been employed by Elgin Watch Company to paint watch faces. No one had told them of the dangers – perhaps no one knew. Their habit was to lick the brushes to point them. They were in very bad shape with fractures and tumors. As I stopped to talk to them, someone came out to get them and recruited me to help. I was then asked if I would be willing to be in a control group – I was close to the same age as these women.

They wanted to measure my bone density once every five years. They measured the density using an Iodine 131 source above and a recorder below a tub of water in which my arm was stabilized. They measure three places: a finger and a wrist and a mid arm area. I was at the top of the normal bone density scale.

Alvin

One short but fascinating assignment was to a brainstorming session on the design and building of the Woods Hole two man mini deep sea submersible Alvin. I was probably assigned to that project because I had the highest security clearance of the math/computer staff.

Space Navigation

About 1962/63 Jim Cody showed me an article indicating that the space program was being held up because no one had been able to solve the differential equations of space flight accurately. The problem is known as the three-body problem. Three bodies are assumed to attract each other by the inverse square force, but one of the bodies (e.g. the space ship) is considered small enough that its influence on the two massive bodies (e.g. earth and moon) can be neglected. Just to give an idea of the complexity, here are the two equations of motion. .

$$x'' = x+2y'-(1-u)(x+u) / ((x+u)^2 + y^2)^{3/2} - u(x-1+u) / ((x-1+u)^2 + y^2)^{3/2}$$
$$y'' = y-2x'-(1-u)(y/((x+u)^2+y^2)^{3/2} - u\, y/((x-1+u)^2+y^2)^{3/2}$$

I asked for time to try to solve this accuracy problem and, about three months later, I mailed a program to NASA that could navigate to a theoretical twenty-two decimal digits of accuracy. I proved its accuracy by theoretically guiding a ship using three Aarnsdorf "perfect" orbits. That program went on the first unmanned ship to the moon and later was used to guide ships out toward other planets on Pioneer, Voyager and Galileo.

The day after the first unmanned ship landed on the moon, I got a call from the University of Minsk. They wanted to know if they could use my navigation technique on their ships and wanted me to visit Russia as a consultant. I told them to call me back in two days. Meanwhile I asked if there were security problems. They said that since I had not been informed of any secrecy, I could do as I liked. I told the Russians where they could get a copy of my Fortran program. I wasn't at all comfortable with the idea of going to Russia at that time. Based on some phone consultations I had with the people at the University of Minsk, I believe the Russians also used my navigation program.

I showed that the rational extrapolation technique I used applied to many other sets of initial value differential equations and I wrote a paper comparing my method to three other methods in common use and demonstrated that, whenever more than a few digits of accuracy were needed, my technique was better and faster. My work on this was included in a Princeton textbook on numerical methods for initial boundary problems. I recently learned that my computer specification and guidance programming were still in use although undoubtedly smaller and faster.

University of Michigan Seminar

In the fall of 1969 I was invited to present a seminar on my work with numerical solution of differential equations as part of their Distinguished Lecture Series at the University of Michigan in East Lansing Michigan. The lecture and discussion were well received and I did not realize my idiosyncratic behavior until the whole thing was over. I had my notes on index cards and, when I finished a topic, I apparently threw the card. I picked them up afterward from all over the area. Maybe it added the touch of humor often needed during that type of presentation.

Reactor Rods

One was assigned to create a protocol for testing the rods in a reactor. There were three considerations: how frequently rods should be tested, how many should be tested each time, and how are the tested rods to be selected. What I gave them was a 99.9 percentile that involved testing about five rods every month. I also designed a selection method which required selection at least one rod from within the array. Otherwise it would be natural for the workers to test rods in the outside row.

Argonne – working for Frigerio

Norman Frigerio was quite a character. He had a Ph.D. in chemistry from MIT, a MD from Harvard, was an Eastern Rite Catholic priest, was married and had nine children - all boys. He had a broad forehead with prominent brow, large jaw, long beard, a pleasant smile, supercharged energy and a dangerously high metabolism. He told me that he felt as though the rest of the world were swimming in molasses. I replied that I understood that because I was looking out of the molasses. He seemed to have very little concern over the safety of his work, so there was a saying: ‘If there is a warning siren, look for Norm and run the other way’.

Norm believed that his group worked better together. His working group gathered at 6:15PM in the cafeteria and moved tables together until we could all spread out our stuff and work together. A number of projects went on at once with questions and brainstorming across the group. The most difficult part of working in the group was going home. Everyone was so wound up in the work that Norm would have to declare an everybody-out time at about 2AM.

Visualization in 3-D

In 1965 Oak Ridge National Laboratory published a Fortran program to produce stereoptic crystal illustrations with a mechanical plotter. They did not tackle the hidden line problem so the stereoptic pairs of perspective projections were touched-up by hand to correct for overlap. In 1973 I rewrote and added to this program in the PL1 language to allow production of the images on the 35-mm IBM 2280 Film recorder or onto a cathode ray tube. To aid in visualization we made both perspective stereoptic pairs and rotating perspective images. Several of us learned to focus one eye on the right image and the other on the left image. We learned quickly that was not good for our eyes.

Among the objects pictured were the crystal structure of p-toluene sulfonic acid and the force field produced as a hydrogen atom approached a pair of lithium atoms. For some of this work we used a language called Speakeasy (by Phillip Cohen) that was great for scientific work because one could declare something a matrix, an array or a vector and the treat it as if it were a single variable. Phillip had built in a method for changing the language on the fly. The command HENCEFORTH would allow the user the option of creating their own commands or changing existing ones.

.

Spine

To study the motion of the normal and abnormal spine, we started by describing the fifth lumbar vertebra using line segments on buttock sections as if the vertebra had been sliced. My son, Tom did the slicing and measuring. By the time we had segments describing the vertebra parts – the body, pedicles, transverse processes, mammillary processes, laminae, superior facet, inferior faced and spinous process – we had a about eight thousand segments for just the single vertebra. Using the 3-D and stereoptic program we produced stereo and rotating views of that vertebra and began to build an array for the sacral base below it and the lumbar vertebrae above it. We could add flexion, extension, twisting and lateral tilt to the positioning. We had just reached the point of producing a really useful tool when the funding for the project was not renewed. For many years I kept the interrupted work ready to pick it back up if funding came through.

Random Numbers

In scientific research (or in Monte Carlo or other games) one frequently needs something to happen randomly. Since randomness is not natural to computers and we cannot flip a coin within a program, programmers have sought a method for producing pseudo-random numbers. In 1967 I wrote a program to chi-square test the randomness of a dozen pseudorandom numbers generators in use. I tested the distribution of the numbers produced in one, two and three dimensions, and the length of ascending and descending series. In all series the last digit of the numbers were rotations of 1,3,5,7,9.

Two of the favorites have been X_{i+1} = (X_i * 5**15) mod 2**24 or X_{i+1} = (X_i * 3**29) mod 2**24. The initial value of x must be odd and so will all the other values of X_{i+n}. Since at that time most users needed only a small (less than 2000) set of the numbers many users never noticed problems, but some users were affected radically. My first try at remedying this situation was to change the progression to X_{i+1} = ((X_i * 5**15) mod 2**36) / 2**12. This allowed much larger series before one could detect a pattern in the low order digits.

All the random numbers produced with this type of algorithm lay on the vertices of a crystal in n-space. Another way of stating the limitations is that all numbers generated lay on a small number of hyperplanes. My fix only increased the number of these planes.

Still there were some applications where one needed a more truly random sequence. To obtain a suitable series of numbers we took a sample of uranium235, stabilized it in methane and counted the alpha emissions over time. For each time interval we recorded only one bit - whether the count was odd or even. We accumulated 93 million of these bits on magnetic tape for anyone to use as random flips of a coin or in groups as random integers. For this project a computer called a Neurotron was dedicated for several weeks.

Neurotron

The Neurotron was a small fast (for the time) computer capable of up to 128 parallel computing paths. It stood in a corner of the football field sized computer room with a methane tank next to it and a chamber in which gas bubbled around a sample of uranium 235.

One day when I checked on progress I noticed that there were no bubbles of methane stabilizing the uranium. I panicked and left the computer room and ran to the Physics building looking for someone to help. The person I found did not have clearance to enter the area, so I yelled that I was bringing someone with me tight thru the card controlled security door. Some alarms went off but we made it back to where he kicked the methane tank and everything was working again.

Planning for Safety and Emergency in Northern Illinois

A 399 step Fortran program I wrote in 1975 involved safety, especially the concerns for safe siting of power stations, but also applicable to other problems such as tornados, floods, water contamination, earthquakes, etc. In the original study the sample area extended from thirty-two miles outside northern Illinois into Wisconsin down to thirty-two miles south of route 80 and from 32 miles west of the Mississippi to 32 miles into Indiana. We gathered information for each of 20,000 two kilometer square grid points indicating population of humans, population of cows, dairies, temperature means and extremes, heat islands, wind velocity and direction, rainfall, water ways, water supplies, fossil fuels, energy sources, transportation, evacuation possibilities, endangered animals, endangered plants, tornadoes alleys, recreation areas, natural preserves, seismic fault lines, waste disposal facilities, land prices, etc.

Gathering all the data must have been a humongous task. For each set of data I produced a computer generated 35-mm film map of northern Illinois showing the least endangered areas as clear, the most endangered as black and shades of gray logarithmically assigned to values between. Each 35mm film took .5 minutes to produce on the IBM370/195. For different kinds of potential disasters we could produce a combined map by computer or just overlay one transparency over another to define relatively ”safe” clear areas and the dark endangered areas and could prepare to avoid problems.

Studying Anatomy (1974)

I was only assigned to work for Norman Frigerio for four hours per week. I asked Norm what I could do to become more valuable to him. He suggested that I take an anatomy class at the Chiropractic College. I questioned this because Argonne had associations with the University of Chicago and the University of Illinois. His response was that the Chiropractic College had the best anatomy class. When the Chiropractic College dean of students was reluctant to allow me to take a class without being an accredited student, Norm called the college and talked with the president, Dr. Janse. When I took daytime classes is was convenient that Norm liked to work at night

The End of Frigerio

Norm was also an excommunicated Catholic priest. One Sunday, with his wife and children, I attended his High Mass with Gregorian chant using the breakfast table in the kitchen of his farmhouse as an altar. There was a beautiful combination of chant, ritual and simplicity. The period of time in which he was excommunicated was not long, but unfortunately soon thereafter he died suddenly of ruptured aortic aneurysm or cardiac tamponade.

An effort was made to replace him with other scientists. It failed miserably and his group was disbanded; the individuals assigned to other projects.

Argonne – Working for Louis

Government squeeze had eliminated most of the interesting projects. Unfortunately for me, this left me working full time for the group leader who assigned a task to me that was uninteresting, impossible and unnecessary and for which my talents were not suited. He had already wasted six man-years of good people's time on it until two quit when their request to be transferred was ignored. The project was aptly named CRAP (computerized resource accounting program). In response to a request for a progress report I wrote, "On this CRAP project I have done nothing of which I am proud". He wanted me to work on it 40 hours per week. My response was to call the payroll department and reduce my hours and proportionate salary to twenty-eight hours per week and take a half-day vacation every week so that I could work three days per week and continue taking classes.. I continued the classes and he fired me with two years notice effective in August of 1978. Our great secretary was not happy with typing the pink slip, so she kept the original handwritten one. Two years later the project still was not fully functional. However it pleased some people by producing attractive looking reports that seemed to me to be virtually meaningless bureaucratic garbage

Concerned Scientists

About once a month a group of employees got together to discuss and to see if they could help with societal problems: problem pregnancy, care of mental patients, wrongful incarceration, etc. One campaign was to allow use of vitamins in mental hospitals. A friend had experienced the difference that vitamins made in the behavior of her schizophrenic son I don't know whether we actually accomplished anything, but I learned a lot. Later I noticed that there was a correlation between membership and layoff.

Mensa, Triple 9, Four Sigma

Three high IQ clubs were open to qualified members. Mensa accepted anyone in the top 2%. Triple nine accepted those in the top 0.1%. While the most elite, Four Sigma, took those four standard deviations above the mean, I think that was the top 0.03%

I took three IQ tests. One of them called the hardest IQ test in the world was intended to spread the top percentage areas of the other IQ tests. I had been feeling sick through much of 1973 so I was curious as to the mental effect of illness. I scored 142 on one of them and 165 on another. The third gave results in percentiles and I was ranked at the 99.93 percentile. That was not quite high enough to qualify for Four Sigma, so they sent me a letter of regret and forwarded my score to the Triple Nine group. In Mensa I enjoyed the duplicate bridge group, the annual HalloweeM party and lectures at their monthly meetings.

.

Continuing Education

Argonne arranged with the University of Illinois for interested employees to work toward Masters Degrees in Mathematics. A math professor came from the University of Illinois to teach a postgraduate 3-credit hour mathematics class. I took Advanced Calculus, Complex Variables, Modern Higher Algebra, Fourier series and Boundary Value Problems, and Partial Differential Equations,

My semester hours would have been enough for a master's degree in Math except that U of I rules require that there be a set of two sequential courses. My attempt was frustrated when the University closed the Navy Pier Campus, opened its Circle campus, changed from semesters to quarters and rewrote the master's requirements and dropped the extension program. Five of my papers which could have been made into theses: "Inductive proof by computer," "Exact computer arithmetic," "Use of rational fractions in function evaluation," "An accurate method for numerical solution of differential equation," and "Statistical critique of pseudo random number generators."

I spent the summer school session in 1966 at MIT taking a concentrated Computer Aided Design class that concentrated on airplane and ship design. With other evening and weekend classes I earned a bachelors in Biology

Do It Yourself Computer

In the mid 60's I bought some of Edmund Berkeley's books (Edmund Scientific, Edmund Enterprises and later American Science stores). Computers were not household toys yet. With this we learned to build circuits such as Boolean "and" or "or" circuits. My son and I built a simple computer using simple materials: pegboard, flashlight batteries, Christmas light bulbs, sliding contact switches, washers, nuts and bolts, paper clips, etc. The main problem we had with it was the wattage to the lights, frequently too much and they burned out or too little and they failed to glow.

Chiropractic College

For four years ending in August of 1978, I attended classes for twelve trimesters at the Chiropractic College while working 28 hours per week at Argonne. Just for the challenge of it, I took the South Dakota Medical boards and passed them when I was half way through. I amazed the board, they had a physician's license ready for me, when I told them that I did not have an appropriate degree. After graduation I continued on a three-year weekend program leading to a radiology/imaging certification. I completed the class work, but not the residency. My practice over the next fifteen years consisted of gentle treatment using acupressure, trigger point, toftness touch technique and applied kineseology.

Omitted here is a sizable description of my practice and patients

A Computer for the Office

In 1982 Pat, a patient/friend who worked for Heathkit. Worked with me to build a Heathkit computer... It was big enough to put together with a soldering iron. It had a weird amount of memory 256 bytes ram plus an 11K, disk and a floppy drive using 8 inch floppies. Its addressing scheme was not upward compatible. We built it in my office waiting room with patients hanging around to watch. Its Peach ware software was later on the IBM 286, 386, 486. But it was not supported for long.by IBM. Interestingly it will still work from a floppy but cannot be loaded into computer memory.

Narrative Report Writer

Most doctors hate to write narrative reports - so. Using Peachtext I began developing report writing rsoftware that asked an appropriate tree of questions and then produced a narrative ready for editing, if necessary, It had two main paths: one appropriate for auto accidents and one for work accidents. It collected data such as physician name, patient name, date of birth date of accident, date of examination, date of accident and addressee. It tpresented choices such as male/female, right/left, severe/moderate/mild or continual/ intermittent /occasional. Peachtext was an incredibly good program. It let one in clude Fortran like – IF ,THEN, GOTO decisions and had extensive macro capanbilties Even raw, the narrative reports I produced were better than most doctors were writing. Later when I took the training and passed the examination to become a Certified Disability Evaluator for the state, I incorporated some of that legal verbiage.

Senior Olympics

On reaching the ripe old ago of 55 in 1983, I was recovering slowly from a blood clto in my thigh. In order to stimulate my drive to regain strength, I signed up for the Illinois State

Senior Olympics to swim 200m backstroke and 100m breaststroke. That was the beginning of my participation in the state games for many years including participating in the nationals in St Louis and in Syracuse.

Computerizing the Senior Center

After my sister, Ginger, died in 1995 I became interested in the park Ridge Senior Center. There I began a group to teach seniors to use computers and their text and spreadsheet programs. I also started moving toward computerizing the center's records, mailings and newsletter

Genealogy

My sister had collected lots of family information so after she died in 1995 I entered her data in the Later Day Saints program.PAF and a year later into Family Tree Maker FTM. Over the next years I made a few contributions to the FTM program as it grew. I finally traced my paternal Welty tree and my maternal female side Tolen-Mills-Demerly family tree back about 300 years with and found a total of 5130 family members. Some of the relatives were found using Y-DNA

Developing an Image Filing System

As the medical profession obtained a growing number of digital images, Doctors and researchers needed a way to file, retrieve and cross reference them. My son and I developed a system for naming image files using an eight-character code that included a date code, a patient ID and a code indicating the pathology or area of interest (these corresponded to the general ICDA (International Classification of Disease) medical codes. An individual patient might have several identifiers, one for each area of patient pathology. With this, for instance, it was easy to pick out patients with knee images. This proved important as we looked for examples for workbooks we wrote on musculoskeletal ultrasound diagnosis of joint pathologies.

Changing Horses.

In 1990 I was winding down my practice and getting ready to retire and my son Tom was getting enthusiastic about diagnostic ultrasound. He accumulated shoulder and knee images wrote a shoulder manual and a knee ultrasound manual and began teaching weekend classes At that time I started to get interested in PC's and in the internet. I traded some of my expertise and work (that I thought of as fun) with an oriental named John for a PC. I got involved in early web site design language and after a while I designed a website for my son. It was called Sketalus. It no longer exists. His revised site now is MSKUS.

Tom's Ultrasound Publications

Using WORD, POWERPOINT and PHOTOSHOP with Tom's case histories, examinations and images I began formatting slides for my son's classes. Tom and I got a system. Tom attached ultrasound images to emails with comment on their contents. I cropped sized added background text and arrows and attached slides to a return email. Sometimes there were several iterations before the slides were good.. By now we have over 4000 slides ready for use in convention presentations, classes and illustrations in texts. Recently we have produced training DVD's. Subjects include Ultrasound physics, knobology, technique, protocols, and nomal/pathological anatomical areas: shoulder, elbow, wrist, hand, knee, ankle, foot, occitput, cervical spine, thoracic spine, lumbar spine, hip/ pelvis, and needling and stem cells. Presentations have been held in Australia, Netherland, Chile, Argentina, Columbia, Hawaii, India, Hong Kong, Saigon, Italy, Germany, Dubai, Spain, Turkey, Mexico, Canada, and many places in the United States.

May 13, 2012
Nancy Welty Clark
(April 8 1928 -)
drnwcdc@aol.com