An interview with
PAUL N. SWARZTRAUBER

Conducted by Thomas Haigh
On
16 & 17 July, 2005
Westminster, Colorado

Interview conducted by the Society for Industrial and Applied Mathematics, as part of grant # DE-FG02-01ER25547 awarded by the US Department of Energy.

Transcript and original tapes donated to the Computer History Museum by the Society for Industrial and Applied Mathematics

ABSTRACT

Paul N. Swarztrauber discusses his career to date. Swarztrauber grew up in Illinois and gained a B.S. in Engineering Physics from the University of Illinois in 1959. During his studies he used the ILLIAC computer and studied mathematical computing under Lloyd Fosdick. After graduation Swarztrauber spent three years in the U.S. Air Force, and he discusses his work at Kirtland Air Force Base in New Mexico and its IBM 704 computing environment. In 1963, he went to work as a scientific programmer at the embryonic National Center for Atmospheric Research in Boulder, Colorado. Swarztrauber discusses the initial computing environment at NCAR and its development over the years, including the CDC 3600 and 6600 computers, the Cray 1 and Cray 3, the center's experience with its Connection Machine, and its ill-fated attempts to procure an NEC SX-4 machine. Concurrently with his programming work, he enrolled in the Ph.D. program of the University of Colorado, Boulder, earning his Ph.D. in 1970 under the direction of Robert Richtmyer. This led to a gradual shift of responsibilities at NCAR from assisting researchers with application programming work toward mathematical research. He has published over sixty refereed papers in areas including elliptic, parabolic, and hyperbolic partial differential equations, computational fluid dynamics, linear algebra, parallel computation and communication algorithms, computational aspects of weather and climate modeling, vector and scalar harmonic transforms, the fast Fourier transform, signal processing, multi-computer design, Gauss quadrature, and singular integration. He is also the author of several large scientific software packages that are in general use throughout the scientific community. Swarztrauber discusses his role as author or co-author of three significant packages, including for each their relationship to NCAR's needs, related original mathematical research, their origins, collaborators, source of algorithms, documentation, distribution methods, relations with users, testing procedures and development through multiple revisions. These were FISHPACK (with Roland Sweet and John C. Adams) for the approximate solution of separable elliptic partial differential equations, FFTPACK for the fast Fourier transform of periodic and other symmetric sequences, and SPHEREPACK (with John C. Adams) for the modeling of geophysical processes using spherical coordinates. Swarztrauber also discusses his work on the CHAMMP project to evaluate scientific applications for the massively parallel machines during the late 1980s, and his resulting attempts to conceptualize and design a new architecture for multiprocessor units he calls the "communication machine." Swarztrauber was involved with ACM SIGNUM from the 1960s onward and with SIAM from the 1970s, including work as a conference organizer, as a lecturer, as a member of its council, and as an editor for several journals. He retired from NCAR in 2004 where he continues to hold the position of Senior Scientist Emeritus.

HAIGH: Thank you very much for agreeing to take part in the interview. I wonder if you could begin by saying a few things about your family life and family background.

SWARZTRAUBER: I grew up in Zion, Illinois, which is about forty miles north of Chicago. It was a small town, maybe 2,000 people. Nobody ever locked doors. I was free to ride my bike down to Lake Michigan and go swimming. It was a very safe and secure environment, and the things that people are concerned with these days just didn't exist in those days.

In my early childhood, I took everything apart. It would drive my parents' nuts. Anything I could get my hands on, I would just take apart. When tape recorders first came out, my father bought this beautiful tape recorder. It had what were called electrostatic speakers in it. I just wanted to take a look at those speakers and started taking it apart. I just didn't stop. I took the whole thing apart. The guts of it were spread around the living room floor. When my mother returned from shopping she almost fainted, because this was Dad's new toy. I got it put back together; however, my dad could hear a slight warble in the recorder and asked me, "Do you hear that?" Of course I said "It sounds fine to me." But that was just an example. I took apart my motor scooter and later my car, including the engine, everything and anything I could lay my hands on. That would probably be characteristic of my youth. I just had an insatiable desire to see how things worked and how they were put together.

A lot was expected of me in high school. Both my father and older brother were valedictorians and my sister was salutatorian. A lot of family has a history of very good students; however, I must say I was not a good student at all. I graduated right in the middle of my high school class. But I do remember one critical thing. I took a course in physics when I was a senior, and read the entire book in the first couple of study halls. I loved it, really enjoyed it. Of course, I did well in that course. That was my introduction to science, and probably the first indication that science was going to be in my future.

HAIGH: So before taking that course, you were really more interested in taking things apart and tinkering with them than with academic studies.

SWARZTRAUBER: Well yes, but I also wanted to know how things worked. I was probably in sixth or seventh grade when I realized I didn't understand radio. So I got a beginners book on radio [Abraham Marcus, Elements of Radio, Prentice Hall, NY, 1943]. Of course, it was all vacuum tubes at that time. Later I picked up a Heathkit short wave radio kit that consists of a box of resistors, capacitors, and so forth and had the parts spread out on the ping-pong table. My father's comment was "he will never get that working." But I kept at it and when I invited the family to listen to radio Moscow my father was horrified at the possibility that his son could become a communist.

HAIGH: Did your family have a tradition of people going to college?

SWARZTRAUBER: Actually, my parents did not attend a traditional college. My father attended one year of Bible College and my mother attended three years of the same Bible College. Her father was a medical doctor who joined a cult leader in the establishment of Zion, Illinois where I was born. "Dr" Dowie was the cult leader who established quite a following through his healing ministry. Indeed he "healed" my grandmother, over the radio no less. This clearly impressed my grandfather who then moved to Zion to work for Dowie. It's a rather sad story because he had to give up medicine to join the “church”. My brother is a retired Admiral and has written a thousand page book entitled "Schwarztrauber, Stewart and Related Families", by Sayre Archie Schwarztrauber, Gateway Press Inc., Baltimore, 1993. Note that our father changed the family name from Schwarztrauer to Swarztrauber and my brother changed it back[1].

When Zion was first established, the whole town would kneel and say a prayer when the bell rang at noon. This was around the turn of the century. Of course, there was a lot of beer traffic between Milwaukee and Chicago, and the Zion people would stop that traffic and spill the beer. I have seen pictures of that, and also pictures of the original Dowie Tabernacle with a massive number of his followers. Evidently he went nuts and his followers went their different ways with sub followings and the establishment of traditional churches. I grew up in the Pentacostal church, which is a book in itself.

HAIGH: You were saying that your father had gone to Bible school?

SWARZTRAUBER: Yes, that’s where my folks met. But that was about the extent of his education. He was interested in accounting, and worked for an outfit in Waukegan [Paul Pettengill & Co.]. He got a CPA at night school and established his own business in both Waukegan and Chicago. He did very well. The Waukegan office still carries his name. But in terms of schooling, it was minimal. But they were very bright. My aunt Vera was a Christian missionary to the Middle East! My other aunt Evelyn had a good job in the Foreign Service and traveled the world over. They were complete opposites, philosophically.

HAIGH: So as you approached graduation from high school, you said that you were pretty much in the middle of your class. You had discovered this interest in physics, and you knew that you liked tinkering with technology.

SWARZTRAUBER: There was no question about that. But I had set my eye on electrical engineering. I didn’t think about being a mathematician at the time. That would have seemed beyond my reach and my capabilities as an average high school student. I went to the University of Illinois. It was cheap. At that time I think my tuition was sixty dollars. It also had a very good reputation, particularly in electrical engineering. So that was basically the appeal, a good education at a very reasonable price.

HAIGH: So was that Urbana Champaign campus?

SWARZTRAUBER: Yes. There was some talk that I might go to IIT in Chicago but the campus seemed a little bleak at that time and a couple of my high school friends were going to U of I so that’s were I ended up. Of course the first few years were just sort of general courses. Actually, I have an interesting story to tell. At that time, the University of Illinois, gave entrance exams to determine your level. Well, I flunked both the english and mathematics entrance exams! So I had to take remedial courses in both. That really got my attention, and I became a serious student. I was on the honor roll the second year. I did very well in math and physics but was still not fond of english. After I was finished with the rhetoric classes, I figured I had it made. I remember

[1] Last two sentences added later and do not appear on the tape.

arguing with my physics teacher when he gave me a 97 on an exam. I thought, “wait a second, where did I lose the three points?’ I went to see the teacher about it, and he said, “Well as a matter of principle, I don’t give hundreds on exams.” In retrospect, thinking about that, I can’t imagine why I complained to that teacher.

In any event, I do remember my college physics courses. I really liked them. The instructor encouraged me to go into physics, so that was the change. At that particular time I went into engineering physics, probably in my sophomore year.

HAIGH: So was “engineering physics” a well-established major at that point?

SWARZTRAUBER: It was established at the University of Illinois, but my sense was that it was not widely established in the overall university system. It was just applied physics which was perfect for me because that was my interest.

HAIGH: What kind of applied mathematics would have been featured in that curriculum?

SWARZTRAUBER: Up through advanced calculus, ordinary differential equations, and complex variables. I mainly focused on physics, mechanics, optics and electronics. I also enjoyed chemistry. I also enjoyed the labs, measuring the charge on an electron and so forth. I began to think of myself as reasonably bright. It was a new and very nice feeling.

HAIGH: Did you have any exposure to automatic calculation or numerical analysis?

SWARZTRAUBER: No hand calculators, we used slide rules. Do you know what a slide rule is?

HAIGH: Analog, logarithmic.

SWARZTRAUBER: Yes, that’s right. We would whip them out in exams. They gave about three digits of accuracy, perhaps not that much. But there was no classroom computing, as it exists today.

HAIGH: Would you have used any desk calculating machines?

SWARZTRAUBER: Well, that’s an interesting story. Not at that particular time, because a slide rule was certainly adequate for the way the courses were taught. My first recollection of desk calculators was at my father’s office. I was very young, maybe just tall enough to look across the desk. He showed me his new Freiden calculator, a big light brown calculator that he was very proud of. I would watch it do a division with its carriage bouncing around. It would do a ten-digit division in about 15 seconds or something like that; I just thought that was marvelous. So that was my introduction to machine calculation. Computation is now at about ten orders of magnitude faster

HAIGH: You graduated in 1959, and I know that Illinois was an early user of computer technology with its ILLIAC I and II machines. Did you have any exposure to the campus computer?

SWARZTRAUBER: That’s where the whole thing started. This was in the fall of 1958. I had heard about the computer that was installed, the ILLIAC, at the University of Illinois. I was finishing up my degree at that time, and I had an opportunity to take electives. I thought "this sounds interesting," so I took a course in computation. At that particular time, it included everything: it included programming; it included some very straightforward, simple algorithms for determining the trajectory of a satellite around the moon. But it was all machine code language, hexadecimal. L4 was an addition or something like that. I took that course, and that was it as far as I was concerned. I knew what I wanted to do. The course was taught by Lloyd Fosdick as a young man, who later, to my delight, became head of Computer Science at the University of Colorado [1970].

HAIGH: So was he on the staff of the computer center at that point?

SWARZTRAUBER: I think he was with the newly established Computer Science Department. I still have the notes from that course. Like I said, it embodied all the various aspects of computing, which are now divided into the fields of math and computer science.

HAIGH: Before you took that course, did you have a clear sense of what a computer was?

SWARZTRAUBER: Well of course in the sense of my father's mechanical computers. I'd heard about electronic computers maybe one or two years before that. And I suppose that sparked my interest when I heard about the ILLIAC being installed.

HAIGH: Do you have a sense of whether taking this course in computing and using the machine was something that many of your classmates in engineering physics would have done, or was it something that only a small portion of the students would have been exposed to in those days?

SWARZTRAUBER: Oh, very small. The University of Illinois was huge and this was the only class, perhaps even the first class. There were probably about fifteen of us, something like that. We went over to look at the computer. Everything in those days was in terms of a thousand or a thousandth. The machine had 32 thousand words. That was huge! And a millisecond was blindingly fast compared to my father's mechanical calculator. It was very exciting for everybody in the class.

HAIGH: Can you describe just the physical process of using the computer? For example, would you have physically interacted with it, or did you just drop the cards off somewhere and receive some output back later?

SWARZTRAUBER: No cards, it was even before cards. We used ticker tape and teletype for both input and output.

HAIGH: Five-channel paper tape?

SWARZTRAUBER: I'm not sure how many channels it had. It would just chug away. It would input the instructions and the computer would execute them. The results would be printed out on a teletype machine. And also, they had a small CRT that displayed the computed orbit for the satellite problem. It was a little humiliating if you computed the orbit incorrectly for all to see. It had lots of lights. It was a huge computer. Just the face of it would have been half the size of the wall with the accumulators and various registers. The display could be slowed so that the contents of the registers could actually be followed by the students as the program was executed.

It had a Williams tube memory that consisted of a row of CRT tubes along the top and a grid of spots on each tube corresponded to memory bits.

HAIGH: Would you have signed up for a time slot on the machine to go and use it?

SWARZTRAUBER: That’s right. Well, the class would sign up, and you would kind of line up. You would type the tape first, and then you would hand it to an operator. She was a pretty crusty lady and perhaps she had to be. She was dealing with many students, and she would supervise the actual operation of the computer. I suppose it’s just reading the tape, but if the computer would hang, she would be responsible for arranging service. They would diagnose the problem by tapping with a small rubber hammer until they located the faulty connection or tube.

HAIGH: Did you use an assembler, or did you have to write directly into the machine code?

SWARZTRAUBER: This was even before machine code. It was binary, actually hexadecimal. I can remember a few things, like L5 was basically an instruction for loading the accumulator. There was a whole series, probably twenty instructions including transfers. Later, in the Air Force, I was initially puzzled by the machine code on the IBM 704 such as a transfer or TRA, which did not have a binary equivalent. This was a very brief period because machine language was quickly replaced by FORTRAN with GO TO statements.

HAIGH: Jumps.

SWARZTRAUBER: Jump instructions, exactly.

HAIGH: How did you react personally to this first experience in programming?

SWARZTRAUBER: Oh, I loved it. In fact, it was rather like the first physics course I took when I was in high school. I would just do all the reading and homework before it was due. Lloyd had put together this marvelous set of notes and I read them like a novel or newspaper. There was just simply no question as to the direction in which my career was going to go. I was just very eager. I was fortunate because some of my fellow students were undecided at that particular time as to what they wanted to do, and it was really difficult for them. Some of them would graduate and they wouldn’t have jobs. I had a job automatically because I went into the Air Force. In those days, ROTC was mandatory, at least two years of it. I took the full four years and graduated as a second lieutenant. At one time, I wanted to fly, but they changed the requirement from three to five years in the Air Force. I was really getting into physics by that time, and beginning to think a little bit clearer. I thought, “Well, that’s probably not a very good idea because I could forget a lot of physics in five years.” So I took the three-year option and luckily, worked on state-of-the-art computers the whole time.

HAIGH: I’ll ask you more about that in a second. I just have a lot of questions about ILLIAC. So, I’ve actually been able to look at some of the papers concerning it in the archive at the Charles Babbage Institute. One of the things I found was that from the early 1950s the computer center had been receiving some grant money directly to produce a library of useful routines and programs for numerical analysis. So I was wondering if you had any memory of there having been a library of routines available, and if you might have used any of them for the course?

SWARZTRAUBER: Well, maybe. For example, the orbit problem that I mentioned earlier, called a subprogram for integrating the ODE. That was not something I wrote. But that is a vague recollection. Even the concept of a subroutine came a little later. But you may know more about this if you've been to the archives. Are you sure that was the first ILLIAC around that particular time frame?

HAIGH: Yes. In fact, somewhat earlier, I think starting in 1950, very early, there was small grant basically to try out the machine. I don't even think it was operational then. Maybe they were just writing the programs on paper for numerical analysis and then looking at the annual progress reports that were being filed. It's clear that by maybe 1952 or 1953 that the project had turned into developing a library of routines. I imagine that those might have been more useful to people who were coming in with problems that they needed to solve rather than just introductory courses.

SWARZTRAUBER: Right, researchers.

HAIGH: Rather than just for an introductory course.

SWARZTRAUBER: Exactly.

HAIGH: But I thought it would be an interesting topic to bring up, as you were there and had used the machine. Now, when I brought you back to ILLIAC, you said you automatically had a job in the Air Force. So was it mandatory for you to go into the Air Force because you'd already enrolled in the ROTC program?

SWARZTRAUBER: Two years was mandatory, but it seemed reasonable to take the additional two years and complete my military obligation as an officer. As a result, I received three years of very important experience and education. I could only specify a region in the country and I chose the southwest because I thought I would end up in sunny California, maybe L.A. But actually, when I got my assignment, it was in Albuquerque, New Mexico at Kirtland Air Force Base. I went down there a day or so early, and went to headquarters and said, "Look, I want to work on computers," and they said, "We've got just the place for you. You might want to go over and talk to the people at Crumley Lab." So I went over and I talked to the computing group. When I actually reported for duty the next day I said, "Well, I've already talked to the people in computing," so they sent me directly over to that group. Kirtland had the very latest computing equipment, an IBM 704 and FORTRAN was the programming language. It was an open shop operation. I would put my own cards in the card reader and load tapes on those big drives. It gave me a real jump-start in terms of my computing experience, and also numerical analysis. However, I began to feel deficient in mathematics. We were basically doing bomb calculations that would run all night using the pseudo-viscosity method given in the book by Bob Richtmyer [Difference Methods for Initial-Value Problems, First edition]. Calculations were all one-dimensional at that time with spherical symmetry. We computed the evolution of materials that were subject to a horrendous amount of radiation. In retrospect, I think we were actually doing independent checks, on computations that were being done at Los Alamos and Livermore. Kirtland was an Air Force Special Weapons Center [AFSWC]. Of course, we had a very high security clearance.

HAIGH: Was getting the security a drawn-out process?

SWARZTRAUBER: Not for me, I didn't have much history. I was twenty-three years old. It didn't take anywhere near as long as it does now. I was never outside the fence. But then I was already an officer so perhaps the paperwork preceded my assignment to Kirtland. In fact, I don't recall any facilities outside the fence. The programmers were all in the same room. It was a good- sized room with about six desks; two rows of three, facing each other. The chief had a master's degree from Stanford, and his name was Dwayne Jensen. He was very sharp. Again, it was a time for learning for me. I also enrolled at the University of New Mexico. I realized that I was going to have to go into math in order to be more effective in scientific computing. I took a number of courses in mathematics; it was analysis and higher math at that point. The University of New Mexico didn't have courses in computational math at that time. However that didn't really matter too much because I read all the books available at that time, and continued to learn computational math on my own.

HAIGH: You mentioned that this was an open shop system and that there were about six programmers?

SWARZTRAUBER: That's right, and that's really how they worked. We wrote the codes that solved the physicist's equations. We then drove a couple of miles to the computer that was located on Sandia base. We spent most of the day at either one or the other site.

HAIGH: So it was basically the six of you and the machine, and you had it to yourselves?

SWARZTRAUBER: Pretty much but there were others running on the machine. It was located on Sandia base and I believe they had access to the machine as well. However I do not remember any significant competition for machine time. It provided an excellent learning experience, with state-of-the-art machines and implementing methods in Richtmyer's book. Richtmyer was at NYU and into computing very early. His book was basically the bible at Kirtland. Everything, all the computational methods came right out of his work. Later, as a student at CU, to my absolute delight, Bob came to CU, and I became his student. Can you imagine my delight after spending all this time with his work, his writings, and implementing his methods! He was basically a god. To have him as major professor was overwhelming. He was also a terrific human being.

HAIGH: So a lot of the methods you were using would come out of this book.

SWARZTRAUBER: Yes.

HAIGH: Now, was there also a library of reusable routines associated with that computer installation?

SWARZTRAUBER: Not really. My recollection is, in fact, of starting a "library". I kept the special purpose programs that I wrote and found that I could sometimes scavenge parts for new special purpose programs. So I began to convert these to subroutines for later use by myself or others. The same sort of thing carried over into NCAR. Even to this day I have a couple of boxes buried downstairs. I've got at least one or two boxes full of cards for these old routines that did various things: ordinary differential equations, partial differential equations, permutations. All of the work that I published was tested on the computer and much was cleaned up and made suitable for distribution but still at an informal level.

HAIGH: So when you were doing the programming in the Air Force, each programmer would develop their own personal collection, but there wouldn't have been any institutional mechanism to share them?

SWARZTRAUBER: It was very informal. In fact, like I said, this mode carried over into NCAR. The first serious packages I remember are IMSL and NAG. They had individuals that knew what they were doing and did the community a valuable service.

HAIGH: Yes, but those would have appeared about ten years later in the beginning of the 1970s.

SWARZTRAUBER: Yes. My first recollection of a "package" was at NCAR, something else, a statistical package of some sort. It's still around.

HAIGH: SPSS?

SWARZTRAUBER: I think that's the name of it, but back then it was something else. SPSS, isn't that the same IBM's scientific software?

HAIGH: You may be thinking of SSP, IBM scientific software.

SWARZTRAUBER: Okay, then it must have been something else. [A biomedical statistics package that evolved into SAS software]

HAIGH: That came out about 1965. What would have been available at that point was the SHARE user group software library. The 709 family installations were pretty much all members of the SHARE group, and they contributed routines for system software, mathematical things, and specific applications into what was basically the first public domain software library. One of the issues with that was the quality of documentation, the quality of the routines, but I know at least some sites would retrieve routines from the library. There was essentially a catalog that went to all the installations, and the machinery to supply copies of the cards if requested.

SWARZTRAUBER: Well, in terms of special functions and things like that, those were available very early on, so I'm not exactly sure if we're talking about the same thing. Those libraries would grow as well. There was a very useful book by Hastings [Approximations for Digital Computers, 1955] that had approximations to special functions. I remember coding those and adding them to my "private stock". Certainly I was doing that at Kirtland, and then also at NCAR. I recall coding a fast exponential function specifically for the 709.

HAIGH: So your recollection is that if you needed a specialized routine that you would be more likely to go to a book, to find the algorithm and code it yourself, than to try and find a version that had already been coded.

SWARZTRAUBER: At that time there was not that much "black box" software out there. And I was not comfortable until I understood every line of code. Confidence with other software grew but even then it would usually be dissected or modified in some way to suit the specific application. I would do this even with my own "private stock". Without detailed knowledge of a library subroutine, one cannot be altogether certain of the limitations that are imposed on your calculations. And there are always limitations.

HAIGH: How did you experience this transition from programming and machine code to programming in FORTRAN?

SWARZTRAUBER: Well, at first that was a little puzzling. When I first got down to Kirtland, I picked some manuals right off the bat, but could not find the hexadecimal instructions. But it didn't take long to understand the concept of an assembler and a compiler. The computer itself converted instructions like TRA or transfer to the actual binary equivalent, how clever! So at that particular time, there were two steps. In other words, there was the assembler that converted machine language instructions into their binary equivalent, and the FORTRAN compiler that converted mathematical expressions into machine language. It was no big deal. It took about an hour or two of reading to understand what was going on. In fact I thought of FORTRAN as something you just did rather than being taught. But of course today there is much more to FORTRAN90 than the early compilers.

HAIGH: Did you find that FORTRAN made you much more productive?

SWARZTRAUBER: Yes. There's no question about that. It was a giant step. It was an obvious step when you think about it in retrospect, but it was an important step in terms of putting together models and computations. It's phenomenal how long it has lasted. It's still very much alive. So that's over 40 years!

[Tape 1, Side B]

HAIGH: So, you've been talking about your experiences programming during your Air Force service. Now, were you doing basically the same job during the whole three years?

SWARZTRAUBER: Yes. It was an exciting time because the 704 was a state-of-art machine. It was almost the last of the tube machines, which were not all that stable. They would have to change out a few tubes every night. Later Kirtland got a CDC 1604, which was the first all-transistor machine. It was much faster and very stable. It had certain new functions on it, like a max function that could pick out the maximum of an array. It was a very exciting time. I learned a lot.

HAIGH: So, this machine was installed in the same computer center during your time at the Air Force?

SWARZTRAUBER: That's right. My office was in Crumley Laboratory on Kirtland AFB, but the computer itself was actually located on Sandia Base, where I spent a lot of time.

HAIGH: As your three years were drawing to a close, what kind of options did you consider for what to do next?

SWARZTRAUBER: The Bay of Pigs invasion and the confrontation with the Soviets kept me in the service three additional months. So I got out in June of '62 and went to work for a very small service bureau in Albuquerque by the name of McAllister and Associates. I just did whatever work that came along. I don't think it was a terribly good career move, but it ultimately led to Denver and NCAR. They opened a Denver office at 655 South Broadway. We did the 1962 Colorado elections for UPI, which was quite exciting. A large room full of people were answering phones, recording votes, and key punching them onto cards that were then read into a CDC 150A. We also had a second CDC 150A for a backup. The 150A was a desk computer, not a desk top, it was the whole desk!

HAIGH: Did the service bureau have its own full sized computer?

SWARZTRAUBER: Only the 150A. We had a very close connection to the local CDC office at that time, which was in the same building. We did various projects like the election.

HAIGH: At that point you definitely saw your career as being a programmer?

SWARZTRAUBER: Oh, no question. I loved computers. There was just absolutely no question in my mind what I wanted to do. However, I would say in light the fact that we did those 1962 elections, my direction within computing was not terribly well defined, but that was to come later.

HAIGH: But you didn't stay very long with the service bureau?

SWARZTRAUBER: No, they in fact folded. They didn't have any money, and basically it just collapsed. We got work, we were busy, but nevertheless, they collapsed. My supervisor at the time was Robert Strieby, an exceptional man. He worked to provide us all with new jobs. In fact, one day he came in, and said, "There's an ad in the *Denver Post* for a programmer at the National Center for Atmospheric Research in Boulder." I told him I wasn't interested because I already had a good job offer with the RAND Corporation through contacts I'd made in the Air Force.

HAIGH: Would that have been out in Santa Monica?

SWARZTRAUBER: Right, in sunny California, where I originally thought I would go. Bob gave me some really good advice. He said, "Paul, always give yourself the opportunity to say no," and that made an impact on me. So I made an appointment, and I drove up to Boulder. However I couldn't find the National Center for Atmospheric Research. I could not find it! I was right there with the address, and I'm looking at this small building but expecting a national center to be a really good-sized building. Actually NCAR was in rented quarters at that time. Well, I thought I've got a good job offer; I'm not going to worry about this. I started to leave, but then thought well, I'll take one more drive around the block. And when I did, I saw this little sign, like a real estate sign, pushed into the lawn that read, "National Center for Atmospheric Research." I interviewed most of the afternoon, and I was very impressed. The head of computing was Glenn Lewis. He got his degree from NYU under Bob Richtmyer although I did not know this at the time. He was an extremely bright man and a little eccentric. I knew I would learn a lot working for him. He actually gave me an interview exam! It was an oral exam, and he was asking very fundamental questions about mathematics and programming. His interview "exams" became legendary. I decided to take the job.

HAIGH: How did the salary compare with the RAND Corporation?

SWARZTRAUBER: You know, I don't remember what RAND was offering. Well, let me tell you a story; this'll be interesting to a young man like yourself. My starting salary at NCAR was

$10,000. Now, my first house, the first house that I bought in Boulder, was $17,650. That's 1.7 times my annual salary. Same job, same house, is now a factor of five! You know, it's no wonder that even with working wives young people are struggling to make ends meet..

HAIGH: Now, given the experience that you had had in the Air Force with this kind of simulation on atomic problems, did you ever consider going to work for Sandia or Los Alamos or Livermore?

SWARZTRAUBER: Well, the RAND Corporation was, of course, a defense contractor; I'd be doing the same sorts of things there. So, yes I did consider that. But I'd been to Boulder years earlier. As you drive into Boulder you get to the top of a ridge and view this beautiful valley, rather like travel pictures of Switzerland. So much of life is random, like my finally spotting that sign in the front lawn, which led to Boulder and NCAR. A few synapses closed or something or other, and I decided to take one more drive around the block and that made a huge difference in my life. I never regretted it. It's not that it saved my life from ruin, if I had gone to RAND then I probably would have finished school at UCLA and life would also have been good. Santa Monica is beautiful.

HAIGH: You've suggested that NCAR was quite small at that point.

SWARZTRAUBER: Yes, in computing, initially there were probably about six of us compared to about a hundred now.

HAIGH: How big would the organization have been as a whole?

SWARZTRAUBER: Oh, fifty; no more. That's a real rough guess. These figures are available. You interviewed Buzbee. He was in management.

HAIGH: Well, Buzbee was at Los Alamos until about the 1980s.

SWARZTRAUBER: That's true. Every time there was an opening for the manager of computing, I would call Bill, and he finally decided to apply for the job. He was a good director. All of the directors were pretty good, but some better than others.

HAIGH: When you joined NCAR, what kind of problems were people working on? Were there a whole bunch of topics, or were there a few big projects?

SWARZTRAUBER: Well, the big project was weather prediction, but there was a whole bunch of small projects. Virtually everybody was using the computer: aerosols, chemistry, balloonists, experimentalists, and solar physicists. The first president of UCAR and director of NCAR was Walt Roberts who was a solar physicist. I remember a lot of problems. In fact, that's one thing about atmospheric science. You name it, and there's a computational math application, everything including linear algebra, ordinary differential equations, partial differential equations, and statistics. You encounter just about everything. I remember computing the shape and loading of a high altitude balloon.

HAIGH: What kind of computer was installed there?

SWARZTRAUBER: We used the University of Colorado computer for about a year or so. It was an IBM 709, which was replaced with a 7090. It was a transistor machine, I'm pretty sure of that.

HAIGH: So that would be the 7090. They added the extra digit to celebrate transistorization.

SWARZTRAUBER: Well, the transition to transistors was very exciting because tube machines were so unreliable. It's irrelevant to people who grew up with transistors.

HAIGH: Were there any big differences between the way that the University computer center was set up and what you'd been used to in the Air Force?

SWARZTRAUBER: Yes, it was a closed shop operation. They would log your deck in, run it, and you would drop by later for the printout.

HAIGH: Did that make a big difference to how you had to work as a programmer?

SWARZTRAUBER: No, because we still punched our own cards. Well no, that's not exactly true; there were keypunchers available, so if you wanted to go to the trouble of writing it down, on the big FORTRAN forms, they would punch it for you. But this was just an added nuisance so I'd key punch it in myself.

HAIGH: About how long would you have to wait after you dropped the cards off before you got your results back?

SWARZTRAUBER: I would say a relatively short time, within the hour. The jobs that ran longer would be run overnight. I don't think delay was much of a problem. Of course there was the usual compliment of problems that would run over night. My observation over the years is that jobs always run the same amount of time no matter how fast the machine. The length of time is determined by the time constant of human patience.

HAIGH: So the response you were getting was quick enough that it didn't make any real difference the way that you had to approach things.

SWARZTRAUBER: Well, I think that you probably would try to do a little multi-tasking. Of course, you didn't wait around during that period. You'd go back and work on somebody else's problem, or developing an algorithm or other programming or reading. Although I took courses at the University of New Mexico, I didn't really focus on an advanced degree until I got to NCAR. I am so grateful to NCAR; they mentored me in many ways including time off to take the courses. I would study in the evenings, so I basically got my degree on the job. I remember a professor telling me, "Well, we can't consider you a serious student if you don't come down here and take an assistantship." But that would have meant a sizeable reduction in salary.

HAIGH: When did you enroll in the university?

SWARZTRAUBER: I think it was almost immediate. I'm pretty sure that was fall 1963, because I went to work at NCAR on January 9th. So I would have been thinking about that immediately, and my guess is, the first opportunity I would have had would've been the fall of '63. I would take a couple of courses each semester and grind away, year after year. I got my Ph.D. in 1970.

HAIGH: Clearly your salary and career as a programmer were satisfactory, so what was it that you were hoping would happen as a result of getting some advanced degrees?

SWARZTRAUBER: Well, I can tell you exactly. I worked for a number of scientists and understood their projects very well and was able to make substantive contributions. Starting with their equations I did the numerical analysis and got the numbers for them. Usually I would also write a section of their paper on the computational methods. I had been doing that sort of thing very early on and the scientists were good to me. They put my name on the papers, which jump-started my publications when I went into research. But to answer your question, I liked their life with the freedom to choose the direction of their work and the associated respect. You are right about salaries, programmers, at least at the highest levels of expertise, were in demand and salaries were comparable to the scientists.

HAIGH: So you liked their life more than yours. Was it because they had more freedom to decide what kind of things they did?

SWARZTRAUBER: Yes. I don't know if it's quite the same today because scientists are encouraged to chase the money. But for most of the time at NCAR, scientists chose the direction of their research. In fact, the senior scientists were responsible for defining the research directions and goals of the institution. What could be better than waking up in the morning, going to work, and solving the problem that interests you the most? One is "driven" to solve ones own problem whereas one simply "works" on somebody else's problem. The real draw was to have that kind of freedom, the responsibility to define the computational directions that would most benefit NCAR.

HAIGH: So the culture there would be that to rise up in the organization, you would be required to have some advanced degrees?

SWARZTRAUBER: That's a strong statement. I think that a softer version of your statement would be appropriate. But yes, in general, I think what you're saying is correct. It would be easier to "rise up". Like any research organization, there are individuals with very high salaries and in management positions that do not have advanced degrees. But to pursue my interests, the degree made it easier. NCAR nurtured me in many ways, including support for an advanced degree and then support for my research, which took many directions, not only computational mathematics but also computer science and design.

HAIGH: So looking at your resume, you made these first few publications in 1966, 1967, 1968, where you're one author out of two, one a couple, one author out of four on the first one. [P. C. Kendall, S. Chapman, S. I. Akasofu, and P. N. Swarztrauber, Computation of the magnetic field of any axisymmetric current distribution with magnetospheric applications, *Geophys. J. R. Astr. Soc.,* **11**(1966), pp. 349-364]. So those are cases where people would come along to you with a problem that needed solving?

SWARZTRAUBER: The physicists, right.

HAIGH: You would help them with the coding and the mathematical methods.

SWARZTRAUBER: Exactly.

HAIGH: And then you would finish up as author.

SWARZTRAUBER: Yes, They were very appreciative and expressed it in a very concrete way. I felt appreciated and respected and began to think of a research career. My first programming job at NCAR was for George Platzman. He knew exactly what he wanted and had a very good sense of what could be expected of a computer. He was a gentleman and set the bar very high.

HAIGH: Now, would the users at this point mostly write their own code with some assistance from you, or would most of the general scientific application programming be done by people in the computer center?

SWARZTRAUBER: Very good question, because at that particular time, there was something called the "programming pool." So the physicists were separate from the computers. They would rely on and be assigned somebody out of the programming pool to help with the computer part of their problem. I still think that that's a pretty good way to do it. But when money got tight, the various groups would solidify their position with money and staff by bringing individuals, and their salaries, from the pool, into their group. So the pool with its budget was ultimately distributed among the divisions.

As time went on many of the scientists were coding their individual research on vector computers. But in the early era of multiprocessors in the early 90s, the pool reemerged in the sense that, again, most of the coding was done by individuals in the computing division.

HAIGH: Do you have a rough sense of what point chronologically the programmer pool model would have withered away?

SWARZTRAUBER: No, but let me make a note, because I can find that. My feeling is that it was in the '80s sometime. I don't really remember. But I understand the significance of that, so I'll try to find that out. [Dick Valent tells me that the pool was abandoned in 1976].

HAIGH: So then, through the '60s and the '70s, most of the actual programming would have been done by specialists in the computer center rather than by the scientists themselves?

SWARZTRAUBER: Right.

HAIGH: That's interesting.

SWARZTRAUBER: There were a lot of machinations associated with the advent of vector computers, a lot of grousing about the difficulty of vector coding. But there was no question that the effort would be made because the rewards were substantial in terms of speed up. I'm pretty sure the programmer pool was still intact.

HAIGH: All right. When you first arrived there in 1963, was there any kind of software library?

SWARZTRAUBER: No. I'm very clear on that, because at that time I continued to gather "private stock." I would have a tray or two of those, and people would ask, "Do you have a program for doing this or that?" or, "Do you have a program for an FFT?" or, "Do you have a program for ODEs?" I had coded up all of them, Runge-Kutta and stuff like that, so I could just hand it to them.

HAIGH: So it would take place informally between individuals.

SWARZTRAUBER: Yes, absolutely, graphics the same way.

HAIGH: We'll return to that question later on and obviously talk about how that changed over time. In these very early days, you'd said that for the first few years you were using a computer at the university.

SWARZTRAUBER: Yes.

HAIGH: When did NCAR get its own computer?

SWARZTRAUBER: Before the mesa laboratory was built, because NCAR did own a computer in the building they were leasing from the university. So that would have been some time between 1963 and '65, because the mesa building was finished in '65. So we're talking 1964, and I think that was a CDC 3600, if I remember.[2]

HAIGH: Do you know if the installation of the computer had any particular impact on your work or the functioning of the computer center?

SWARZTRAUBER: Well, it was the same format. It was a closed shop like the University. The first director of computing was Glenn Lewis. At first the operating system was in terrible shape. So he personally took on the task of debugging it! One evening I saw him at the consol of the 3600 with a foot thick listing. He was debugging the system. The talent of the man was mind-boggling. He was a real Renaissance fellow. He taught me how to play Go. To some extent, that has to answer your question. Yes, CDC delivered that machine. It was an excellent machine, but the software was pretty much up to us at that time.

HAIGH: Did the programming group expand to include any system software people?

SWARZTRAUBER: Yes. Glenn hired a couple systems software people shortly after I came on board and that was the first clear division of responsibilities in the group, although it was not formalized until much later [1973]. An interesting competition developed between the systems and applications people who saw more of the computer memory being used by the system. Paul Rotar came in at that time and I was relegated to "second place" because his job brought him close to Glenn and management. Eventually I developed a profound respect for Paul who chose good computers and made them work, all the way through the Cray XMP. He had a lot of responsibility over the years and was always equal to it.

HAIGH: So you've said that it stayed a closed shop in terms of operation.

---

[2] According to the NCAR website, the CDC 3600 arrived in November 1963, but was not usable until early 1964 because it did not come with an operating system of any kind.
http://www.cisl.ucar.edu/computers/gallery/cdc/3600.jsp

SWARZTRAUBER: Yes. I wasn't putting cards in a card reader at that particular time. They were being handed over the counter and to somebody who was running the programs and scheduling the jobs.

HAIGH: So there had also been a team of operators?

SWARZTRAUBER: Yes. Well enough to keep the machine running around the clock.

HAIGH: So did you have anything to do with those people, or did you see them as a kind of separate group?

SWARZTRAUBER: Yes, separate, as a working group, but NCAR was small then and everybody knew everybody.

HAIGH: So in those early days, the operating systems would have scheduling for batches of jobs, handling input and output, would they?

SWARZTRAUBER: In the early days of the 3600, scheduling was done by the operator. Short debug runs would have been given priority. Long runs or "production" runs would be put on at night and weekends or if the machine became idle.

HAIGH: But as a programmer, can you recall back in these days in the mid 1960s, as you wrote your program, would you have to worry about how the operating system handled things like input and output? Or was that stuff still pretty much handled by the application programmer directly?

SWARZTRAUBER: Input was a FORTRAN read statement, and output was a write statement. Your results came in the form of large printed pages called listings from very large printers, about the size of a large desk.

HAIGH: So FORTRAN was taking care of all that?

SWARZTRAUBER: Yes, FORTRAN was handling that as well as disk access. We also had microfilm processing rather early, a microfilm camera mounted on a CRT. This permitted the animation of computer generated weather patterns.

HAIGH: So the CDC provided one, but it wasn't very good?

SWARZTRAUBER: I don't think CDC provided that. It came from a separate company. [It was called a Dicomed microfilm plotter]. CDC gave us help with the operating system but we had the responsibility for developing the FORTRAN compiler. Dave Kitts wrote the compiler for several machines. But later, very good FORTRAN compilers became generally available.

[Lunch break]

Session Two begins, July 16th, 2005

HAIGH: I believe that when we concluded the first session, you'd just been talking about the early days of the NCAR computing facilities and through the arrival of the first computer on site.

Now, I think it might be appropriate at this point to return to a topic you had mentioned earlier, which is your graduate education. You've talked already about motivations for this, particularly the idea that within the culture of NCAR that this would bring you more freedom to choose your own interests and the kinds of problems you worked on, and in general bring you more respect and appreciation around the place. But perhaps you could talk more about your experiences in graduate school, who you worked with, how your interests developed, the kinds of courses that you took, those kinds of issues.

SWARZTRAUBER: I think the main thing to mention at the start is that NCAR was extremely supportive of my attending classes and continuing my education. I had to maintain an indication that I was a serious student by keeping a good grade average and so forth. But I was permitted on work time to take two classes at the university. Now there was a problem because although I was interested in computational mathematics, the curriculum was not yet established at CU. But that didn't trouble me too much because I could pick that up on my own. There were some very good books at the time like the twin books by Ralston and Wilf. By this time it was pretty clear that I needed to augment my physics background with mathematics. So I took all of the applied mathematics I could get my hands on. But by that time I was in the degree program and required to take a number of pure math courses, which to my astonishment, I enjoyed, thanks to the remarkable teaching skills and dedication of teachers like Jack Hodges. Actually, I was able to use some abstract algebra later on in the design of multiprocessor communication algorithms. [P. N. Swarztrauber, Multiprocessor FFTs, *Parallel Computing*, **5**(1987), pp. 197-210]. But a lot of the math at the University of Colorado was just your basic solid courses in classical mathematics. I think I also mentioned that I was very, very fortunate to get Bob Richtmyer as major professor in the last two or three years of my work and of course he was able to guide me in computational mathematics. His first love was actually physics, so I had several courses in physics from him as well.

HAIGH: What was the reputation of the mathematics department at UC Boulder at this point?

SWARZTRAUBER: I think it's about the same as it is today. Probably not on par with the schools on the coast, but I think the solid math education that I received was perfectly adequate. I never had the feeling that I was somehow behind in my education. I recall the feeling that I needed more classical math to deal effectively with computational math. But once having that background from CU, I always felt I had the necessary tools. Computational math was not widely available at that time but books could be found. I recall reading some very nice notes on the subject from NYU.

HAIGH: Were their any subjects that the department was particularly well known in?

SWARZTRAUBER: Bob Richtmyer was internationally famous as a physicist and for his work on computational fluids. I think the pure math department was also pretty well recognized by the international community.

HAIGH: You've mentioned Richtmyer. Beyond that, what kinds of topics would have been covered in the applied mathematics curriculum at this point, and how well integrated would the practical use of computers have been into the ordinary graduate education?

SWARZTRAUBER: At that particular time, there wasn't a computer science department so I had to teach myself. I was insatiable, so I would get my hands on every conceivable book possible at the time. One of my favorites was called simply "Applied Analysis" by Cornelius Lanczos. The books were coming out of the schools that had early instructions in computational mathematics: NYU, Stanford, Illinois, and Harvard. So I was basically self-taught in that area. This was ok because at the same time I was receiving a strong applied mathematical background: ODEs, PDEs, linear analysis, functional analysis complex analysis. So much of numerical analysis is based on these subjects. It was okay to get just a very good background in classical mathematics. I recommend that to any student who is interested in computational mathematics.

HAIGH: Numerical analysis and electronic computing were still topics that you more had to seek out for yourself than something that would be a standard part of the curriculum?

SWARZTRAUBER: Yes, that's absolutely right.

[Tape 2, Side A]

HAIGH: Did you form any other relationships in grad school with fellow students or with faculty that proved in any way significant later in your career?

SWARZTRAUBER: Well Bob Richtmyer immediately comes to mind; however, because I was working at NCAR, I didn't have a lot of interaction with the students. I was told at one time that I really couldn't be considered a serious student if I were not at the university in the position of a research assistant. After graduation there was good interaction with the Computer Science Department. Lloyd Fosdick was head of computer science for quite some time, and he had initiated my interest in computers at the University of Illinois. I also had an appointment as an adjunct professor and joined the department meetings for some time. I also served on a couple doctorate committees. We started a Rocky Mountain chapter of the ACM, with monthly meetings. The founding institutions included: NCAR, CU, NBS, and CSU. The Colorado School of Mines was also represented and I vaguely recall meeting at Denver University.

HAIGH: Now, I think I saw that group referred to in your resume as a chapter of the ACM SIGNUM special interest group for numerical analysis. Was that a new chapter of ACM itself within which was a new chapter of SIGNUM? Had there been an ACM chapter in general there before?

SWARZTRAUBER: Now that you mention it, it is quite possible that only the special interest group SIGNUM was new. We had a good turn out. The lectures were well attended.

HAIGH: Would that have been during the late 1960s

SWARZTRAUBER: I'd have to research that, but it sounds about right.

HAIGH: It sounds as though you did this in conjunction with graduate school, so I assume it would be about the same time period.

SWARZTRAUBER: It was after graduation so that would have put it in the early 70s.

HAIGH: While we're on the subject of SIGNUM, do you remember any kind of interaction with the group on a national level or involvement with its newsletter, or maybe anything you might have learned from it that influenced your own career?

SWARZTRAUBER: SIGNUM helped develop local interaction and a local collegial base and SIAM helped develop the national and international base. I was oblivious to its importance then because I was focused on the events of the time. It was only later that I realized the importance of this collegial base. This became important for all the usual reasons but also a very important reason for me personally. I was required to submit ten letters of support from internationally respected scientists with my materials when being considered for promotion to Senior Scientist. So SIAM was extremely important to me, not only in terms of the meetings. Particularly at the beginning of one's career, there's a learning curve that develops very rapidly just in terms of discussing problems and seeing the various approaches that people take to solving the problems.

HAIGH: So, I'll return to those questions then later on as we discuss your involvement with SIAM and other organizations in the '70s and '80s. I suggest finishing up then on the graduate school topic. Could you talk a bit about your dissertation project and your experiences with it?

SWARZTRAUBER: Yes, I can. Bob Richtmyer gave me a very interesting problem. The title of my dissertation was "A study of the time dependent free boundary of an ideal fluid." I struggled with that problem, and of course I was treating it numerically because that was my interest and Bob's as well. But it was a difficult problem, and although several results were obtained, the key result was a nonexistence proof, which was not really very satisfying to me. But the interactions that we had at that particular time were important. Bob had purchased a place in the foothills just west of Boulder, I would drive up every Saturday morning, spend about fifteen minutes talking about this week's results, and then we'd spend most of the time talking about just stuff. So I learned a lot about free boundaries and published a couple papers, but also learned some other things as well. Bob was a gentleman's gentleman. He was the best part of my graduate studies.

HAIGH: Which papers then would have come directly from the dissertation?

SWARZTRAUBER: Here's one [P. N. Swarztrauber, On the numerical solution of the Dirichlet problem for a region of general shape *SIAM J. Numer. Anal.,* **9**(1971), pp. 300-306]. Here's the second one [P. N. Swarztrauber, A numerical model of the unsteady free-boundary of an ideal fluid, *Quart. Appl. Math.,* **31**(1973), pp. 245-251].] Those two.

After I got the degree, my supervisor [Ted Hildebrandt] asked, "Do you want to be classified as a research scientist now?' I told him I didn't know, because at that particular time I was a little bit insecure about my dissertation, less than completely satisfied and a little bit insecure about my future in research. So I said, "Maybe give me 25 percent of my time. Let me just see what comes out of it in terms of research." That's why I was categorized at that particular time as a Ph.D. scientist, rather than saying something about research in the title or putting me on the research ladder. It was shortly after that, that a very interesting problem came through the door having to do with computing atmospheric pressure in certain atmospheric circulation models. Any solution method would have broad application say, to computing electrostatic or magnetic fields. The underlying mathematical problem was the solution of what are called elliptic equations, which are common to all of these areas. At that time "iterative methods" were widely available in the literature and considered the method of choice for solving elliptic equations. So I was using these methods to develop software for the scientists.

About that time [early 70s] Roland Sweet came to NCAR. Actually he came to the University of Colorado with a joint appointment at NCAR and we worked together on solving elliptic equations. I believe it was Jack Miller at NCAR that brought the paper by Golub, Buzbee and Nielson on cyclic-reduction to our attention, and we began looking at these new “fast” methods including the Fourier method by Hockney. It’s also important to mention Oscar Buneman who stabilized cyclic-reduction and made it useful. While looking into these methods it became evident that cyclic-reduction could be generalized to solve a larger class of problems, which resulted in the paper that marked the beginning of my research career. [A direct method for the discrete solution of separable elliptic equations, *SIAM J. Numer. Anal.*, **11**(1974), pp.1136-1150].

HAIGH: So then prior to that, you said that you were normally spending 25 percent of your time on research?

SWARZTRAUBER: That’s right.

HAIGH: And the other 75 percent, you were helping people by writing programs for them?

SWARZTRAUBER: Writing programs and software including the embryonic beginnings of FISHPACK would have been included in the 75 percent at the time.

HAIGH: Would your background to that point have been fairly typical compared with the other programmers who were working at NCAR at that point?

SWARZTRAUBER: Well at that time, of course, I had a degree. So I would say I was given more time to pursue my research interest, but other than that, the 75 percent, yes, it was pretty much like the other programmers.

HAIGH: So then, your interest in establishing your own research agenda was not shared by the other programmers? They were more content to stay where they were?

SWARZTRAUBER: Yes, I think that’s right.

HAIGH: Well, maybe at this point then, it would be appropriate to talk in more detail about the FISHPACK project. So you’ve said that that paper, and then the subsequent research that you conducted inspired by Gene Golub, led into the creation of FISHPACK?

SWARZTRAUBER: Gene Golub became aware of my work and invited me to present it at Stanford. I was very excited about this and consequently forgot my presentation materials at the rental car agency in San Francisco. I realized this after arriving at Stanford but only about an hour before the seminar. I wrote a five line outline of my talk on a piece of paper and stuck it in my shirt pocket. I was scared out of my mind but the talk went well even without the outline. Gene Golub found out that it was my birthday and so we had birthday cake before the talk. The icing had a grid placed on it, which was related to my talk. I recall meeting Oscar Buneman at that time whose work was fundamental to a large body of work that followed, including my own at that time.

That was really the start of my research career. After that, I moved into an entirely different relationship with the research establishment and computational mathematics. And, simultaneously, that began FISHPACK. What we [Roland Sweet and myself] did, was to implement that method plus a number of others as the embryonic start of FISHPACK.

HAIGH: Maybe it might be appropriate, then, to talk about prior to the creation of what became FISHPACK, were there any standard routines in this area available at NCAR?

SWARZTRAUBER: Not really – application specific codes that implemented iterative methods were developed at NCAR as well as elsewhere. However it was easier to write your own application specific code than to figure out someone else's code well enough to be able to extract anything that might be of use. A very good book on iterative methods at that time was by Richard Varga. However the fast direct methods were vastly superior and consequently the iterative codes were not included in FISHPACK. There was a problem with our early work that included only "utilities" because it required the user to write a driver program or a "wrapper" that was a nontrivial exercise. We got very little interest. We had implemented these brand new, very efficient methods in software. We were going to offer these to the users, but we just didn't have many takers, it was puzzlement to us, but we finally found out the reason. What they needed was something that starts with their partial differential equations and actually sets up the finite difference equations for the utilities that do the hard work. The utility routines were a product of the research but the drivers were needed to make them useful. Usage increased dramatically with the drivers. At Netlib alone, FISHPACK has been hit a half a million times or so, which does not include downloads from NCAR. We got a very good response to the package.

But there's one other thing associated with FISHPACK that made it possible. We found out how to "solve the unsolvable". Now, what I mean by that is that there are many reasonable cases, which could be posed by a scientist, for which there is no mathematical solution. Now, in the past, we would have to tell the user, "I'm sorry, we can't solve your equations because they don't have a solution," and of course, this was unsatisfactory to everyone, particularly the scientist who could care less about mathematical existence proofs and simply wanted solutions that were known to exist in nature on a daily basis. We got around this problem by computing the closest solution. Although we couldn't solve the equations exactly, we could get a solution that was in most cases close enough. This made the package robust. It wasn't going to fail. Well, any package can be made to fail, but they could put their problem parameters into the package, and if it couldn't solve it exactly, it would give them the closest solution and tell them how far it was from an exact solution. That was absolutely key to the development of a robust FISHPACK.

HAIGH: I have some follow up questions on those things. A while ago, you made this distinction between just a collection of utilities, which didn't get very much use, and then the addition of these drivers so that the scientists could basically start with their equation and get it solved. Now, to your mind at the time, was it the addition of those drivers that made it a "PACK" rather than just a collection of utility codes?

SWARZTRAUBER: Well, at first, our embryonic efforts resulted in three or four programs. It would have been a bit of a stretch to call it a "PACK" but I suppose one could. They were big codes that did the heart of the computation, but they required a certain level of numerical expertise that clearly limited the class of users. So the drivers greatly expanded the class of users. They were not that complex mathematically, but nevertheless time consuming to write. The lessons learned in the development of FISHPACK were incorporated into later packages. For example, in FFTPACK, the utility at the heart of the thing is the complex FFT, and the drivers translate the input and output of the complex FFT into something that most individuals are familiar with, like the trigonometric representation of a function. The drivers were essential to these packages, something that the scientists could look at and say, “This is meaningful to me. I can provide the required input and I know what to do with the result.”

HAIGH: Now, when you were working on FISHPACK, were there existing packages that you had in mind as the model for what a high quality mathematical package would look like?

SWARZTRAUBER: I don’t think so. In fact, I remember that Roland and I would have numerous coffee breaks where we would argue about many aspects of the package. So I don’t think we really had any kind of a model to work with. I thought we were basically establishing a model.

HAIGH: Well, my impression has been that this “somethingPACK” naming convention was influenced by the success of EISPACK from Argonne. Do you think that the name would have been influenced?

SWARZTRAUBER: You could probably find somebody who talked about subroutine packages previous to that, but EISPACK and LINPACK set the standard very high for anything called “something PACK”. I’m hopeful that we were able to maintain that standard. FISHPACK was directed to our particular user base, the atmospheric science community (well, it’s broader than that, but we were focused on that particular community at the time) and added drivers, which would make it easier for the scientific community to use. EISPACK consists of utilities, the core computations, and is the most useful to one with some background in computational mathematics.

HAIGH: I had previously come across this concept of driver in conjunction with EISPACK, and I think there was a version that was specific to one of Argonne’s own platforms, that was EISPAC, without the ‘k’ at the end, that was more of a driver or a bundle, if I remember it correctly.

SWARZTRAUBER: It likely depends on whom you talk to but I think of a driver as providing a match between a common mathematical computation and an application. But “driver” has been used in a more general setting. For example, there’s a version of SPHEREPACK with “wrappers”, which is called NCAR Computer Language (NCL). Instead of converting the whole thing to FORTRAN 90, it was embedded in a “wrapper”. It looks like FORTRAN 90 to the person calling it. It’s a very nice contribution because the user does not have to specify work arrays that create about 90 percent of the problems associated with its use when the user’s arrays are not long enough. Dennis Shea and Mary Haley produced NCL, which further simplifies the use of SPHEREPACK. They did a very nice service to the community. It was a sizeable effort and got a few bugs out of SHEREPACK in the process.

HAIGH: Returning to the origins of FISHPACK, you had mentioned that these particular kinds of mathematical problem were found in a variety of different areas.

SWARZTRAUBER: Right.

HAIGH: Was there anything about the kinds of work undertaken at NCAR that made problems of this nature show up more frequently than they might do at other institutions?

SWARZTRAUBER: Oh yes. Like I said, every problem started with somebody coming through the door. And there is a very large class of atmospheric models which require the solution of that kind of equation. Any model with a scalar function like pressure or vorticity or divergence, all of these quantities are found in virtually every atmospheric model and often computed by FISHPACK. They're expensive to compute so the package was of great interest to modelers. In fact, some researchers reformulated their models just so they could use FISHPACK.

HAIGH: So you mentioned that the motivation was the development of these new and much faster mathematical methods. Were you just taking the methods from the published literature?

SWARZTRAUBER: Well, of course, it starts that way. The starting point was the Buzbee, Golub, and Nielson paper. But then we added things, like the separable elliptic equation. So we're generalizing techniques; we added things like finding the closest possible solution to the problem. I think that between Roland and I there were 12 refereed papers that were in some way or another, implemented in FISHPACK.

HAIGH: I know that during this era, new methods were often presented in the form of ALGOL algorithms. When you looked at papers that presented new methods for solving equations, did they include any kind of algorithmic implementation, or were they just more on the level of mathematics?

SWARZTRAUBER: Usually it began with the mathematics. I recall translating a few ALGOL algorithms to FORTRAN but usually one translated computational mathematics into FORTRAN because most papers did not include an ALGOL algorithm. We did not write in ALGOL because FORTRAN was so established at NCAR. This was also true of other languages that would surface from time to time.

HAIGH: I don't think, except for a brief period in Europe, that anyone much wrote in ALGOL as a practical matter, but I do know that in the Communications of the ACM, there was a long running algorithms department.

SWARZTRAUBER: Over the years, any number of languages have been developed; however, the huge body of legacy codes at NCAR, and for that matter the other labs, prevented their adoption. A new language would have to demonstrate a huge advantage to overcome this inertia.

HAIGH: And in the 1960s algorithms for CACM were required to be written in ALGOL.

SWARZTRAUBER: I remember that, and it didn't make any difference to me because it was easy to convert to FORTRAN. My guess is that most ALGOL was converted to FORTRAN for use. They were attempting to establish a standard and ALGOL had a nice mathematical "feel". I was an Associate editor for TOMS [Transactions on Mathematical Software] for a while.

HAIGH: Yes, when TOMS came along in the 1970s, I think that broadened things out and allowed for FORTRAN as their kind of nod to the real world. But why I'm asking this is: I know in the case of EISPACK, it was implementing work associated with Wilkinson and Reinsch.

SWARZTRAUBER: Right.

HAIGH: And I know that Wilkinson had some initial publications that didn't include any kind of example implementation of the methods. Then there was a later book, I think, the handbook. [J. H. Wilkinson and C. Reinsch, editors. *Handbook for Automatic Computation*, vol 2.: Linear Algebra. Springer-Verlag, Heidelberg, 1971]

SWARZTRAUBER: I know what you're talking about, the yellow books.

HAIGH: Yes, which included example implementations, but only in ALGOL, and then work on EISPACK started with the translation of the ALGOL into FORTRAN, originally in a quick and dirty manner, but then they wanted to convert it into the best FORTRAN that they could. So I was wondering, it's clear that something analogous is going on with FISHPACK, in the sense that there are these new, radically better mathematical methods, and that your interest (at least in part) is in packaging them so that they can be widely used. I was wondering, did you have a comparable kind of source of mathematical methods to start off with, where the mathematics had been laid out, and there was maybe an example implementation in ALGOL, and your job was take that, turn it into FORTRAN and package it well? Or did you have to kind of dig deeper and pull together more things in order to get the methods together and implemented in the first place?

SWARZTRAUBER: Well, of course, we built on the research that came before us, including the excellent paper by Golub, Buzbee and Nielson on cyclic reduction, which really started our interest in fast direct methods. None of the early work was published in either ALGOL or FORTRAN for that matter, so it was up to us to produce the FORTRAN code. But this was not a problem; we had been doing it for a long time.

We always had the user in mind because we worked for the National Center of Atmospheric Research, so our papers were directed toward that community and to making it easier for the user and at the same time, extending the class of problems that FISHPACK could solve.

HAIGH: So then in this case, your work implementing and packaging these methods fed back directly into improvements the methods, which were then published in their own right. Were there any of those papers that you would identify as being particularly important?

SWARZTRAUBER: Well, yes, of course, the "separable" paper. Not only did it help FISHPACK, but it certainly established some confidence in my mind that I would be able to survive in a research environment.

The "surface of the sphere" paper [P. N. Swarztrauber, The direct solution of the discrete Poisson equation on the surface of a sphere. *J. Comput. Phys.,* **15**(1974), pp. 46-54] was important because it introduced the concept of "solving the unsolvable". The documentation [P. N. Swarztrauber and R. A. Sweet, Efficient FORTRAN subprograms for the solution of elliptic partial differential equations, Technical Note TN/IA-109, National Center for Atmospheric Research, 1975] had very wide distribution. Interestingly the FACR algorithm [P. N. Swarztrauber, The methods of cyclic reduction, Fourier analysis and the FACR algorithm for the discrete solution of Poisson's equation on a rectangle, *SIAM Rev.*, **19**(1977), pp. 490-501] is the fastest known direct method for solving Poisson's equation, but it has *not* been implemented in FISHPACK.

A users guide to the Fourier method [P. N. Swarztrauber, Fast Poisson Solvers, In: Studies in Numerical Analysis, MAA Studies in Mathematics, G. H. Golub, ed., Vol. 24, Mathematical Association of America, 1984] was written when visiting Stanford. Both cyclic reduction and Fourier methods are described in an applied setting with boundary conditions other than the usual periodic. Yes, I think that's a few of them.

HAIGH: All right. Now, I know that the names of Roland Sweet and John Adams are also associated with FISHPACK.

SWARZTRAUBER: Yes, absolutely.

HAIGH: Can you talk about how the collaboration developed, who those other two individuals were, and what part you all played in the project?

SWARZTRAUBER: Sure. Roland came to the University of Colorado in about 1970, and also had an associate position at NCAR. We very quickly became collaborators because we got along and we were interested in the same topics. We wrote a number of papers and developed the embryonic software or the utilities that I talked about earlier. Pretty soon, it was becoming apparent to us that we could solve a large class of problems that are frequently encountered in the atmospheric sciences. So we began. With each paper we would add a feature or perhaps a program to FISHPACK. We were interested in spherical geometry, so that was a whole new application for these methods. If a person is interested in weather, chances are the work will be posed on a sphere. We also included software for other standard geometries. It was a terrific collaboration, and as I said, we had a lot of interaction in developing that package. We argued about everything: names, parameter lists, work arrays, temporary arrays, initialization, and so forth. I think those arguments were very effective in producing a nice package. John Adams came along a little bit later, and implemented a driver called SEPLI for separable elliptic equations. Later he developed his own package called MUDPACK for solving the general linear elliptic equations using the multigrid method. This is another important package that is well known and supported by the multigrid community. Later still, John developed the drivers for SPHEREPACK. John had his Ph.D. and used considerable creativity and initiative in the development of these drivers.

HAIGH: Now, you mentioned earlier that, at least through the 1970s at NCAR, programs were written by people in the programming pool rather than, in general, by the researchers themselves. So when you talk about "users" here, and packaging the utilities into a form that would be more accessible to them, are you really talking about the programmers in the programmer pool?

SWARZTRAUBER: Both, the drivers were written to facilitate the use of the software by scientists within the atmospheric science community, which was an international community. The drivers also facilitated use by a member of the pool who programmed for a scientist. As time went by, NCAR scientists seemed to spend more time with the development of their codes. This was quite reasonable and clearly my approach to research. It is the only way to make a good assessment of ones results.

[Tape 2, Side B]

HAIGH: Now, let's talk in a little bit more detail then about the actual process by which FISHPACK was created. You've talked about the sources of the mathematical methods and about the importance of developing these drivers. Did you define any coding practices for the project in terms of things like naming conventions, calling sequences, indentation of source code. Did you find it necessary to set standards for things like that?

SWARZTRAUBER: Certainly not to the extent that current practices are standardized. Our focus was on the user interface and not so much on the coding itself. This was pre FORTRAN 90 so we still had statement numbers. Long before this time I had written a code that renumbered FORTRAN statements [called RENO] that also did some cleanup. FORTRAN 90 made a nice contribution by eliminating most statement numbers.

We developed a certain pattern in the names that we would assign to the various programs, in some way related to the problems that they would solve. So we had a series of names, and those were things we used to argue about. And we defined a sort of standard way of listing the various arguments in the calling sequence. We defined a way, which at the time was very useful, of dealing with just one work array and then in the wrapper, or the first subroutine, busting that up into all of these work arrays that would ultimately be used in the other subprograms. By the way, that was an important step, because it would have driven the user nuts to specify all the work arrays and their lengths as well. But then, FORTRAN 90 eliminates the need to specify some work arrays, which is important because about 90 percent of the errors were made in specifying the length of those work arrays.

HAIGH: So that would have been the main area that you were concerned with in terms of coding practices? Do you think that approach you took in terms of the work arrays was novel, or do you know if it had been used by people working on other packages?

SWARZTRAUBER: Well, it certainly was novel to us. Actually, I think this method is now called "establishing pointers." I was at a meeting once, where Gene Golub mentioned to me that a user of the package was impressed with the way that multiple work arrays were handled.

HAIGH: How about testing of these routines?

SWARZTRAUBER: We were very lucky. Buzbee was at Los Alamos at the time, and said, "We would like to certify the programs for you." He took the whole package and Mike Steuerwalt ran test after test of the package, and of course we interacted with them. Actually, the certification was published by Mike in TOMS [Sept 1979]. That was a very important contribution to the package.

HAIGH: And that was prior to the first general release, was it?

SWARZTRAUBER: Probably not! [Chuckles] The only thing that might have come along after that would have been John's driver SEPLI for separable equations, but we pounded on that quite a bit ourselves. And of course, the entire package has had years of pounding. It is very old but is still distributed a few times every month.

HAIGH: Yes. So when would the first version of the pack have entered general use within NCAR?

SWARZTRAUBER: Roland and I were asked to contribute to a new Divisional publication called Atmospheric Technology, in which we described the programs that we had at the time [September 1973 issue]. The article was called "Efficient Subroutines for the Solution of General Elliptic and Parabolic Partial Differential Equations." However the programs would have already been in use before that time, particularly within NCAR. It was published as a package in NCAR Technical note TN/IA-109 in 1975, "Efficient FORTRAN subprograms for the solution of elliptic partial differential equations." It became the most widely distributed NCAR technical note with several printings of over 1300 copies.

HAIGH: Was there a period where the software was only used inside the lab, and then it would have been published and distributed more widely after that?

SWARZTRAUBER: Yes, and then the word gets out that NCAR has a package for solving this or that, and it spreads within the atmospheric science community first, even before the package is completed.

HAIGH: So some version of the package would have been in use within the lab, and to some extent outside it, even before the first official release?

SWARZTRAUBER: Yes. No question about that. Then, of course, we would get calls and users would help us out by saying, "we've got a problem here," which would help to identify bugs.

HAIGH: So, in terms of testing, you've mentioned Bill Buzbee was responsible for this effort at Los Alamos to rigorously and exhaustively test and certify. Did you also have the experience that sometimes just ordinary people would try to use the package would report some kind of bug?

SWARZTRAUBER: Oh, you bet. And that was both a favor and a pain. It was that feedback that ultimately saved a lot of time for other users. We would make those changes, but it was rather informal. We'd get a phone call [email didn't exist!] and add it to the list. Then at some point we'd go in and correct them all. It was pretty straightforward.

HAIGH: Was there a pattern to the kinds of bugs that might be discovered by users, rather than being spotted by the members of the development team?

SWARZTRAUBER: Well the development "team" consisted of Roland and me, and we knew how to use the code and consequently we could not even conceive of some of the uses and abuses that the code would be subjected too. So we would uncover bugs in the mathematical implementation. Users needed help with the basics like the specification of input parameters or the initialization routines. So we added a series of tests on the input parameters that would alert the user to a problem with their specification of such. Of course, we would run our tests; we had an extensive set of test programs. But it was home grown and it was really Bill's and Mike's rigorous effort that provided the testing effort with credibility.

HAIGH: Right, and that's kind of the point behind my question. Obviously before you put anything in the hands of the users, you're going to do your best job of testing it, and you'd probably have some kind of standard problems that you try each new version against. And you've also said that clearly that wasn't successful in catching all the kinds of problems. I was wondering, was there any kind of pattern you can generalize about the kinds of bugs or errors that were harder for you to catch and that the users themselves might eventually notice? Would they be issues with interaction between the code and the compiler, for example?

SWARZTRAUBER: Yes, but these came later. We thought we were very clever by asking the user to provide a single work array which would then be broken into multiple work arrays that were necessary for the subprograms. However, later this led to multiple compiler errors of the type “mismatched argument types.” We had to break up the work arrays into real, integer, and complex work arrays. This was not good because it was already a nuisance for the user to compute the length of even a single work array. Indeed sometimes the user would simply guess a large number that wasn’t large enough. So we asked the user to input the length of the work arrays, which would generate an error if they were not long enough. However the user had to output the error parameter to detect the error, in other words the new argument also generated a new error source.

HAIGH: Would they be subtle areas where the mathematical method turns out to be not quite suitable for all the cases you thought it was suitable for?

SWARZTRAUBER: Absolutely, all of the above. Every piece of software has a legitimate parameter space. If you try to shove it outside that parameter space, there’s going to be trouble. You can make the resolution too large, and so ultimately the errors pollute the solution to the point that it’s not good. You can specify a certain set of parameters that will lie outside, will break the software. This is true of any software package. We test for obvious errors in the input parameters, on average about ten such tests. Nevertheless sometimes we would get phone calls from very irate graduate students who were using the software incorrectly but had managed to pass these tests but were still unhappy with the results. Their research hadn’t panned out and it was our fault! Another typical call would be: “This is my problem. Will the software work?’ or “This is my problem, this is what I did, and I’m getting incorrect results.” That led to consulting, which was just a part of our job.

HAIGH: Were there any cases where feedback you got from users actually made you think in a different way about any of the mathematical methods, or exposed weakness is their applicability that you wouldn’t have previously been aware of?

SWARZTRAUBER: Those happened very early on. For example, we had these really fast methods, which we and others before us had published. So we implemented those. I’m talking about just the core, the utility at this point. As mentioned earlier, we knew at the time, that they could be made to fail because the equations did not have a solution for a particular set of parameters (boundary conditions). Interestingly this was true of almost all the programs in FISHPACK. This led to the work on finding “nearby solutions” and solving the unsolvable, which increased robustness considerably. [P. N. Swarztrauber, The direct solution of the discrete Poisson equation on the surface of a sphere. *J. Comput. Phys.,* **15**(1974), pp. 46-54]. This was a direct consequence of user input.

But in terms of programming errors, sure, early on we had those kinds of bugs as well, and we would track them down and fix them as we could. It was difficult to notify users of these bugs. Today, users could be notified simultaneously that an update is available via the web. In fact, I don’t think some of the current huge software packages would be practical without the web.

HAIGH: I'll ask you more about these issues of distribution and bug fixes in a minute. So another part of the development and packaging progress would presumably be documentation.

SWARZTRAUBER: Right. We were determined to provide adequate documentation because it was essentially nonexistent for so many programs. The FFT is a classic example. Many programs were available for the complex FFT but once computed it was not at all clear what to do with it. So we were very intent on providing some decent documentation for these programs. Originally this was available on FORTRAN comment cards that provided the relationship between the FFT and the coefficients in the trigonometric representation of the input data. This type of information was also contained in the technical note for SPHEREPACK.

HAIGH: So, imagine that you are an internal user within NCAR who's interested in using this. Presumably there's some repository you can go to and pick up the decks with the source code? Is that how it would work, that you would physically pick up cards?

SWARZTRAUBER: Yes, good question. I'm trying to think back what did we do? Cards were distributed but this would have been very early in the days that the library consisted of a few trays in my personal collection. However, very soon thereafter, these programs were a part of the NCAR library and could be loaded from the main-frame disks by inserting a compiler directive in the users program (cards).

HAIGH: All right, so you think by this point circa 1973, '74 that the library source code would be stored on disk and accessible to users?

SWARZTRAUBER: Yes. I'm pretty sure that they didn't have to come to us. They would just use the calling sequence in their code, and it would be available to them together with a collection that was greatly expanded beyond my collection. Libraries such as IMSL and NAG were available quite early.

HAIGH: Would there be somewhere they could go and read printed documentation to explain to them how they should use these routines?

SWARZTRAUBER: Yes, there would be two places. They could view the comment cards and the technical note [P. N. Swarztrauber and R. A. Sweet, Efficient FORTRAN Subprograms for the Solution of Elliptic Partial Differential Equations, Technical Note TN/IA-109, National Center for Atmospheric Research, 1975]. Later, John Adams provided documentation for his driver SEPLI for the separable elliptic equations.

HAIGH: As far as that printed documentation, did you do anything like organizing seminars or putting up flyers to let the internal NCAR users know that these new tools were available?

SWARZTRAUBER: It was announced in the newsletters and presented at many meetings both national and international. We appreciated the feedback that we got from the users and felt that the software was appreciated. This package as well as others helped to elevate software to a position of science and respect. Nevertheless it will probably never get the respect that research does even though it can require the same or more creativity. I don't think I used my brain any less when writing software compared to writing a paper. The problem is that software creativity is difficult to measure. The community was very supportive of software but it was the papers that resulted in promotions. I knew this very early on and it was likely one of the reasons I decided to pursue research. I always had two "jobs"; namely, to be relevant to the applied side of the atmospheric research community and second to my colleagues in the computational math research community. The software was for the first and the papers were for the second.

HAIGH: That's something I wanted to ask about. So this work on FISHPACK you'd said was the beginning of your research career and your shift into more of a scientific position rather than a program within the lab. But I'm also thinking you had mentioned that most programmers didn't want to be researchers, and also obviously, NCAR is a center that's about doing atmospheric research, not applied mathematics research per se. So was it initially difficult for you to persuade people that research in applied mathematics in general, and the creation of high quality software packages in particular, was something that NCAR should be taking seriously as a research area?

SWARZTRAUBER: Well, I don't think NCAR ever thought of software, even now, as a research topic. I think that they certainly value it, but in support of research. And that's perfectly okay because I don't feel working at the National Center for Atmospheric Research that I can strictly do algorithms that have no relevance whatsoever to that community. That's why I say, and I'm very adamant about it, that with respect to these packages, the first thing that happened was somebody came through the door, and they had a question. And that's absolutely true of FISHPACK, FFTPACK, and certainly SPHEREPACK. I think if you're in the university it's absolutely okay to pursue a more varied agenda. But I've always felt I had to be relevant, and I felt very relevant when a nice software package would come out of my research.

HAIGH: Let's talk about distribution of FISHPACK outside the lab. So you've already said that word would get around initially within the atmospheric science community.

SWARZTRAUBER: And the labs, too. I talked about Los Alamos and Bill Buzbee, and Stanford and Gene Golub. I was invited and spent some time out there, and interfaced with folks from engineering who wanted to solve their partial differential equations using the methods and the package.

HAIGH: How would that work then? Somebody would hear about the package and ask you about it, and then would you just run off some cards and mail them to them?

SWARZTRAUBER: Well, it would have been by tape by that time. Somebody had the responsibility for software distribution and it was relatively simple to request that a tape be mailed. Then, like NCAR, the receiving institution would place it on their library so that future requests could be directed to existing in-house libraries. But the question is a good one, because I do remember times when we would get a request for a particular routine. Well, that's okay, except that the routine may call other routines, and so usually we would just send the whole package. This is now greatly simplified by using the "make" file utility and the web.

HAIGH: Were there any kinds of licensing or copyright concerns?

SWARZTRAUBER: That came along later because NCAR, like so many others, developed an intellectual properties group. So they really couldn't distribute it for money because it had been in the public domain, but they did ask us to put a copyright notice in front of every routine.

HAIGH: Do you know what time period that would have occurred in?

SWARZTRAUBER: No, I don’t remember when. At least 15 years ago, but that’s a wild guess.

HAIGH: So you think probably sometime in the 1980s?

SWARZTRAUBER: Maybe.

HAIGH: So in the early days there was no copyright notice contained in the code or the documentation?

SWARZTRAUBER: Correct. And they convinced me it was okay because somebody else could copyright the code and then charge us! And of course I didn’t like that idea at all.

HAIGH: At the very least, say, for the first ten years or so of FISHPACK, there would have been no copyright statement?

SWARZTRAUBER: Yes, I think that’s probably right. There would have been no copyright on the original packages at all, on what would have been the initial package plus drivers.

HAIGH: Did you have a sense that in general scientific software should be freely distributed?

SWARZTRAUBER: If it’s produced with taxpayers’ money, absolutely. NCAR does draw the line between research and profit making companies that are expected to license SPHEREPACK. In fact, a speaker manufacturer licensed SPHEREPACK. They built very high-end systems for concert halls including Carnegie Hall. They actually model the hall and all the reflective surfaces and use SPHEREPACK to analyze reception. They were asked to license the software because they were a profit making company.

HAIGH: Do you know approximately when that would have been?

SWARZTRAUBER: That would have been about 3 years ago.

HAIGH: So that’s more recent. So presumably in the earlier days when there was no copyright statement at all, they would have been able to take that and incorporate it in their commercial product and do that without any worries?

SWARZTRAUBER: Yes, I assume that happened, many times.

HAIGH: Now, do you know if the FISHPACK code was eventually incorporated in any of the commercial software libraries, such as NAG or IMSL?

SWARZTRAUBER: I think it was in all of them. I’m not positive of that, but those are good libraries; I would imagine they would pick it up like Netlib.

HAIGH: So at that point there would have been no objection, even though IMSL was a full profit company?

SWARZTRAUBER: That is correct. It simply was not an issue back then.

HAIGH: I would suspect that you're probably right, but it's just interesting to get this on the historical record because it is quite different from the situation today.

SWARZTRAUBER: Nobody was concerned with those things in those days, but they sure are now. I don't know what to say about it. I suppose it's good to have an understanding amongst all parties, and how unfair it would be if all of a sudden NCAR had to pay for the software that it developed using taxpayers' money.

HAIGH: In this case, clearly, my impression is that you would have no worries that IMSL might be charging for its library if that would be a way of getting FISHPACK into the hands of more users?

SWARZTRAUBER: I wouldn't worry about it, I didn't worry about it. The IP group considers FISHPACK and FFTPACK in public domain. However they contain copyright notices to avoid the situation in which NCAR would have to pay someone else to use this software. On the other hand NCAR requests that SPHEREPACK be licensed by profit making companies.

HAIGH: So your personal position would have been that anything that got the code into the hands of users would be a good thing?

SWARZTRAUBER: This sort of thing didn't even cross my mind at that time. But it's now a different world. If a corporation makes money on software that was funded by taxpayers, it seems reasonable that they pass some to NCAR to support its work. I don't receive any money. I think it's okay if the taxpayers get back a little something. However it's my understanding that, as a corporation, UCAR actually owns the intellectual property and the government retains a nonexclusive license.

HAIGH: How about the process by which the package was supported and improved over the years? We've already talked about the basic debugging process, and you've mentioned this certification effort from Los Alamos. So I was thinking more generally: the FISHPACK numbering has now reached at least 4.0, so there must have been at least four major releases. How did those releases come about? What were the major things added in each version?

SWARZTRAUBER: I don't remember, but the first was probably just the utilities, the fundamental calculations that implemented the new methods. Then maybe drivers were second. Maybe third was, okay, now we can make the package more robust, we can solve the unsolvable. I just don't remember, but it would have been something like that.

HAIGH: So your recollection is that this solving the unsolvable problems and the drivers you've mentioned would not have been in version one?

SWARZTRAUBER: Well, I guess it sounds that way. I just don't remember, and I don't know of anybody who would because "release" would be much too glamorous a word for the software. We would keep track of problems and suggestions and update the package usually timed with a major addition such as "solving the unsolvable". It was probably too much work for a couple of people who were also expected to publish. But we went ahead anyway because we knew that our research could substantially benefit the atmospheric science community, particularly if implemented in software.

HAIGH: So then your impression is that the numbering would have been fairly arbitrary in terms of what's a new release, what's a bug fix, what's intermediate kind of improvement?

SWARZTRAUBER: I think the only thing that would change the number would be a significant addition, something that would probably affect most of the software like extending the package from the Poisson to the Helmholtz equation. We would not have issued a new number for correcting a bug in the program or something like that. It would have had to be a significant change.

HAIGH: Now, one of the things that might've influenced the creation of new versions, I imagine, would have been the arrival of new platforms or new FORTRAN standards.

SWARZTRAUBER: Yes, we tried to keep the architecture in mind but it was difficult, for example to code cyclic-reduction for a vector machine. Although Roland and I wrote papers on this subject, not all that work was implemented in FISHPACK. [P. N. Swarztrauber and R. A. Sweet, Vector and parallel methods for the direct solution of Poisson's equation, *J. Compt. and Appl. Math.,* Special issue on parallel algorithms for numerical linear algebra, **27**(1989), pp. 241-263]. My vague recollection was that we changed the utilities somewhat in a way that would have been have been invisible to the user. However, a little later Roland developed a vectorized version of FISHPACK, which he called CRAYFISHPACK. He sold a few copies and I believe he's still selling them.

HAIGH: All right, so I just pulled up some of the FISHPACK source codes from Netlib, and I'm seeing a little section here of history with some numbers. The numbers and the dates are the same in two different routines, so I suspect that these are for the pack as a whole. So it says version one, September 1973; version two, April 1976; and version three, January 1978.

SWARZTRAUBER: There you go. I'm sorry that I don't have that kind of memory. Does it say what prompted it, which is basically your question?

HAIGH: Not in the two routines I've stumbled on at random. No, I'm seeing lots of nice comments here, and the particular source code I found here says that it's version 3.1, October 1980.

SWARZTRAUBER: I would have no idea just off the top of my head as to what was changed. I could give you examples of things. My impression is that we did not even think of numbering versions until version 2, so version one probably corresponds to the 1975 NCAR tech note, although it would have existed as early as 1973. A new version would correspond to a significant change. The change from the Poisson to the Helmoltz equation was important because it also extended solutions from two to three dimensions. That would have come rather early, perhaps version 2.

HAIGH: Do you think that increased generalization? I would imagine that you might add new routines to handle particular special cases of equations or implement new methods that had come along since the previous version.

SWARZTRAUBER: Well, I think that's certainly correct in terms of the contribution that John Adams made when he implemented the driver for the more general separable equation. Later

John wrote a separate package for solving the general linear elliptic equation. It implemented the multigrid method and was called MUDPACK. By the way, one thing you learn very early in software development is that if you change the user interface [argument lists] you must change the version number. More importantly you must change the name of the individual programs. But I can't help but think of all the help that current technology could provide in terms of software development. A popup could appear in the window of all users saying "There are updates available for FISHPACK. Press the button and download them now." This would be quite remarkable. Actually I don't like popups so email would probably be more appropriate.

HAIGH: Now, you had mentioned that what you had to do, because that wasn't, obviously, available in those days, was to make an effort to contact users and let them know that a new version was available. So how would you go about doing that?

SWARZTRAUBER: I don't think there was a mechanism. I think it was just like the first version; namely by word of mouth. Later it would be announced in the computing newsletter.

HAIGH: So that would let the internal NCAR users know about the new version?

SWARZTRAUBER: The newsletter went to the entire atmospheric science community, which consists of many universities. But that's still a fraction of the users of FISHPACK.

[Tape 3, Side A]

HAIGH: So we don't have in hand a copy of the original 1975 NCAR tech note. However, I do have here, retrieved from ACM Transactions on Mathematical Software, the 1979 paper that you wrote with Roland A. Sweet, entitled "Algorithm 541: Efficient FORTRAN Programs for the Solution of Separable Elliptical Partial Differential Equations." This paper does appear to summarize what was contained in that tech note, so here on page 353 of that article, it says that the NCAR technical note contains seven chapters, each dealing with one of these seven FORTRAN subroutines and an appendix dealing with least square solution of singular linear systems of equations. Then it says, "The first five subroutines solve a Helmholtz equation in Cartesian, polar, cylindrical, interior spherical, and surface spherical coordinates respectively," and then it gives a little bit more detail. Then it says that each of these five subroutines calls one of the last two subroutines, called POIS and BLKTRI, which can be used to solve a more general class of equation. So presumably, that's a description-- Now, let me just check that the tech note that they're citing here really is the original one. Yes, the note it's citing here is given in the references as being 1975 with errata in 1976. It doesn't mention the version number, but it does describe the system as it would have been available in 1975.

SWARZTRAUBER: Okay. That's my recollection as well. The original version, as described in the 1973 Atmospheric Technology article, solved the Poisson rather than the Helmholtz equation.

HAIGH: That's true. Well, let's cross reference that with the dates for the versions that they found in the source code, which says version one 1973, version two 1976. So, I guess if the tech note came out originally in 1975, then it would have described something that was maybe somewhat expanded from the original version one, but hadn't quite reached version two.

SWARZTRAUBER: Could be.

HAIGH: Now, I guess shifting temporarily away from the packages themselves, I think you'd said a little bit earlier that as a result of producing FISHPACK that you and your colleagues were invited to pretty much all the software oriented conferences to describe the work. Can you say something about the reception that the package received within this community and perhaps about your own growing involvement with the community of mathematical software people?

SWARZTRAUBER: The community of mathematical software people was basically a subset of computational mathematicians. My general recollection is that the meetings were very informal and included folks that attended the SIAM meetings and focused on computational mathematics. The driving force in the SIAM community was the computational mathematicians, including those interested in solving elliptic partial differential equations. So at the SIAM meetings we would present the solution. This would be followed by "Oh, by the way, you don't have to implement these methods yourself. You can have the package developed at NCAR and use these methods without delay." You could almost hear a collective sigh of relief.

There is a particular presentation that comes to mind. It was in Austin and presenting the method for solving separable elliptic equations. I got very bogged down in the details of a particular development and had in fact lost my way. I paused for a somewhat lengthy time trying to regroup but realized the hopelessness of my situation. So I said "I'm not sure I believe this result myself." It drew a good amount of laughter from the audience who by that time were well aware of my predicament. I just continued from that point. I have never been that comfortable with public speaking although I have had a couple successes.

HAIGH: So those kinds of presentations would be taking place at conferences in the area that potential users of the packages might be attending?

SWARZTRAUBER: Yes, the applied mathematics aspect of SIAM would draw a good representation of companies as well as laboratories.

HAIGH: Do you remember anything about reactions to the package that you might have received from other people who were interested in mathematical software?

SWARZTRAUBER: It was very positive, but I hasten to add the difference between just the package of utilities and the package of utilities plus drivers. We would get interest in the utilities from some of the people who were proficient in computational mathematics, but the number of interested people grew substantially following the development of the drivers. People didn't have to understand the fundamental algorithms of cyclic reduction or Fourier analysis.

HAIGH: So just to cross reference this with what we found in the 1979 paper, what you're calling "drivers" would be these first five subroutines, which solve a Helmholtz equation, in Cartesian, polar, cylindrical, interior spherical, and surface spherical coordinates.

SWARZTRAUBER: Exactly.

HAIGH: And what you're calling utilities would be the POIS and BLKTRI?

SWARZTRAUBER: Right.

HAIGH: So POIS was for cyclic reduction and BLKTRI was for generalized cyclic reduction?

SWARZTRAUBER: Yes. And those last two utility packages, we got very little interest in the physics community, because they really needed the drivers, the set up routines.

HAIGH: So from your point of view then, your deeper mathematical contributions were being made with these two generalized utility routines, not with the things that actually made them accessible for your audience.

SWARZTRAUBER: Well the “deeper mathematical contributions” were bound up in POIS and BLKTRI, which were made conveniently accessible by the drivers.

HAIGH: In that point of view, the drivers would be relatively trivial to construct, would they?

SWARZTRAUBER: Perhaps compared to the years of research that finally resulted in the last two; however, I don’t like to think that anything I’ve done is trivial (smile). “Solving the unsolvable” is part of the drivers.

Session three begins July 17, 2005 in Westminster, Colorado.

HAIGH: Yesterday we concluded the last session by discussing FISHPACK. I understand that overnight you’ve been able to track down some more information relating to the early history of this package.

SWARZTRAUBER: That’s correct. It was in the last box I looked in. This is a 1973 article. It appeared in a new divisional periodical called “Atmospheric Technology” published by the National Center for Atmospheric Research, number three, September 1973. It gives, basically, a progress report on the development of FISHPACK, although it’s not called FISHPACK at the time. It’s simply called a package of programs for-- Well, here’s the title: “Efficient Subroutines for the Solution of General Elliptic and Parabolic Partial Differential Equations,” and it’s by Paul Swarztrauber and Roland Sweet. It describes, basically, the equations that it solves and also gives a list of the various subroutines that are going to appear in the package. At this point the drivers had been added and we were on a roll. The next significant addition was to generalize Poisson’s equation to the Helmholtz equation.

HAIGH: So that description that we read with the five driver subroutines and the two core ones, would that have applied to FISHPACK version 2, that solves the more general Helmholtz equations?

SWARZTRAUBER: Yes, It seems version 2 included the Helmholtz solvers. Version 1 corresponds to the 1975 tech note.

HAIGH: You also found the original 1975 tech note?

SWARZTRAUBER: That’s right. Yes, indeed.

HAIGH: Did that provide any additional information?

SWARZTRAUBER: A version number is not included so that makes it version 1. Of course, it gives a detailed description of not only the input parameters, but also the equations that are solved, and if a solution doesn't exist, how we manage to get as close as possible to a solution.

HAIGH: By the way, whose idea was the name?

SWARZTRAUBER: That was Roland's idea. We argued about it, and I must say I'm glad I lost. I wasn't particularly fond of it at the beginning, but everybody seemed to resonate to it, so I certainly changed my mind as time went by.

HAIGH: Did you think that the name was too jovial initially?

SWARZTRAUBER: Well, not very descriptive. But it's just fine, I think it's well known at this point and I have no problem with it.

HAIGH: I wonder if you could also compare, as much as you're able to, FISHPACK with packages that appeared in the 1970s or early '80s for partial differential equations, possibly John Rice's ELLPACK system.

SWARZTRAUBER: I think John's package was focused more on users who had an understanding of some of the methods, and certainly on the numerical analysis community in general, whereas FISHPACK and SPHEREPACK and FFTPACK were developed for specific user community. The big difference, I think, is the drivers, focused on the ability for scientists to be able to use it. But I really think you should ask John that because he may have a different understanding of it.

HAIGH: As I recall it, one of the distinctive things about ELLPACK was its modular nature, so that you could experiment by swapping in or out a new discretization module or other method and then experiment with it in that kind of framework, meaning that they would be particularly appealing to numerical analysis researchers.

SWARZTRAUBER: Right, that coincides with my understanding.

HAIGH: Are you aware of any other packages in this are that might have been widely used and that in some sense would have been competition for FISHPACK?

SWARZTRAUBER: An interesting question. I recall several presentations by researchers that compared their latest performance results with FISHPACK. Evidently FISHPACK had become the "fast gun" which inevitably brings out challengers. On a specific problem and architecture it is always possible to code more optimally. I have done that myself for specific applications. But FISHPACK is still being distributed on a pretty regular basis after 30 years. Of course MUDPACK, which was developed by John Adams at NCAR, was applicable to even a wider class of equations, a more general system of equations. However, even though it solved a larger class of equations, FISHPACK would be faster for its subclass. So it really augmented rather than competed with FISHPACK. You've talked to a lot of people. Perhaps you would know of some other packages that compete with FISHPACK?

HAIGH: I know that there are a large number of packages today. I don't have a sense historically of when they might have appeared. It should say for people who are interested in this, it appears that the 1983 IFIP Working Group conference was held in Sweden and devoted to PDE and ODE software, modules, interfaces, and systems, and that the proceedings were published by North Holland in 1984, so I imagine that would give a good overview of the work that was going on at that point. But my understanding is that this is an area where there's such a wide range of different problems that you might encounter that it is not the case that there would be a one-size-fits-all kind of solution that would remove the need for further work in that area. In contrast with the linear algebra and eigenvalue software for example, as I understand it, pretty much solved those problems for the vast majority of users.

SWARZTRAUBER: I think that's right. In fact, in a lot of the calls we would get, the question was "How can I use FISHPACK to solve this particular problem?" There are some tricks that one can use in order to expand the class of problems, but sometimes you would just have to say, "I'm sorry, it won't work." MUDPACK was a really nice contribution because that did treat the general linear elliptic equation, which includes a larger class of problems. I think that John's package is a good package, particularly with the aid of John or somebody who knows about multigrid.

HAIGH: Would you say that FISHPACK tackled the substantial majority of PDEs that people at NCAR would have a need to deal with in their research?

SWARZTRAUBER: The majority of problems? Hmmm, I don't think so. There are a large number of computational problems at NCAR that use a broad spectrum of computational methods. Elliptic solvers are a subset and can make an impact because they use a lot of computer time. For example, FISHPACK can assist in the solution of nonlinear elliptic equations but it's more likely that bits and pieces would be extracted and included in custom software for such problems. I don't think it solves the majority of problems; rather, just a good-sized chunk.

HAIGH: So for the other problems, was it the case that those just weren't tractable, or would there have been other packages in the NCAR library that researchers would have reached for when one of those other problems came up?

SWARZTRAUBER: Well, Yes, There are several computational fluid models that have been developed over the years by NCAR scientists. These models solve time-dependent nonlinear equations and are very complex and can be used to predict weather or climate. They are in wide use by the world community of researchers. For example, they can be used to track pollutants on a global scale. They are a fundamental product of NCAR and have grown over the years to include the oceans, sun, ice and much more. I worked closely with Dave Williamson and Jim Hack over the years and would like to think that bits and pieces of the research that led to FISHPACK and SPHEREPACK have found their way into these models. SPHEREPACK is called a "model development facility," because it can aid graduate students and researchers to develop geodynamic models quickly. However the performance requirements and complexity of the world-class weather and climate models require specific code dedicated to the application and platform.

HAIGH: Well, I think that gives us a natural transition to talk about the other packages then. Can you talk a bit about FFTPACK and how that system came to be?

SWARZTRAUBER: Okay. There were codes for the complex fast Fourier transform very early on. But one of the problems was that if you gave those codes to somebody, they would come back the next day and ask, “How do I make use of the results?” So I would sit down. I would go through the process of telling them what they had. This usually amounted to expressing the complex transform in terms of the real trigonometric representation of their data. I did that a number of times, and finally decided to write both a program and documentation for this process.

It just expanded from that point. We had the complex transform first and then went to the real transform. Probably the most widely used program in FFTPACK is something called EZFFT, which is very widely used because it produces a result that everybody understands. It is the simple trigonometric representation of the data. That is something that the scientists relate to immediately. But there are so many things about the FFT that really needed to be said, which are now included in the tutorials on my web site.

For example, the complex FFT in its simplest form is not suitable for application. It is a lovely and concise form; however, it cannot be differentiated because some of the complex exponentials are aliased or represented in terms of unnecessarily high wave numbers. So things like that needed to be stated and well documented so that a scientist can effectively use FFTPACK in their models or for time series analysis or for something as simple as differentiating the data. And it just grew from there. We included transforms that took advantage of symmetries in the data. For example, the sine and cosine transforms of odd and even data. These could be pulled from the literature; however the fast quarter-wave transforms were developed at NCAR and also added to FFTPACK. [P. N. Swarztrauber, Symmetric FFTs, *Math. Comp.*, **47**(1986), pp. 323-346]. The symmetric transforms in FFTPACK are also widely used for solving partial differential equations subject to the usual boundary conditions.

HAIGH: So there’s a mathematical connection there with the previous work on FISHPACK?

SWARZTRAUBER: Well, there is a connection in the sense that for a certain class of problems you would use the Fourier method rather than FISHPACK.

HAIGH: What’s the time scale on FFTPACK? Would that 1986 publication have been around the time that the package appeared, or was that sometime later?

SWARZTRAUBER: I don’t remember. I began writing FFTPACK very early on because codes for doing Fourier transforms were so frequently requested. I would have to review. Maybe if we were just to look at the web site there would be a date for FFTPACK. But again, there were multiple versions. [Dick Valent tells me that FFTPACK became part of the NCAR libraries in January of 1980]

HAIGH: But your impression is then that it would be fairly arbitrary. There wasn’t a firm distinction chronologically between working on FFTPACK and working on something else. That it was more bundling together and packaging of work that had been underway for some time just as part of your job.

SWARZTRAUBER: Yes, that certainly is an accurate statement. But the key was documentation: clearly defining the transforms so that they can be differentiated, integrated or analyzed in any manner by anybody with a basic knowledge of trigonometric representations.

HAIGH: My rather skimpy research on this topic suggested that the main FFT algorithm was developed by James Cooley and John Tukey around 1965.

SWARZTRAUBER: Yes indeed. In fact, I think I listened to Jim Cooley talk about even earlier developments in that particular algorithm, all the way back to Gauss. But I think in general, they're credited as sort of having put it all together at some point and making it available to contemporary numerical analysts. Lanczos is also given some credit. By the way, I was a green-horn session chairman during one of Jim's talks and in the very awkward position of having to remind him of the time limit (a couple of times). It was particularly painful because his talk was very interesting to everybody. One of life's no win situations.

HAIGH: So that would have been a little bit after you arrived at NCAR. So do you remember when the first implementation of the Cooley-Tukey algorithm would have appeared at NCAR? Did it have a dramatic effect on what people were able to do computationally?

SWARZTRAUBER: Well, it certainly had a dramatic effect on the speed at which the Fourier transform could be computed. Like I said, I think we started working with that pretty early on. I don't remember exactly when I wrote the first FFT or if I was the first to use it. It would have been very early on, but it also would have been just the complex transform. Virtually every numerical analyst has written a complex FFT. It's just a lot of fun, for one thing. The complex FFT can be written in about a couple dozen lines of code. My students at FSU did it for a home work problem. Clearly that is just the beginning of a package with several thousand lines of code.

HAIGH: So homemade implementations of this algorithm would have appeared rapidly in pretty much every scientific research center?

SWARZTRAUBER: Yes, all over the place. Absolutely, it was a national pastime. [Morven] Gentlemen and [Gordon] Sande were very much involved in FFTs. The list would go on, Edson's algorithm for real transforms, for example. FFTPACK was certainly built on the work of many, many people. The quarter-wave transforms were developed at NCAR for solving the kinds of problems that occurred there. It can provide an attractive alternative to FISHPACK for a limited class of elliptic partial differential equations. Roger Hockney is credited for the initial work in this area. But I don't know of any numerical analyst who hasn't written an FFT. But the key was also, again I state, documentation, because, once distributed, it was inevitable that somebody would come back in the office and say, "Okay, I've done the FFT. How do I use the results?'

HAIGH: So these additional methods and algorithms that appeared later, would their advantage primarily be that they could achieve better results with special cases.

SWARZTRAUBER: Yes, that's right. If your data had certain symmetry, you could take advantage of that to speed the transform.

HAIGH: And FFTPACK bundled together a collection of these with this easy to use interface and documentation?

SWARZTRAUBER: That's right.

HAIGH: As you were gathering together these different versions, would you just take a published description and re-implement it, or were you actually incorporating code from different sources?

SWARZTRAUBER: FFTPACK was written at NCAR, but there were other codes in use. The large weather and climate models were using two-dimensional FFTs that were developed at ECMWF [European Centre for Medium-Range Weather Forecasts] by Clive Temperton. They were very good codes that ran fast on the vector platforms that we both had at the time. They were used in spectral models by people who were knowledgeable about their use.

HAIGH: Why was it that you never incorporated any actual code? Was it that the routines were so small that it was practical just to rewrite them and that there would have been no real advantage to start with somebody else's code if you had to completely restructure it anyway? Because obviously today people would be less inclined to start from scratch, I think. Everything's gotten bigger and more complicated.

SWARZTRAUBER: That's a good question. Yes, codes for the complex FFT were available but they were not vectorized or constituted a package. With the advent of vector computers, the codes had to be rewritten, and the vectorization of the FFT became a very popular topic. NCAR contributed to this body of research. [P. N. Swarztrauber, Vectorizing the FFT's, In: *Parallel Computations, G. Rodrigue, ed., Academic Press,* New York, 1982] and then by FFTPACK, which also included the new quarter-wave transforms.

HAIGH: Let's talk about the actual process of coding then. Were there other key collaborators on the project?

SWARZTRAUBER: I wrote every single line of FFTPACK. Now since then, Rodney James converted some of the package to multi-processor systems when he was at NCAR. For several years, he was converting FFTPACK code to parallel machines.

HAIGH: How about testing?

SWARZTRAUBER: Well, it's kind of the same as FISHPACK. We would run basically all the tests that we knew. Basically, the best way to test FFT codes is to just write a slow transform for comparison. It is written in double precision so you can get an accurate estimate of the error in the FFT. The FFT is more accurate than the slow transform so double precision is required to determine the single precision error in the FFT. This also provided a check on the documentation, which included the slow transform.

HAIGH: Would you have developed a collection of test data that you would then feed into both systems?

SWARZTRAUBER: A random number generator is basically what we would use to compute the pre-transform data.

HAIGH: Was that process automated to any extent?

SWARZTRAUBER: No.

HAIGH: And then leading on from that, distribution. So, obviously this was picking up on things that had been developed over a number of years within NCAR. Was there a point where you packaged it altogether and said to the world, “Here is FFTPACK. Why don’t you try it?”

SWARZTRAUBER: Exactly the same answer as to FISHPACK. Word got out, people started and asking for it. Netlib asked for it, and I forget the figures, but I think it has been hit something like a half a million times on Netlib. But that’s only a small part of the usage because there are secondary distributions. It’s distributed frequently from NCAR as well. Actually, it’s held up very well. I think that comparisons have been made. Everybody makes a comparison, because, like I said, any numerical analyst has probably written an FFT code. I’m not aware, at least, of any other package that has overtaken it in terms of distribution.

HAIGH: When the package was distributed outside the lab, did that make a difference to you in any way in terms of the kinds of support issues that you were coming across? Did it make things, for example, much more complicated in terms of having reports of problems with particular compilers or portability to particular platforms?

SWARZTRAUBER: About the same kinds of problems. There were problems, for example, with new compilers. They didn’t like the mixing of modes when calling subroutines within subroutines. The compilers have really improved, and they’re a lot smarter than they used to be, and of course we thought it was very clever to pack everything into a single workspace. But that didn’t pass the standards that came later. But the nice thing about the fast Fourier transform is that it’s so robust. You have a hard time breaking that software. It’s a very stable transform, and the FFT is even more accurate than the slow Fourier transform. So you get fewer questions about FFTPACK than FISHPACK or SPHEREPACK.

[Tape 3, Side B]

HAIGH: Now, I saw a reference online that the FFTPACK project had been sponsored by the National Science Foundation. Is that correct?

SWARZTRAUBER: Well, that is correct because NCAR is sponsored by the National Science Foundation. There were no additional monies provided for its development.

HAIGH: Did you ever consider seeking any kind of grant or special funding to support any of these package developments?

SWARZTRAUBER: No, but today I would, because it’s a totally different environment for the young people today at NCAR. They are encouraged, actually expected, to write grants to supplement the income of the organization. When I started out in science, it was an absolute Garden of Eden. There was a lot of money, and we didn’t have to spend time doing that sort of thing. I don’t know what to say about that because that’s just the way it is now, other than I wish it were different. I wish that the young researchers could spend more time pursuing their interests [while being relevant], because a person who loves their work, is going to do a better job than a person who is simply doing work that he or she has been told to do. They cannot compete with somebody with similar talent who is obsessive about their work.

HAIGH: So you didn't seek funding because basically you didn't need to. Your salary was paid and that was the resource that you needed?

SWARZTRAUBER: Yes, absolutely, it was a terrific job. I couldn't wait to get to work in the morning.

HAIGH: Now, you've mentioned that the FFTPACK code was vectorizable, and in fact, I saw that for at least one of the papers that you published on it that "vectorizable" was in the title of the paper. So was it the case that earlier implementations of the algorithm ran into problems when used on the new generation of vector supercomputers?

SWARZTRAUBER: Yes, no question about it. It is the nature of the FFT that the sequences get smaller as the algorithm proceeds, and so you have to do kind of tricky things to keep those sequences long for vector computers. There were lots of little gimmicks that we used. [P. N. Swarztrauber, FFT algorithms for vector computers, *Parallel Computing,* **1**(1984), pp. 45-63].

It's interesting to me because when vector computers came out, there was the same uneasiness among members of the user community, you know, the complaining, 'They're more difficult to code" and so forth and so on, that you hear now in connection with the use of multiprocessor computers. The difference is that one could obtain a factor of five on the vector computers whereas the early multiprocessors would, at best, just break even. It wasn't worth the effort.

HAIGH: Yes. So was the first vector computer at NCAR a Cray-1?

SWARZTRAUBER: Yes, the Cray-1. Acquisition began during an after hours conversation with Paul Rotar. This would have been during the period that I was acting manager of the computing facility and before it was a division of NCAR. We were in the market for a machine and were considering machines from CDC and Texas Instruments and another company that I do not remember. I mentioned to Paul that we had not yet spoken to the man who had built all of the machines we had purchased to that point. Paul returned to his office and spent about an hour or so on the phone with Seymour [Cray]. He returned, and said "I think we may have found our next computer." We had a long talk about Seymour's new company and machine. It was installed in 1977. I take no credit other than directing attention to Seymour, Paul did all the work.

HAIGH: And do you recall anything in general terms about what it was like after that machine arrived? Was it immediately usable? Was everyone very excited?

SWARZTRAUBER: That was a very exciting time, because if you coded vectors properly, you could see a factor five increase in performance over the computer that just moved out the door. For the whole line of Cray computers, you would see that factor of five. You don't see that now and the reason is just the cost of communication between processors. Vector computers were an initial attempt to deal with the internal communication of data, moving data between memory and registers. So very early on, designers were seeing communication as a fundamental problem, which is even more so today in the context of multiprocessing.

HAIGH: So looking at your resume, I see that from 1979 on, there's a spate of publications on vectorizing things and efficient vector methods. Was your interest in this bringing you in contact with the community of people at other elite Cray sites?

SWARZTRAUBER: Yes. For example, ECMWF, which is not quite the equivalent of NCAR, well not quite equivalent in the sense of their activities but certainly equivalent in terms of talent, definitely an "elite site" to use your words. We talked to those people regularly, in particular with Clive Temperton and Geerd Hoffman. Geerd was the head of computing at ECMWF for a number of years and we exchanged several visits. He held a biannual meeting that was right at the state of the art and several books were published as a result. [For example: P. N. Swarztrauber, MPP: What went wrong? Can it be made right? In: Proceedings of the Seventh Workshop on the Use of Parallel Processors in Meteorology; Making its Mark: The Use of Parallel Processors in Meteorology, G.-R. Hoffmann and Norbert Kreitz, eds., World Scientific Publishing Co. Pte. Ltd., London, 1997]. Both NCAR and ECMWF interfaced with the international atmospheric science community, so it was very natural that we would get together and discuss the latest innovations in the development of machines and algorithms.

HAIGH: How good was Cray Research at supporting people in trying to get good performance out of the machines by reworking their software?

SWARZTRAUBER: I think that they were very good, but they focused more on the general community. Nevertheless the things that they did, certainly in terms of the development of systems were very valuable. And I must say they really had their hands full because that's a difficult problem. The Cray's were clearly superior computers and so successful that they had the resources at that time to do that sort of thing. At the beginning, it was a little slow in terms of development, but they put the effort in. They did, I think, as good a job as anybody could do, faced with that kind of task. Cray user groups were quite active and also contributed substantially to the community.

HAIGH: Do you recall anything about the sequence of releases of FFTPACK? I know that the version four has been widely distributed, and I understand that there's a version five that appeared in the 1990s?

SWARZTRAUBER: Yes. I would love to tell you about version five.

HAIGH: All right, well why don't you tell me about version five then?

SWARZTRAUBER: Okay, because I don't remember the rest of the versions! It was, I would say very much like FISHPACK. If we added major features like vectorization or if we added a new program we would change the version number. But version five presented a special problem. Dick Valent wrote the drivers and I began to write the documentation. Version 5 differs from previous versions by the inclusion of programs for multiple FFTs and multi-dimensional FFTs. This version was motivated in part by the fact that they could be parallelized quite easily Again, I was aware of how important the documentation was, so I began to write the documentation but ran into a very big problem. It is very important to give precise formulas for what is computed so there can be no question in the scientist's mind as to what has been computed.

HAIGH: And that would just be given in the form of an equation, would it?

SWARZTRAUBER: Yes. That would be given, actually, in the form of a slow transform because that's what is needed for differentiation, or integration, or any number of analyses that can be derived from the Fourier series representation. If you are to really understand what has been computed, or for that matter what you will compute based on the results provided by FFTPACK, then the documentation must include the slow transforms. Now, for multidimensional transforms there are a huge number of transforms corresponding to a very large number of cases. The documentation would be huge and unwieldy, perhaps 400 or more pages to include all possible transforms. So it became evident that hypertext documentation would be the way to go. The user could simply specify the type of Fourier transform in each dimension and the hypertext would provide the equations and documentation specific to the user's application. But life took different directions for both Dick and I about that time. Documentation in hypertext would have been a big project and we simply didn't have the time or resources to continue the project.

Now, FFTPACK 5 does exist with partial documentation in the form of something called "Using FFTPACK 5", but the package itself will generate compiler errors because of the reasons that we discussed above. It has been used by a number of savvy people who have experience with that problem. Both FFTPACK and SPHEREPACK could benefit from a FORTRAN 90 conversion like FISHPACK. We ran out of time in terms of updates to the packages, which was probably inevitable considering the fact that the packages seemed to evolve continuously.

HAIGH: So the version that I'm seeing in Netlib at the moment, it says version four, April 1985. So that gives us a date on version four at least.

SWARZTRAUBER: Well, I think that's a good place to refresh my memory.

HAIGH: So I guess then there would've been a flurry of releases up to that point, and then we end up with version four. Really, it would have just settled down until this version five effort was launched.

SWARZTRAUBER: That's right. We had version 5 in the back of our minds for a long time. [Dick Valent tells me that FFTPACK, presumably version 1, was installed on the NCAR libraries in January, 1980]. Of course bits and pieces would have been in use before that date.

HAIGH: Yes, there don't seem to be many dates in the source code on these ones. So that's FFTPACK. Now, you've talked about this kind of gradual process by which it found its way out of the lab, and you've talked about the Netlib distribution. Do you know if this code found its way into widely distributed libraries like NAG and IMSL?

SWARZTRAUBER: I think so. I would be surprised if it's not in there, but can't be positive. I recall directing users to Netlib so there must have been a period where it was easier to get it from Netlib than NCAR. That seems to have happened a number of times. And, of course, NCAR users could just call the subroutines, which would be loaded automatically from the mathematical library. This was also soon true for the UCAR community with the package installed on their computers. I don't keep track of FFTPACK distributions but know that Netlib had distributed a half million copies when I last checked a couple of years ago. I do record SPHEREPACK downloads just because that mechanism was in place at that at the particular time it was made available on the web. It's still distributed a few times a week.

HAIGH: How about if we talk about SPHEREPACK now, then?

SWARZTRAUBER: Well, SPHEREPACK. Again, the development was much the same as the other packages. Atmospheric physicists work in spherical geometry where harmonic transforms are required to provide the same information that Fourier transforms provide in Cartesian geometry. Well, let me just say the following: SPHEREPACK does, in spherical geometry, what FFTPACK does in what's called Cartesian geometry, or just x-y-z. This must be qualified; a fast harmonic transform does not exist in the sense of the "fast" in the FFT. Many researchers including myself have tried in vain to develop a true fast harmonic transform. Nevertheless the harmonic transform can be made "faster". The software developed pretty much like the FFT. So we wrote the harmonic transform, we vectorized the harmonic transform, and documented the harmonic transform. However there was always something else to add. Graduate students and young faculty, are constantly building models of geophysical processes, which require other quantities. They needed what are called "derived" quantities. For example, the harmonic transforms provide the spectral representation of, say, a pressure field in terms of the spherical harmonics or a velocity field in terms of vector harmonics. These models need derived quantities such as the divergence, gradient, or the Laplacian of these fields. So at first these would be provided from my private stock. However, after doing this a few times it was clear that I could save both myself and the user some time if these codes were documented and made available in a package. Eventually, this evolved into a "model development facility" that included most of the derived quantities required for models of geophysical processes. Graduate students and young faculty could build a model very efficiently and very rapidly just by incorporating these derived quantities into their program. That, basically, is how SPHEREPACK was built.

HAIGH: So the description I saw of SPHEREPACK included the text that it "contains programs for computing certain common differential operators including divergence, vorticity, gradients, and the Laplacian of both scalar and vector functions."

SWARZTRAUBER: That's correct.

HAIGH: What is it about the kinds of research that were under way at NCAR that meant that there was a need to use this kind of geometry and these operators with such frequency? In other words, what does the nature of those geophysical processes have to do with the need for these specific kinds of mathematics? And would those same functions be needed by many other kinds of researchers, or is the need fairly specific?

SWARZTRAUBER: The earth is most closely approximated by a sphere and therefore geophysical processes are most appropriately modeled using spherical coordinates. Processes are first described in terms of the equations that they satisfy and it remains to solve these equations to determine the process itself. For example, the atmosphere is known to satisfy a somewhat complicated system of partial differential equations in which its evolution is given in terms of spatial derivatives of pressure, velocity, temperature, and so forth. In Cartesian x, y, z space, these derivatives are readily determined using FFTPACK and differentiating the resulting trigonometric representations. However, the equivalent in spherical coordinates is quite different. The Fourier transform is replaced by the harmonic transform and the spatial derivatives are considerably more difficult to compute because, in fact, they may be unbounded if the equations are not carefully written [P. N. Swarztrauber, The approximation of vector functions and their derivatives on the sphere, *SIAM J. Numer. Anal.*, **18**(1981), pp. 191-210]. SPHEREPACK takes care of all this by computing the appropriate spatial derivatives that can be substituted into the equations, which can then be integrated in time to compute the evolution of the process under consideration.

SPHEREPACK is applicable to any process that is posed on the sphere such as the earth and planetary atmospheres, oceans, seismic, oil exploration, and our sun and stars. But there are other applications. I think I mentioned earlier the company that was building sound systems for big auditoriums, where the sound wave propagates from its source in a spherical way. On average, about five copies of SPHERPACK are distributed weekly.

HAIGH: So you've mentioned that over time you've produced an increasing number of these functions. What was the perceived benefit gained by bundling them together and calling them a "PACK"?

SWARZTRAUBER: It provided a single place where researchers could obtain all the derived quantities necessary to model the process of interest. It's important to add that SPHEREPACK is a research tool. It provides a quick way of investigating a process. Although the big models may incorporate the computational methods implemented in SPHEREPACK, they are usually implemented in much more extensive custom software.

HAIGH: So it would make it easier to tell people where to look, that you would refer them to the whole package instead of just one function?

SWARZTRAUBER: Yes, the SPHEREPACK web site begins with a table of functions with corresponding subroutine names. The table also lists the other files that are required and the resulting tree is very short. So it is relatively easy for the user to select the appropriate programs and files. It also eliminated requests by users who may have thought they needed one program when in fact they needed another, or users who, after seeing what is available, may choose an alternate modeling approach. Most users require multiple functions or codes anyway, which would be inconvenient to collect and provide separately.

HAIGH: So then the packaging process was motivated by internal use? It wasn't that making it into a pack was something you did primarily in order to get it distributed outside of NCAR?

SWARZTRAUBER: Probably for both. Internal users had the advantage that they could just call the subroutines directly without worrying about what called what. This was also true of institutions that installed the package.

HAIGH: How did the collaboration with John Adams work on this project?

SWARZTRAUBER: John developed and tested the drivers. I focused on the core computational methods, the scalar and vector harmonic transforms and documentation. It was a great time. We worked very well together over an extended period of time. John was more involved in SPHEREPACK than FISHPACK. He came later to FISHPACK and earlier to SPHEREPACK.

HAIGH: Was it again the case that you were making contributions to the actual mathematical methods involved in the package?

SWARZTRAUBER: Yes, it followed the pattern established by FISHPACK where the research preceded the software.

HAIGH: I'm seeing a 1989 publication with G. L. Browning and J. J. Hack, "A Comparison of Three Numerical Methods for Solving Differential Equations on the Sphere". [G. L. Browning, J. J. Hack, and P. N. Swarztrauber, A comparison of three numerical methods for solving differential equations on the sphere, *The Mon.Wea. Rev.,* **117**(1989), pp.1058-1075].

SWARZTRAUBER: Actually, that paper is a prime example of using SPHEREPACK to do some numerical studies. At that particular time, that was relatively early, I would have been using private stock codes that I had developed for the harmonic transforms and derived quantities.

But to answer your earlier question, like the Fourier transform, the harmonic transform existed in the literature but its computation was another matter. There were some aspects that were radically different than the discrete Fourier transform. For example, the discrete harmonic transform is not a one-to-one transform because there are fewer harmonic coefficients than points on the sphere. This was a little mind boggling. At first I thought the result was wrong and in fact tabled the research until it became overwhelmingly evident why the result was true. [P. N. Swarztrauber, On the spectral approximation of discrete scalar and vector functions on the sphere, *SIAM J. Numer. Anal.,* **16**(1979), pp. 934-949.] Then there was the little matter of vectorizing the harmonic transform and developing the software. It was always comforting to know that the research that you were publishing had proven itself on the computer.

The computer itself provided research directions. As they became more powerful, model resolutions increased and errors began to creep into the calculations. Phil Rasch brought this problem to my attention. It was traced to the accuracy of the Gauss quadrature. I was puzzled by this because it used the QR algorithm, which was thought to be very accurate. Nevertheless it was clear that the QR algorithm was not giving what was known to be the best possible accuracy. In the end, we were able to compute Gauss points and weights with an error that was the square of that provided by the QR algorithm. Not just half or a tenth of the error but the error squared! [P. N. Swarztrauber, Computing the points and weights for Gauss-Legendre quadrature, *SIAM J. Sci. Comput.,* **24**(2002) pp. 945-954]

Here's another paper [P. N. Swarztrauber and W. F. Spotz, Spherical harmonic projectors, *Math. Comp.* **73**(2004) pp. 753-760] that was associated with minimizing storage required for harmonic projections. This reduced storage by an order of magnitude.

This paper [P. N. Swarztrauber, Shallow water flow on the sphere, *Mon. Wea. Rev.,* 132(2004) pp. 3010-3018] was one of my most recent papers. This is a second example of simply using SPHEREPACK to develop a model for investigating the flow of fluids on the sphere. Bits and pieces of SPHEREPACK were used for this research as well.

HAIGH: That would be the same process where you would scour the literature for the best methods and test them and implement them rather than taking any externally produced code?

SWARZTRAUBER: Yes, where they could be found; however, both FISHPACK and SPHEREPACK would not have been possible if we had to rely strictly on the methods that were available at that time. And of course codes were not available in the sense that they could be "plugged in" to form such packages. Much research was required before the concept of such packages could be considered a reality. In those days, NCAR led the research that resulted in these packages as well as the software development. This was evident by the response from the scientific community, not only to the packages but also to the computational methods used to implement them on NCAR's state-of-the-art vector computers.

HAIGH: So, I'm trying to get the date on this. I'm seeing an NCAR technical note on SPHEREPACK 2.0 dated September 1997, and there appears to be a version 3.0 now and actually a version 3.1 August 2003. So that's pretty recent. Do you remember when version one would have appeared?

SWARZTRAUBER: The first paper on the harmonic transform was published in 1979 so bits and pieces would have been around at that time. It would have evolved much like FISHPACK with new functionality being added from time to time. My guess is that first version of SPHEREPACK would have been in place by the mid 80s but without John's drivers that resulted in version 2 [J. C. Adams and P. N. Swarztrauber, SPHEREPACK 2.0: A model development facility, NCAR Technical Note NCAR/TN-436-STR, September 1997].

HAIGH: You had said that with the other packages, basically people would hear about them. You would give away copies; they would spread that way.

SWARZTRAUBER: Well yes, and also from presentations.

HAIGH: I think you had also said earlier that by the 1990s that NCAR was becoming more concerned with protecting its intellectual property. Do you remember any issues involved with disseminating SPHEREPACK?

SWARZTRAUBER: Yes, I think at that point the programs contained information copyright notices and information about how commercial users could obtain permission to use the codes.

HAIGH: So you were still able to give the software away, but you would have to give it away with a license that said it couldn't be used commercially?

SWARZTRAUBER: I don't recall any research organization ever having to obtain a license. We would just take their word for it, they could go ahead and download it and use it in their research. However, commercial organizations were asked to obtain a license.

HAIGH: I would think, though, that if you wanted to restrict someone's ability to use it commercially a license for all users would be the usual way to do it. I suppose you could do that with copyright to some extent, but you often agree to a license statement even for software that you don't pay for. You have to click before you download, "Yes, I promise I won't use it commercially. I won't disassemble it. I won't make any changes to it. I won't export it to China. I won't hold you legally liable for anything that might happen. There aren't any warranties." Click, and then you download something. So, it's not that a license is just a way to get money, but also it's a way to legally restrict people's ability to use it in certain ways, such as to redistribute it commercially.

SWARZTRAUBER: You know, now that you mention it, I think the user must click on a button that says, “I agree to the terms.” I had just forgotten about that.

HAIGH: Actually, what should I find here on the Net but the SPHEREPACK licensing agreement? It’s a couple of pages. There are some big capitals “YOU ARE GOING TO BE BOUND TO THE TERMS OF THIS LICENSE. IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE, YOU ARE NOT AUTHORIZED TO DOWNLOAD THIS SOFTWARE.” So you agree that the software is owned by the University Corporation for Atmospheric Research, UCAR. You agree that you can use it for educational research and nonprofit purposes only. You can make a copy of it, but you must include this copyright information. You can’t export it outside the United States except as authorized by United States law. And there are no warranties whatsoever. So presumably, this would be something that UCAR would have come up with in general?

SWARZTRAUBER: Yes.

HAIGH: So you didn’t have to worry about that too much then?

SWARZTRAUBER: Right. Evidently (smile)

HAIGH: And then when you click “I agree,” you are able to download a copy of the source code.

SWARZTRAUBER: Right. And I have mixed feelings about that, but that is the kind of world we live in now. I was so protected from this. In general, the researchers at NCAR were protected from matters other than research, for years and years. Well, protection is not the right word. I think we were insulated from such things. But NCAR has to live in this new world.

HAIGH: So, we’ve talked about those packages now. Let’s return and talk more in general terms about your career at NCAR. That seems to lead on from what you were just saying. Now, it appears that with this shift after you got your Ph.D., into more of a research function, from 1975 onward you were a “staff scientist” of various grades, and from 1982 you were a “senior scientist.” Did those changes in job title reflect any actual change in the work you were doing?

SWARZTRAUBER: It was recognition of the level of work I had reached at that particular time more than anything, and I appreciated it. I had some hiccups later in my career. I took on acting management of the division once for about six months or something like that, and I was so fortunate because it was an acting position and realized very quickly that management was just not something that was very fulfilling for me. I wasn’t using my many years of experience and education and not enjoying my job. I developed a great admiration and respect for good management. Research and management are at opposite ends of the spectrum. Research is tidy when compared to management.

It was six months that was sort of cut out of my life. I think that perhaps it saved me from making a more serious mistake later in my career. After that, I wasn’t the least bit interested in management. As a result, although I didn’t consciously make the decision, I was able to maintain my technical skills. Management at NCAR was relatively short term.

HAIGH: And what would happen to people after that?

SWARZTRAUBER: Well, they would just move on to positions in other sections or at other institutions. Management is a tricky business because with every decision there are some folks who are happy and some who are not and pretty soon just about everybody is unhappy with some decision.

[Tape 4, Side A]

HAIGH: So when the last tape ran out, you were just discussing the problems involved in being a manager in a place like NCAR.

SWARZTRAUBER: Well, I don't think it's just NCAR. There was probably more turnover in management positions than in any other position at NCAR. For a long time at SCD the average tenure of managers was about four to five years, which was relatively short. Now, there were exceptions to that, Buzbee was there for eleven years as an excellent manager. But the managers have a lot of stress because people are beating on them eight or more hours a day. The quality of management decisions is important. If they are viewed as fair, then even though you might personally be unhappy you can still respect the decision and the manager. But for me, management was extremely disorienting and unpleasant. It was very difficult, if not impossible, for me to make decisions that might adversely affect people that I had worked with for years. I was also very concerned about losing my technical edge and not using the education and experience that I developed over many years. The diversion from research and not being able to look forward to my job on a daily basis was very unpleasant. I learned a valuable lesson relatively early in my career.

HAIGH: So I'm seeing the period as acting manager of the computing facility was 1973. As acting director of the computing division of NCAR, what particular issues did you face and what contributions did you make to its direction?

SWARZTRAUBER: As I mentioned, it was not my favorite time at NCAR. It was during this time that an organizational structure was put into place. This included the establishment of sections such as Operations, Systems, Applications, and so forth. This was a good idea because the size of the computing group had outgrown a single person type management. The difficult part was the classification and grading of the staff as programmers I through IV. It was not easy for the staff and gave me some sleepless nights. Joe Oliger was deputy at that time and took a lot of heat during this transition. It was also the beginning of noticeably burdensome paperwork. It was also during this time that we decided to purchase of the Cray-1.

HAIGH: Now, you do have on your resume from 1974 to 1987 the title "Manager, Advanced Methods Group, Computing Facility."

SWARZTRAUBER: Ah, well that was simply a title, because there were maybe half a dozen people who were highly motivated. And you simply let them alone, and they did really nice things. In fact, those people worked for me, and then we sort of swapped the job. First they worked for me and then I worked for them [For example Dick Sato]. It was always a team no matter who was in "charge". It never occurred to me to make my day much different than it would have been if I were not a manager, except perhaps for a slight increase in paperwork.

HAIGH: Now, over the time that you were working at NCAR, were there any reorganizations or shifts in the organization of the computing facility that had any real impact on the way that you did your job or related to users?

SWARZTRAUBER: Well, me personally, I don't think so. At my retirement party, Bill Buzbee introduced himself as "Paul's supervisor," and then added, "That is, if Paul ever had a supervisor." Hopefully, in the course of my career, I was able to, on my own; define the directions of my research and projects in such a way that I made significant contributions to the atmospheric science community and thereby enhance NCAR's reputation. I think that permitted me to have the freedom that I had at NCAR. Research directions were established by the senior scientists at that time. I have some sadness in knowing that this is no longer true; namely, research directions are established by management, usually in the direction of grant monies. There was a time at NCAR when supervisors were able to personally evaluate the quality of your work but now it's pretty much like most other institutions that have and hire professional managers. The tradeoff is between a professional manager with minimal technical skills and an amateur manager with good technical skills. In looking back, it seems the latter was more successful than the former; however, probably the most important factor is the person.

HAIGH: Over the same period, there have obviously been a number of changes in the kinds of computer systems that are in use. You've already discussed the arrival of vector machines and the CDC machines that you worked on. Did the arrival of mini-computers and then subsequently of work stations in personal computers to distribute computing power have any kind of influence on the work that you were doing?

SWARZTRAUBER: Well, workstations, of course, there was a dramatic improvement in the efficiency of the work. I mean, we had word processors at our fingertips. You didn't have to submit a revised manuscript to a typist or an editor. It was a huge difference in terms of the convenience of writing a paper, or for that matter, submitting a program. I mean, to modify and submit a program from your desk was so convenient and considerably more efficient than punching and shuffling cards. The efficiency of manuscript preparation was also greatly improved. In most cases, you could do a much better job because you didn't worry about the number of changes that you made. You could move a whole paragraph around, and build, or for that matter "debug" a manuscript just like a program. I recall spending days in the library looking for related research in the citation index, a task that can now be completed in a tenth the time and at your desk.

HAIGH: How about challenges for the kinds of package that you were producing? For example, if people were running your code on workstations instead of on centralized computers, did that create problems in terms of portability or make it harder to educate the users about how to use the software, for example?

SWARZTRAUBER: It probably made it easier because you could just email the files. Surface mail to Russia used to take a month. Email changed this to a matter of hours or even less. Communication was improved by an order of magnitude, particularly with international colleagues. I'm certain you would get the same story from everybody. I don't think it's surprising.

Portability was not a problem at first because the SUN workstations were running standard software. At first, programs continued to be run on the main frames and only submitted from the workstations. However this evolved into programs being developed and checked out on the workstations and then submitted to the main frame for longer “production” runs. SUN workstations had excellent FORTRAN compilers and dominated the workplace for a long time. However, SUNs began to be replaced by Macs and PCs on employee’s desks. This was slowed by 32-bit arithmetic, which meant that some development might still be done on the main frame. Workstations also made it convenient for users to contact either John or me if they had problems. Now, if you’re talking about multiprocessor computers, that’s a totally different story.

HAIGH: Yes. I’m just working up to that, so I’ll ask you about that in a second. In 1993 NCAR became the first, and indeed only, site to receive a Cray 3, the machine developed by Seymour Cray at his new firm Cray Computer Corporation which was based in Colorado. How did NCAR's relationship with Cray develop during the 1990s, and what was NCAR's experience with the Cray 3?

SWARZTRAUBER: Lets backup a bit. Before Cray Computer Corporation in Colorado Springs, Cray Laboratories was established in Boulder, headed by Stu Patterson who was previously the head of computing at NCAR. They built a very nice laboratory complete with a clean room. The footprint of the building was an annulus segment and not just the usual box, a nice architecture. My recollection was they were looking at multiprocessing but whatever they were doing they didn’t do it long because the lab lasted only a few years before Seymour left the project. The Cray 2 came out a few years later followed by the Cray 3 [Cray Labs: 1979-1982; Cray 2: 1985; Cray 3: 1989].

Now in so far as the Cray 3 is concerned: my recollection is that eventually NCAR got useful cycles out of it. I was on the periphery of this but my sense was that the Cray 3 was eclipsed by the euphoria surrounding massively parallel processing at the time. [See the Appendix for contributions from Jim Hack, Tom Engel, and Bill Buzbee.]

HAIGH: Now, I know for NCAR as a whole, the controversy over the attempt to acquire the SX4 computer from Japan was quite an important development. So now it seems your work on massively parallel computers started before that, so I’ll ask you about that in a second. Obviously for Buzbee personally, that was quite an important and wrenching kind of experience. Did it filter through the lab as a whole?

SWARZTRAUBER: Oh yes. I was incredibly dismayed. I knew the capabilities of the individuals who made the selection and one could not have chosen a better team. It was a real shock to me, a real eye opener. I remember thinking ‘There’s no way that NCAR is not going to get the best computer available.” I was just used to that, one right after the other, the 3600, 6600, 7600, the Cray-1, XMP, and so forth. Like any first rate research center, we always had the best computers. I was naïve.

HAIGH: Had the morale and productivity recovered from that by the time you retired, with improvements in massively parallel systems as the alternative? Or do you think that even in 2004 that the lab hadn’t fully recovered?

SWARZTRAUBER: Well healthy people adjust expectations and go on. It's just part of the new world we live in and NCAR as well as the other labs have adjusted to it. Nevertheless it's been a long time, since we've got a computer that was just simply, boom, five times faster. The new generation of multiprocessors may be five times faster on paper because they have more and faster processors. But in practice, true multiprocessing runs at the speed of the communication channels, actually slower because of contention, bottle necks, protocols, and so forth.

HAIGH: So that brings us naturally to the CHAMMP Project, and then you're patent for the multi-pipeline multi-processor.

SWARZTRAUBER: The CHAMMP Project was certainly an outstanding project. It brought together people who were extremely knowledgeable about parallel computers and also very knowledgeable about modeling geophysical processes including Jim Hack and Dave Williamson. Initially the goal was to examine the impact of massively parallel multiprocessing on weather and climate modeling. It was a joint project with Oak Ridge. We were joined later by Argonne and Los Alamos. The models were implemented on the Connection Machine, built by Thinking Machines Inc.

HAIGH: Give me some background on that, then. When did parallel computers first appear at NCAR?

SWARZTRAUBER: I think the first time we ran was probably in early 1989. Dick Sato and I went to Thinking Machines in the Boston area [Cambridge] and implemented a very simple weather model on the Connection Machine. Of course we got a lot of support from Thinking Machines, in particular, John Richardson. Oliver McBryan at CU also gave us a hand. We worked on it day and night. By the end of about three days, we were encountering the "communication problem". On paper, the machine was very powerful as determined by the number of processors times the speed of the individual processor. But this ignored the time required for communication between the processors, which severely degraded performance. It was degraded to the point that performance was not competitive with the more traditional computers that were available at that time. Furthermore, the simple model that we used was quite multiprocessor friendly. The actual weather models require more interprocessor communication and performance would have been degraded even further.

HAIGH: And would this machine have had something in the order of a hundred processors?

SWARZTRAUBER: Oh, I think it had a couple thousand floating point processors. The processors were retrofitted to the original Connection Machine that had one bit processors that was built for artificial intelligence applications. The retrofit required circuitry that further increased interprocessor communication, which was at least partly responsible for the poor performance. If the Connection Machine 2 had been redesigned from scratch it is quite possible, indeed likely, that it would still be the architecture of choice.

I remember that we'd get on the phone with Buzbee and outline the problem. He was very disappointed in the results. We had also been working with Oliver McBryan at the University of Colorado. So I said, "Bill, let's get Oliver out here because he has had experience with this machine and maybe he knows something we don't. Lets ask him to come out and work with these people" So Oliver came out and we all worked on the project but communication continued to be a problem and performance could not be improved to any appreciable extent. Later, with the establishment of the CHAMMP project, Oak Ridge, Argonne, and Los Alamos all put people on the project. We would have quarterly meetings to discuss progress and the development of algorithms that were specifically designed to speed multiprocessor communication. But communication continued to dominate total compute time. It was not unusual for communication to amount for over 90 percent of the total compute time.

So Bill began to rethink massively parallel computing in terms of its future at NCAR, at least its future in terms of workhorse machines. That was the beginning of a reassessment of the impact that massively parallel computing was going to have on NCAR. Other companies began to build "massively" parallel computers but the results were the same. There must have been five or six companies. I remember Kendahl Square as one, but the story was always the same. A performance analysis of say, an FFT, would have revealed the problem before they so much as raised a soldering iron. But that didn't happen for any of the machines, and still doesn't. It's not too surprising because it's an exercise would have brought some bad news. [P. N. Swarztrauber and S. W. Hammond, A comparison of optimal FFTs on torus and hypercube multicomputers, *Parallel Computing,* **27**(2001) pp. 847-859]. We had a few multiprocessor machines, and got some useful cycles, but they didn't replace the traditional architectures as workhorse machines until the government stepped in and forced us to do so. But of course that is a whole other story.

HAIGH: So then prior to these visits, the assumption had been that there would be an enormous improvement.

SWARZTRAUBER: Oh, yes. Everybody was excited. That was the direction it was going in, and we were not thought of as very nice people because we were bringing some bad news to the high performance computing community. It was shortly thereafter that I got this call from Bob Ward.

HAIGH: So just before we move onto that, did NCAR actually order any of these machines?

SWARZTRAUBER: We had a Connection Machine 2 and later a Connection Machine 5. I would say this about the Connection Machine: it was a fantastic machine in terms of a development tool. Prior to that time the focus had been on parallel computational algorithms dating all the way back to the 70s. But in practice it was clear that communication was ultimately going to dominate compute time and there was a shift from the development of parallel computation algorithms to parallel communication algorithms. This came about quite naturally because we were really motivated to reduce communication time. The resulting body of work is directly related to our experiences with the Connection Machines. In fact, it turned a good sized fraction of my research around. Instead of parallel computation algorithms, I began to work on parallel communication algorithms. [P. N. Swarztrauber, Transposing arrays on multicomputers using de Bruijn sequences, *J. Parallel Distrib. Comput.,* **53**(1998) pp. 63-77.]

HAIGH: So you just mentioned a phone call from Bob Ward at Oak Ridge National Laboratory.

SWARZTRAUBER: That started the CHAMMP Project. Bob had heard that we were putting models on the Connection Machine and was interested in joining the project. Oak Ridge had a lot expertise in parallel algorithms so I said, "That's fantastic." We began to have regular meetings about the models and their performance on massively parallel machines. It was a very exciting time. Shortly after that Argonne joined the project and shortly after that Los Alamos joined. My recollection is that the different Labs would “adopt” different parallel computers that began coming on the market at that time. It was possible to compare the performance of these machines because the same atmospheric models or parts of these models were being implemented.

HAIGH: Was that kind of collaboration with the DOE National Labs an unusual thing for NCAR, or had it been going on in other areas as well?

SWARZTRAUBER: There had been some collaboration between NCAR and Los Alamos on the development of climate models independent of MPP. I believe that DOE had been supporting Warren Washington’s climate modeling group at NCAR. However this work was on traditional shared memory architectures and did not involve the labs to the extent that the CHAMMP project did. So DOE had already demonstrated an interest in climate and weather models. The CHAMMP project involved four Labs, actually five; I think the Pacific Northwest Lab also joined the project. It was very successful. I credit Bob Ward for doing a terrific job of putting it together and keeping it humming. The group was honest about their results and very straightforward and dedicated to the common goal of making these machines work and making them run fast. There was absolutely zero politics, well at least very little. In that sense it was indeed unusual for the Labs or for that matter any group of institutions to be collaborating in this manner. Funding agencies promote collaborative proposals for many good reasons including the fact that they like to distribute grant monies as broadly as possible. So institutions submit collaborative proposals that sometimes basically result in the funding of individual research. CHAMMP was definitely not in that category. And I really credit Bob Ward for having set the tone. And also, of course, Jim Hack and Dave Williamson who would keep us on track and make sure that we were implementing models that could ultimately be used for predicting weather and climate.

HAIGH: And that CHAMMP Project attracted DOE funding of its own?

SWARZTRAUBER: That is correct.

HAIGH: So how long did it run for?

SWARZTRAUBER: I think it’s still running. The focus has changed from massively parallel machines to climate models on a variety of platforms and grew to include a number of university projects. It’s called something else now, and I don’t remember what [Climate Change Prediction Program]. But it’s a considerable sum of money dedicated to weather and climate research, which is of considerable benefit to NCAR, and the atmospheric science community in general

HAIGH: Are there any of your publications directly related to that project?

SWARZTRAUBER: Here’s something: [C. Tong and P. N. Swarztrauber, Ordered fast Fourier transforms on a massively parallel hypercube multiprocessor, *J. Parallel and Dist. Comput.,* **12**(1991), pp. 50-59]. Although I don’t think that was on the Connection Machine, it would have been using things that we learned on the Connection Machine.

Here's another one, the title is [P. N. Swarztrauber, MPP: What went wrong? Can it be made right?, *In:* Proceedings of the Seventh Workshop on the Use of Parallel Processors in Meteorology; Making its Mark: The Use of Parallel Processors in Meteorology, G.-R. Hoffmann and Norbert Kreitz, eds., World Scientific Publishing Co. Pte. Ltd., London, 1997]. That was a talk at the European equivalent to NCAR, ECMWF.

Here's another one: [P. N. Swarztrauber, Transposing arrays on multicomputers using de Bruijn sequences, *J. Parallel Distrib. Comput.,* **53**(1998) pp. 63-77] again focused on communication. The CHAMMP project had a significant impact on the direction of my work. I was always interested in parallel computation but following CHAMMP my interest switched to parallel communication.

Here's another one [P. N. Swarztrauber and S. W. Hammond, A comparison of optimal FFTs on torus and hypercube multicomputers, *Parallel Comput.,* **27**(2001) pp. 847-859]. Here we asked and answered the question: "What is best possible performance?."

And of course all the work on the Communication Machine began following the communication problems that were encountered on the Connection Machine. [P. N. Swarztrauber, The Communication Machine, *Int. J. High Speed Compt.,* **12**(2004) pp. 65-82].

HAIGH: So to what extent would you say that the lessons learned from the CHAMMP Project were put into practice at NCAR, to improve the results that people were getting with the Connection Machine?

SWARZTRAUBER: We were repeatedly frustrated by the fact that, although we had developed algorithms that minimized communication, we were not able to obtain more than about 5 percent of peak performance. So although we were able to improve the performance of that machine, we were unable to make it run sufficiently fast to justify the effort required to port legacy codes to that platform. Its main contribution was to identify the "communication problem" and point multiprocessor research in that direction. The fact that NCAR did not adopt MPP as the workhorse machines was not immediately understood by the computer science community who were just getting on that bandwagon without actual experience with the machines.

Indeed, for awhile we were viewed as just not knowing what we were doing. It was suggested that if we were to use different algorithms or coding practices that MPP would prevail. An NSF panel was formed to look into our performance studies. I don't think this got anywhere because Oak Ridge, Argonne, and Los Alamos had obtained the same results. Nevertheless it was a difficult time in the history of computing at NCAR.

HAIGH: So as I understand the implications of that answer, you're saying that it was not just that you might have been able to work around some of the limitations of the Connection Machine, but more that the experience that was gained collectively would help steer future machine developments towards architectures that would avoid those problems?

SWARZTRAUBER: Absolutely. I was a little disappointed that CHAMMP did not shift gears at that time and work with the manufacturers to rectify the problem that was so evident by that time. I wrote a letter to the DOE suggesting that the labs join forces with the manufacturers to remedy the communication problem, but by that time interest and funding for MPP had essentially dried up. This was truly an unfortunate turn of events at that time, much more so than the fact that NCAR was unable to purchase the computer of its choice. The impact has been widespread and long term. In spite of its poor communication performance, the CM2 came very close to breaking even with traditional architectures. But this was not good enough to warrant the huge investment required to port a large number of legacy codes to its multiprocessor platform. CHAMMP was in a position to describe the hardware changes that would have made the successor to the CM2 the machine of choice. Without formal CHAMMP input, Thinking Machines chose to simplify communication from the programmer's point of view by installing automatic routers in the CM5, which in the final analysis aggravated the communication problem.

HAIGH: All right, and I imagine that this is now going to lead to your idea of a new communication oriented architecture for massively parallel machines. So after lunch, we'll do that and then talk about your involvement in various committees and professional organizations.

[Tape 4, Side B]

Beginning of Session Four, held on the afternoon of July 17, 2005, in the Westin Westminster.

HAIGH: So I understand that you have a couple of things that you'd like to add to our previous discussion of the SX4 procurement travesty.

SWARZTRAUBER: Yes. I mentioned earlier that NCAR couldn't have picked a better team to evaluate the computers. Of course we had an outstanding record for selecting traditional architectures and at that point a lot of experience with MPP. The CHAMMP Project had been underway for quite some time, and we had been working with the best people at Los Alamos, Argonne, and Oak Ridge. So we had a very good idea of the capabilities of both MPP and traditional architectures and able to make as good or better selection than anybody. The other thing I would like to say is that when NCAR was brought to task about the SX4 decision, and Bill testified in Washington, he was the only one there without any vested interest whatsoever, other than to provide the atmospheric science community with the finest computing available. I think that's basically it. The only things I wanted to add.

HAIGH: So in that sense, were the other representatives from the massively parallel super computer producers?

SWARZTRAUBER: They were from companies with a lot at stake. And of course, that's why they were there and why NCAR was brought there. Now, I don't know the details. Bill had the direct experience, and I only observed from the side lines but as someone knowledgeable about traditional machines and MPP via the CHAMMP project. I wanted to add those things, Tom, because the more I think about it, Buzbee got a very raw deal and I'm please to be able to go on record to support that decision. When the politics became evident to the other labs, they understandably did not step forward.

HAIGH: By the way, did CHAMMP stand for something?

SWARZTRAUBER: Yes, Computer Hardware Applied Mathematics and Model Physics. Indeed informally it was expanded to CHAMMPions, which stood for the CHAMMP Interagency Organization for Numerical Simulation, which was contributed by Bob Ward.

HAIGH: Now, I imagine that as a result of your experiences on that project that you became interested in ways of producing architectures that would cope better with these communication issues, and I know that you received a patent for working on it?

SWARZTRAUBER: Yes, that's right. It was very interesting, because I remember when we were working on the CM2, there was a key instruction that was missing. Considerable time could have been saved if transient data could have been kept in a local buffer before it was forwarded to the next processor. However, data was written to, and read from, memory before forwarding. I talked to TMC about this and other communication delays, and it became apparent, at least to me, that the CM2 was developed with a somewhat hasty retrofit of floating point processors to the original single-bit massively parallel computer. It was also evident that Hillis' SIMD architecture was very valid. It was the implementation that was the problem. Had a "pass through" instruction been available and the "retrofit" cleaned up a bit, communication could have been sped up to the point that the machine would have definitely had a future in weather prediction. We would be computing on one of its descendents today.

HAIGH: SIMD in that context would mean basically the vector approach, would it?

SWARZTRAUBER: SIMD means that you have a single processor that sends the same instruction to all processors that execute it simultaneously. So it basically amplifies the power of that single processor. Supposedly it multiplies the power of the single processor by a factor equal to the number of processors. SIMD is a quite reasonable concept. However, because its implementation was so poor, the baby got tossed with the bathwater. I mean, you just didn't talk about SIMD architectures anymore because SIMD was irrevocably bound to the Connection Machines.

Now connecting a bunch of floating point processors to increase compute power, was what everybody had in mind at that time. This seemed like just a great idea. However, when it came down to it, the speed of these machines was really limited by the much slower speed of the communication channels. Indeed, in practice, communication could occupy over 90 percent of total compute time. Therefore, instead of focusing on a chip that performed computations rapidly, it seemed reasonable to focus on one that performed communication rapidly. One with a communication unit along side the other units that comprise the CPU. In terms of hardware the key is to be able to switch and forward incoming data as quickly as possible because most data is transient and on its way to its destination processor.

The second thing that was realized was that for every computation, there is an underlying communication algorithm. An algorithm that brings the operands together in the same processor so that you can add, subtract, multiply, or do whatever you want. To make effective use of the multiprocessor this must occur in all processors simultaneously. This requires some very careful scheduling, which produces the algorithm. The hardware must then be able to implement the algorithm by executing communication instructions that forward incoming data on channel A to outgoing data on channel B. When the algorithm is finished and the operands are in the same processor then it's pretty straight forward to add them or do whatever. Programmable communication is required to implement the scheduling algorithms that determine the performance of multiprocessors to a much greater extent than computation.

There was a third realization. Schedules or algorithms exist that provides a sort of vector “like” performance across the processors. On each communication cycle, every processor receives data intended for that processor. This is significant because most data is transient and passes through the processors. What is different is that at least one datum arrives at its destination in every processor on every cycle. Furthermore this result is independent of the number of processors. These algorithms are complex but then no more complex than certain computational algorithms such as the fast Fourier transform. Actually it was this third realization that drove the first two. So I began putting these three observations together and designed the Multipipeline Multiprocessor, which was subsequently patented. The current version is called the Communication Machine, which was referenced above and integrates the memory system.

It ran into a lot of resistance. Outside of perhaps a couple dozen or so folks, the concept of a parallel communication algorithm was not well understood. It was not even an issue with most machines the routed data automatically. So I would routinely be asked, what is a communication algorithm? To understand, develop, and program parallel communication algorithms presented a rather steep learning curve. Also the accepted dogma at the time was something called “desktop to teraflop”, which meant that supercomputers could be built using the same processors that are in personal computers. This concept was at complete odds with processors that are designed to efficiently handle the communication requirements of a massive number of processors. Companies are not motivated to invest the kind of money that is required to develop such processors, particularly when they are unclear about how to use them. I know that’s windy Tom, but it’s actually the short version!

HAIGH: So to move into this area, did you have to start reading up on computer architecture, involving yourself with the community of people who were thinking about communication issues?

SWARZTRAUBER: Excellent question and part of the problem. I knew a lot about parallel computational algorithms. That was a hobby of mine over the years. In fact a lot of people have written on that subject, because of the general belief that the number of parallel computations was going to determine parallel computer performance. However in practice it was parallel communication that determined performance. So, as I mentioned, my focus changed from parallel computation to parallel communication, and I began looking at that particular subject with the understanding that it was really an algorithmic problem.

I’m sorry Tom, I forgot. What was your question?

HAIGH: Involvement with the literature and community of people already involved in computer architecture.

SWARZTRAUBER: Sort of. I was very familiar with instruction sets and functionality so I was in a position to define the machine at that level. I also had some knowledge of digital circuitry but definitely not at the state of the art. Nevertheless, the fact that I was not an architect damaged my credibility with that community. So we invited Martin Herbordt to visit NCAR to asses the difficulty in building such a computer. He tried to teach me a little bit about circuitry but I was not a very good student. He was patient. He concluded that it would be relatively straightforward to build the circuits that would provide the desired functionality. Indeed the circuitry would be quite a bit simpler than a state-of-the-art floating point processor.

I also talked to the people who were developing communication algorithms. The Connection Machine motivated research on communication algorithms during that era, which unfortunately passed much too quickly. Nevertheless there was a small group of researchers that wrote some very interesting papers [Lennart Johnsson, C.-T. Ho, Youcef Saad, Martin Schultz, Quentin Stout, Oliver McBryan, and Paul Fredrickson].

HAIGH: You mentioned earlier that you were disappointed that the patent was eventually sold.

SWARZTRAUBER: Oh yes. It was so sad, I mean from my point of view. The intellectual properties group at NCAR was probably equally frustrated. They contacted a number of vendors. But as I mentioned earlier, they had all gone in the opposite direction; namely trying to develop routers and things that would handle general communication automatically without having to program communication algorithms. And of course the “desktop to teraflop” is still the basis on which multiprocessors are built.

HAIGH: So the patent was sold rather than licensed.

SWARZTRAUBER: That’s correct. It was bundled with two other patents and sold as a package. That ended my involvement with the patent. However I still work on the design and communication algorithms from time to time.

HAIGH: They weren’t able to interest any business in it. On an intellectual level, did you find any interest from researchers in computer architecture or design in this kind of approach?

SWARZTRAUBER: Well, there were a few; namely, those who understood the communication problem and had some knowledge of the algorithms. For example, a proposal was submitted to NSF with the intent to further develop the design as well as the algorithms. It was almost funded. You could look at the reviews and clearly identify the reviewers who understood the approach and those who did not or perhaps more accurately, those who understood the algorithmic approach and those who did not. One review was all excellent, which is quite unusual in terms of reviews. Overall it rated between excellent and very good. However there is a lot of competition for NSF funding and it did not manage to squeak by.

HAIGH: Any more thoughts on that topic?

SWARZTRAUBER: Well, I could go on for a long time, but I think we should probably move on to the next topic.

HAIGH: All right, well it seems the main area that we still have to deal with would concern your more general involvement in the mathematical software community and with the different professional associations. We’ve already talked a little bit about your involvement with ACM SIGNUM and with the Rocky Mountain Chapter. Now, it seems that from your resume that chronologically, your next important involvement would come with SIAM, that from 1977 to

1984 you were serving as a SIAM Lecturer. How did you find that experience, and what kind of topics would you have covered?

SWARZTRAUBER: Well, of course, the topic would have been computational mathematics including: partial differential equations, parallel computing, harmonic transforms, solutions of elliptic equations, Fourier transforms and so forth. SIAM did a nice thing; they published a list of professors and scientists who would be willing to lecture for the price of their expenses. I did that for a few years. My recollection is that I gave perhaps a half dozen lectures under that program.

HAIGH: And then you were also co-chair for the 1977 SIAM national meeting, which was held in Denver.

SWARZTRAUBER: Roland Sweet and I co-chaired that meeting with a lot of help from Bobby Schnabel at the University of Colorado. It was a lot of work but I also have some very nice memories. The theme was computational methods in the atmospheric sciences, which includes just about all of computational mathematics. It was a nice experience to attend a meeting where just about every talk is of interest. SIAM meetings include a rather extensive book exhibit that is too expensive to ship back to the publishers so Roland, Bobby and I got a nice perk. We divided the books between us which made a noticeable increase in our individual libraries. It snowed heavily the day before the meeting but attendance was good and the meeting went well, thanks in no small part to Ed Block who was the head of SIAM at that time.

HAIGH: So how would you describe him?

SWARZTRAUBER: He had more energy than any ten other people. He was charismatic, dedicated, and effective in terms of his getting things done. When you're with somebody who is such a hard worker, it becomes a real motivator. You feel guilty if you didn't work hard. He expected a lot from the people that he worked with, but he gave a lot in return. If I were pressed to identify one single person who had done the most to advance applied mathematics in this country, it would Ed Block because of the far reaching significance and contributions of SIAM.

HAIGH: Now, it seems that around about the 1970s and the early 1980s, so this general time period, that SIAM really overtook ACM and its SIGNUM group as the main focal point for people interested in scientific computing and numerical analysis, and more specifically for mathematical software. Any comments on that?

SWARZTRAUBER: I think that's a fair assessment. I think that SIAM, at least at that particular time, was focused more on these issues and topics that were of interest to applied mathematicians. It was only natural that I would be more active in such a society. And I would say also that as a younger man, even though I didn't realize it at the time, I was making connections and being introduced to people who had similar interests and dedication that I would work with for the rest of my career, like Bob Ward. We met as young men at the beginning of our careers. Bob was very much involved in computational mathematics at that time. He moved to management later on. Without that particular contact, there may not have been a CHAMMP Project.

HAIGH: So you think that SIAM's main advantage over ACM was just that it was more focused on this area?

SWARZTRAUBER: Yes, in computational mathematics. That was the draw, and applied mathematics as well. But again I credit Ed Block for making it exciting and making us all feel welcome.

HAIGH: So moving into the 1980s, you were associate editor from ’81 to ’85 of *SIAM Journal on Scientific and Statistical Computing*, and then from 1983 to 1988, you were a member of the SIAM Council.

SWARZTRAUBER: That’s true. The SIAM Council, in contrast to the trustees, was responsible for its technical direction. So I was able to provide that as it related to my field of interest. I certainly enjoyed those days as well

HAIGH: So as a council member, do you recall any controversial issues or any topics on which you might have exerted influence?

SWARZTRAUBER: Well, I don’t remember really any controversial issues, and it’s a little embarrassing! I can’t remember any strong need to redirect them because I always felt that they were basically moving in the right direction under the leadership of its president and Ed. But I suppose I would, as I mentioned earlier, contribute in the sense of helping them identify people or subjects or topics. I encouraged them to have the Denver meeting which focused on atmospheric and related sciences. I also recall a dinner with Ed Block and Francoise Chatelin that resulted in the First International Congress on Industrial and Applied Mathematics in Paris [ICIAM 87]. I don’t know if I was on the Council at that particular time or not. But I never really found myself going in any significantly different direction. There would be, obviously, differences in opinion between the council members in terms of what was the most important topic to cover, but they seemed to be quite fair in terms of the relative attention given to the subtopics in computational mathematics.

HAIGH: You’d also written that you served as the council representative to the board of trustees.

SWARZTRAUBER: That was very short, and I think about the only thing I did at that particular time was report back to the council the things that went on in the board of trustees. But there again, I don’t really remember any huge issues at the time. I might if I went back and looked at some of my notes. I might be able to improve my memory somewhat.

HAIGH: So what kind of things did the board of trustees do?

SWARZTRAUBER: They were more concerned with the business end of SIAM.

HAIGH: All right, so then I’d also mentioned that you were associate editor of *SIAM Journal and Scientific and Statistical Computing*. Any particular recollections of that?

SWARZTRAUBER: Yes, I remember that I was attending a meeting and Gene Golub invited me to his room and said, “We’re thinking of establishing a journal called the SIAM Journal on Scientific and Statistical Computing. I asked why not just the SIAM Journal on Scientific Computing?” He indicated that there was an outstanding statistician who wanted to be involved but also wanted statistics to have that kind of recognition. So that was my first recollection of involvement with that journal. And then, of course, after that, there were the usual editorial duties. It was a terrific journal because it was very applied, and presented actual applications of computational mathematics. I knew a lot of individuals in this area and enjoyed seeing what they were doing.

However I just want to comment on editorial duties. A lot of it can be clerical. People who are involved in those kinds of things are really doing a service to the community, because I don't think it advances their own professional careers all that much other than a nice line on their vita. Oh, by the way, they finally changed the name to the SIAM Journal on Scientific Computing.

HAIGH: On the topic of editorial responsibilities, from 1981 to 1985, you were associate editor of the ACM *Transactions on Mathematical Software*.

SWARZTRAUBER: Well, I would do very, very similar sorts of things. TOMS was a terrific Journal because it gave mathematical software the recognition it deserved. Not only did it provide a mechanism for the distribution of valuable programs and clever algorithms, but it also elevated the status of software developers and provided them with concrete evidence of their acceptance into the research arena. Research institutions measure an employee's worth based on publications, which, thanks to TOMS, now includes individuals who would otherwise be thought of strictly in terms of support roles within the institution. Clearly this motivated the development of quality software. Since I was in that business too, I was very pleased, first of all about the journal and then, about being chosen to help.

HAIGH: And you recall, again, any cases where there were questions about what the policy of the journal should be, or what its future direction should be, that were in any way controversial?

SWARZTRAUBER: I don't recall anything that controversial. There may have been, but I think the main controversy was whether a journal like that could make it. Once that controversy was resolved, then, like I said, mathematical software had a legitimate place in the research community.

HAIGH: And in 1983, you were a guest editor for *Communications at the ACM*. Was that a special issue devoted to something?

SWARZTRAUBER: My vague recollection is that it was associated with algorithms. Lloyd Fosdick was algorithms editor at the time and asked me to help. [Neither Lloyd or I recall the details and there is no record in the CACM digital library].

HAIGH: So from 1976 onward, you were working as a joint professor at the University of Colorado, Boulder. What kinds of things were you teaching there, and what motivated you to do this?

SWARZTRAUBER: Well, again, I would have been invited by Lloyd. I attended department meetings and did some work associated with a few doctorial committees. But to be honest my involvement was minimal.

HAIGH: So you weren't teaching any whole seminars?

SWARZTRAUBER: I gave a few seminars, but I did not teach there. I taught at Stanford and Florida State University.

HAIGH: Okay, so let's move on to your various visiting academic positions then. I have a list of some of them there at the bottom of that page if you want to talk through any of them that were important or interesting or led onto other work.

SWARZTRAUBER: Well, I taught a full year at Florida State University. I taught three courses in advanced computational mathematics. That occupied my main time down there. In doing so, I got some ideas that led to the paper on approximate cyclic reduction [P. N. Swarztrauber, Approximate cyclic reduction, *SIAM J. Sci. Stat. Comput.*, **8**(1987), pp. 199-209].

.HAIGH: Was there anyone in particular you were working with there?

SWARZTRAUBER: I visited the Geophysical Fluid Dynamics Institute; Richard Pfeffer was the head of the institute at that time. I worked with a couple of people over in the mathematics department, but I can't think of their names right now. Christopher Hunter was chair at that time. I also worked on an oceanographic problem with Albert Barcilon and his student Chris Miller. We got an interesting result that showed past a certain critical shore to wind angle, a sandy shoreline becomes unstable. . Later, in an aerial photograph, I saw a wavy shoreline just as the theory predicted.

HAIGH: So you'd mentioned that you taught at Stanford. Was that connected with anyone in particular?

SWARZTRAUBER: Well, of course, Gene was the one who had invited me out, and I also knew Joe Oliger very well because he had worked at NCAR at one time. We were office mates as young men, before he went to Stanford. I taught a semester of computational mathematics and really enjoyed it. In all cases, I made the course from scratch and taught it from notes. I enjoyed the interactions with the students, but I think it would have been a little difficult to teach the same course again and again. The course itself was the challenge, although it was a joy to see the students quickly understand topics that took such a long time to research and develop.

HAIGH: So you've never regretted not pursuing a career in a university?

SWARZTRAUBER: No, I haven't regretted it, but I am grateful for the few times I was invited to go to the university and to teach and interact with the students and faculty.

HAIGH: You've previously mentioned that your eight months at NASA AMES in 1989 was an important experience.

SWARZTRAUBER: That was a very exciting time. The main contact there was David Bailey. He had a similar enthusiasm about computational mathematics. So we would get together, and have wonderful conversations about the work that we were doing. I also had an old friend, Palmer Dyal at NASA AMES who had been our neighbor during our Air Force days. It was at his place one evening that Palmer told me about a problem that was associated with the search for extraterrestrial intelligence [SETI] project. It was a very interesting problem. A few days later I was talking to David Bailey about the problem. We were walking back from lunch, past the huge wind tunnels, to the compute lab, and I said, “David, I think that we could make some very serious progress on that problem.”

Here is the problem. I’ll try to make it as brief as possible. When you’re searching for extraterrestrial intelligence, one has to keep in mind that the stars and planets are always accelerating with respect to one another, so the frequency of any signal is always changing. Even if the source is transmitting at a constant frequency, you can’t just tune to a single frequency because the acceleration will change the frequency, just the Doppler Effect. The Fourier transform is the natural tool for such problems and both David and I had considerable experience and interest in the FFT, so we began to work on the problem. Not only did we need to scan all possible frequencies but we also needed to scan all possible frequency shifts. Therefore we needed to “listen” in a two-dimensional space.

We developed a very efficient way to do this using what we called the fractional FFT: [D. H. Bailey and P. N. Swarztrauber, Fast fractional Fourier transforms and applications, *SIAM Rev.*, **33**(1991), pp. 389-404]. The paper that was directed specifically to the SETI project was: [D. H. Bailey and P. N. Swarztrauber, Efficient detection of a continuous-wave signal with a linear frequency drift, *SIAM J. Sci. Comput.*, **16**(1995) pp. 1233-1239]. We wrote those papers, and we wrote a few others. I think David also wrote some papers on applications of the fractional Fourier transform later on. There is an interesting side story. The same method can also be used to detect lines in an image. We tried it out on an image of Ames Research Center and were able to detect the edges of the main buildings. This would permit the digitizing of cities and towns strictly from images. That was a very exciting time. And what was even more exciting was the huge October earthquake that took down the Bay Bridge. That was a real experience. I had been in earthquakes before but that one was violent. It was like continuously bottoming out while four-wheeling on a mountain road.

[Tape 5, Side A

HAIGH: Now, I know that NASA AMES was a leading center for research on computational fluid dynamics. Did that mean that there were ties in general between AMES and NCAR?

SWARZTRAUBER: Computational mathematics forms ties between most research institutions as evidenced by the work that David and I did. But other than that, there were no other joint projects at that time. But that’s a very good question, why not? Because you’re absolutely right, they are at the top of their field. The reason is that the fluid dynamics associated with supersonic flight is very different from the subsonic fluid dynamics associated weather prediction. But later, there was some cross talk because the methods used to compute supersonic flow were better suited to parallel computers. They do not require as much interprocessor communication. So their methods were examined, but I am not aware of any joint papers being written by experts from NCAR and Ames that compared methods on either sub or supersonic flows. But I would not be surprised if in fact there were some.

HAIGH: Then in 1987, you were a member of the supercomputer project subcommittee of the Computer Sciences and Technology board of NRC.

SWARZTRAUBER: Yes.

HAIGH: Did that involve anything much?

SWARZTRAUBER: No. I recall perhaps a couple of meetings where the topics focused on massively parallel computing but I don't think I was able to contribute very much. I recall trying to stress the importance of programmable communication, which did not generate much interest.

HAIGH: You served as the first director of the NSF Computational Mathematics program. How did that come about, and what were your experiences in this position.

SWARZTRAUBER: It was an interesting experience. I knew Mel Ciment for many years, again via SIAM and our mutual interest in computational mathematics. He went to work for NSF and did quite well. He did a tremendous service to our community by starting a program in computational mathematics within the Math and Physics Directorate. He invited me to head the program as a rotator. I accepted on a short term basis beginning in February 1987 because I had accepted another invitation for the summer. I had in mind that I would be able to talk to folks about their work; however, the workload and time crunch prevented this from happening. A lot of the work was clerical. Each program director had the budget on a spread sheet which was very helpful but also a bit time consuming. There was a sort of shared siege mentality because of the sheer numbers of proposals and limited resources available. The secretaries were not paid well so as soon as they gained a little experience they moved on. In spite of this moral was pretty good. I remember a lot of lunches and dinners in the downtown Washington cafes and buffets. The peer review process worked very well except for the one or two times the Directorate would receive a call from the "hill" and of course the proposal under consideration would then be given "very serious" consideration.

HAIGH: Do you have any other perhaps more general comments about your involvement with the various professional societies and the groups of researchers? Do you think we've covered everything?

SWARZTRAUBER: I think we've covered everything. I could only add, I think, maybe a few exclamation points. SIAM is an outstanding organization, and its contributions are just huge, not only to U.S. applied mathematics but also to the international community. It had about five thousand members at that time, which was large for such a society. It was an important component of my career and I always encourage the young people to join.

HAIGH: I think that in the summer of last year you retired from NCAR.

SWARZTRAUBER: That's right.

HAIGH: What led to that? Was that something that you'd been thinking about for a number of years?

SWARZTRAUBER: My early thoughts on retirement put the age at 50. That seemed old at the time. But at age 50 I was having a wonderful time. So, sigh, "Well okay, I'll put it off until 55." And then I put it off until 60, and then I kind of put it off indefinitely. I think at 65 I began to consider it seriously. I retired at age 67, July, 2004.

I think the environment at NCAR has certainly changed, but it has also changed elsewhere. Money is just much more difficult to find, and people have to spend more time getting it. I was very lucky, money for science was plentiful during most of my career and it was only toward the end that I felt some pressure to submit proposals. NCAR was fully funded by the NSF for many years. The CHAMMP Project provided NCAR with a few dollars but with minimal paperwork on my part. Of course research directions are limited to those that are funded, which made a fundamental change in an organization where the senior scientists previously had this responsibility.

HAIGH: Since you've been retired, have you continued to work on these topics? Have you developed any new interests?

SWARZTRAUBER: Well, I actually, to be honest, no new projects, but I spend a little time on computational fluids and on computer design. I have not had a problem staying busy. We have a winter home that required a lot of work, and I have absolutely no trouble filling the hours of the day. I'm probably busier now, certainly physically. I don't sit in front of a terminal as much. So far retirement has not been a problem, but of course it's only been a couple years. Thanks, Tom, it's always nice to talk about one's self. I know there are some gaps here and there, and I'll try and fill those in when you present the results for me to edit.

HAIGH: Well, I'll finish up then with the same two open-ended questions that I've asked everybody. The first of those would be that as you look back over your career, what do you think your single biggest regret would be, either in terms of some decision or action that you took, or perhaps just in terms of something in the world that went one way and you wish it had gone another way?

SWARZTRAUBER: Regrets, you know, I have very few. I think I mentioned earlier that I'm leaving an organization where I think it's more difficult for the young people to accomplish their research goals, or at least they don't have as much time to accomplish their research goals as the organization that I came into. Probably I wouldn't have had any control over that, because as I mentioned earlier, I think that problem is common through the research industry and beyond: it's very difficult to get money. But if I would have been aware of it earlier, perhaps I could have tried somehow or other to correct that. I probably was too focused on my own world and my own research. Working at NCAR was a joy and so too was the environment. It is located in the foothills of the Rockies. I would park at the bottom and hike to the laboratory most days. I calculated that in forty three years I hiked that hill about four thousand times

Oh, I regret not having continued my guitar lessons! I really don't have any serious regrets about a career that took me much further than I expected to go.

HAIGH: And the more positive reverse of that question, over the same span of your career, what do you think the single accomplishment that you would be most proud of would be?

SWARZTRAUBER: Well, I think I can name a few. The early paper, "The Solution of Separable Elliptic Equations", was extremely important for my career because that caught the attention of Gene Golub, and his mentoring was very important in those early years.

In retrospect, I think the software packages were important. Although they seemed just offshoots of my research, they certainly got a lot of use. So that was very important even though it perhaps didn't seem that way at the time. Most researchers would probably agree that there is some difference between the papers they like and the ones that are liked by others.

I'm very proud of the machine that I patented. It is the culmination of so many things learned, both about computers and algorithms. It has the potential to be the most far reaching result. I'm proud but also regret not being more effective in guiding its development.

In terms of projects, I was very pleased about the CHAMMP Project because this was four or five labs cooperating and working in harmony toward a specific goal.

HAIGH: So that concludes all the questions that I'd prepared. If you have any other comments to make or things that you think that we should cover, then now would be the opportunity.

SWARZTRAUBER: Well I have many stories associated with travel but there is one that immediately comes to mind. In 1983 I was visiting the Siberian Academy in Novosibirsk. Each morning I would meet my hosts in the hotel lobby and we would take a very nice walk through the woods to the laboratory. One morning my hosts began talking to each other in Russian, which was quite unusual. Then one turned to me and said "there has been an incident between our countries." I felt the hair rise on the back of my neck. The Korean airliner had been shot down. This precipitated a series of nerve wracking events, which included the closing of multiple airports to travel from the Soviet Union. But then, in the middle of the night, the phone rang and it was my supervisor Walter Macintyre – how sweet the sound. He had arranged for alternate air transportation in the event that I needed it.

Tom, you've put together an outstanding list of questions. You obviously spent some time researching my vita and my website and publications because nothing else comes to mind.

HAIGH: Okay. Well, thank you very much for taking part in the interview, then.

SWARZTRAUBER: It's been a pleasure.

APPENDIX

Here we include a continuation of the answer to the following question posed by Thomas Haigh:

HAIGH: Yes. I'm just working up to that, so I'll ask you about that in a second. In 1993 NCAR became the first, and indeed only, site to receive a Cray 3, the machine developed by Seymour Cray at his new firm Cray Computer Corporation which was based in Colorado. How did NCAR's relationship with Cray develop during the 1990s, and what was NCAR's experience with the Cray 3?

The following is from Jim Hack, Tom Engel, and Bill Buzbee:

From Jim Hack:

The Cray-3 was indeed very much alive when we housed it in the SCD machine room. And yes, we got useful cycles out of it and the performance was extremely competitive with anything else out there. It was the single greatest reason why Buzbee opted to go for a competitive procurement back in 1995 (the ill-fated NEC procurement). His thinking was that Seymour was in a position to give CRI a run for its money. But shortly after the RFP hit the streets the Colorado Springs Cray Computer Corporation went bankrupt. We moved the machine off the floor because (a) it was only there for testing purposes, even though we got useful work out of it, and (b) the parent company went belly up; there was no one to maintain it. It was a good machine and DOD and DOE missed a great opportunity to keep them alive.

From Tom Engel:

Yes, Serial 5 "graywolf" was alive and well and used by NCAR scientists (Hack, Boville, etc. - seems to me you did some benchmarking runs on it - or was that Dick Sato?) from early 1994 up until the day Cray Computer Corp. filed Chapter 11 (24 Mar 1995) ... we actually installed it on the floor at NCAR as a "test" system (no cost to NCAR except to pay the power/cooling bill) in the fall of 1993 (Seymour felt the true test was if it could survive NCAR ... he'd done a similar thing with the Cray-1 and LANL). Serial 5 had stability problems until ~Feb 1994. I had many sleepless nights during the spring of 1994 working with CCC engineers to identify problems with the system, working with Seymour and our designers (we found some design problems at NCAR, too). NCAR also used Serial 7 of the Cray-3 which we were running in the checkout bay at CCC in Colorado Springs.

As for how it stacked up - it was the most powerful system of its time. It took ~$200M to develop and build - we were fab-ing our own GaAs circuits in Colorado Springs - and invented many new technologies (several patents) ... by the time we got the kinks worked out of the Cray-3, Seymour and CCC's board decided the Cray-4 would be where we'd put our effort (much better price/performance) and we'd estimated we could deliver the Cray-4 at ~2x the FLOPs/$ than NEC's SX-4. We demo'd functioning memory and floating point unit of the Cray-4, running with a 980 picosec clock, to GFDL just before CCC filed Chapter 11

Attached are two emails (one to Bill Buzbee; one to Gordon Bell) from a few years ago with more details about the systems & CCC.

From: Tom Engel [engel@ucar.edu]
Sent: Friday, September 28, 2001 09:18
To: BLBuzbee@aol.com; gmichael@home.com
Cc: eugene@nas.nasa.gov
Subject: Re: Cray 3
Bill, et.al.:

NCAR used two Cray-3 systems: sn5 (located at NCAR) and sn7 (located in CCC's checkout bay in Colorado Springs). NCAR also used CCC's Cray-2, sn2025, which was also located at CCC. The
sn5 system was a 2-cpu system, in a 4-cpu chassis; sn7 was a 4-cpu system in an 8-cpu chassis.

The Cray-3 at NCAR was shut down late in the morning of March 24, 1995; the day CCC announced Chapter 11. It stayed, powered down, on NCAR's floor for (I believe) a couple months while CCC evaluated whether it could resume work or must end its existence. When sn5 was removed from NCAR, the chassis, PDU, etc. were shipped back to Colorado Springs and put on the "auction block" (recall, CCC was attempting to sell its facility, fab, material assets and IP; but no entity was interested in buying the whole package - thus the material assets were liquidated). My recollection is that components of sn5 (and other systems we had at CCC) were cannibalized and various parts were sold & given/donated as "trinkets" or memorabilia. For instance, a couple years ago I received an email, with scanned images, from a fellow in Britain who was trying to figure out what the CRAY-3 module he bought was used for - it turned out to be a foreground processor module, which I suspected at the time came from sn5. CCC's building and fab was purchased by M/A-COM. Whatever assets that were not sold at auction was turned over for scrap/recycling.

Tom
```
_____ Tom Engel ________________________________________ engel@ucar.edu
Scientific Computing Division, National Center for Atmospheric Research
P.O. Box 3000                 Phone 303-497-1270        Fax 303-497-1848
Boulder, CO  80307-3000       Pager 800-306-1988 (8003061988@skytel.com)
```

From: Tom Engel [engel@ucar.edu]
Sent: Thursday, March 27, 2003 12:55
To: BLBuzbee@aol.com; gbell@microsoft.com; spicer@computerhistory.org
Subject: RE: Cray-3 "Speed" rating
Gordon, et.al.:

The Cray-3 that was at NCAR (S/N 5, "graywolf") was a 4-processor chassis; Bill is right though, that it was just a 2-processor system: it had both "bricks" of modules, but the CPU modules in one brick were not functional. It was installed at NCAR in the fall of 1993, as Seymour said, "if it survived at NCAR, it'd survive anywhere" … but was not really operational until February 1994 and it, along with an 8-processor system (S/N 7), to which NCAR also had access via the internet, on CCC's checkout bay floor ran operationally until CCC's Chapter 11 filing on 24 March 1995.

The Cray-3 had a 500 MHz clock (2 ns) – 2 FLOP/clock - 1 GFLOPs peak/processor. It was designed to have up to 16 CPUs and 512 Mwords (8-byte word) per octagonal cabinet; memory bandwidth of 16 Gw/sec.

The Cray-4 had a 1 GHz clock (1 ns) – 2 FLOP/clock – 2 GFLOPs peak/processor. It was designed to have up to 32 CPUs per node and up to 4 nodes in a "system"; a node could have up to 2 Gwords (8 byte word) of memory. Thus the largest system would have a peak of 256 GFLOPs. Memory and the floating-point portion of a Cray-4 CPU was demonstrated to GFDL just a couple weeks before we filed Chapter 11 … the clock speed used at that demo was 980 picoseconds and the results calculated during the demo (a kernel from one of their models) were correct.

Seymour and CCC were continuing to use GaAs for the logic circuits in the Cray-4 at the time of the Chapter 11 filing (24 Mar '95).

Seymour's "SRC" (late '95/'96) is when he decided to use commodity processors (Intel) and design a large, fast, flat memory subsystem … he acknowledged that he could see the late '90s being the point where microprocessors would be fast enough that they could be used more cost-effectively in building a "supercomputer" than designing your own custom processor, but that you need a very fast memory to keep them from data-starving … unfortunately, we lost Seymour before he could bring the SRC machine to fruition.

Tom Engel (engel@ucar.edu) 303-497-1270

Also a couple other pointers:

http://www.cisl.ucar.edu/computers/gallery/cray/graywolf.jsp (though some info is not completely accurate)

http://www.digibarn.com/collections/systems/crays/cray3/index.html

http://www.top500.org/system/1284

From Bill Buzbee:

Indeed, I'm one the SIAM Oral History website honorees --

http://history.siam.org/oralhistories.htm

The associated pfd doc -- pp 2. 45-46, 54, 57, 59 --

http://history.siam.org/pdfs2/Buzbee_returned_SIAM_copy.pdf

discusses the SX-4 procurement. Also, mention is made of MUDPACK (PA 48)
and the decoding of Buneman's Fast Poisson Algorithm (pp 24-25).

Re Cray-3, et al -- the attachment was presented at a private event honoring Seymour at SC05 last fall.

Bill Buzbee's "Draft Comments for Seymour"

Thank you for the opportunity to participate in tonight's tribute to Seymour – a truly remarkable person and an a remarkably innovative engineer.

I first met Seymour in the mid-70s, and my favorite memory of him is his description of his career. I attended a CRI customer event in Minneapolis in the early 80s and Seymour was the after-dinner speaker. On that occasion, he gave a short synopsis of his career.   He said he began his career in the 50s as an electrical engineer – and as such, he was able to pay the bills, support the family, etc. Then in the 60s, with the design of the CDC 6600 – the first major system to employ 3-dimensional packaging – he said he had become a mechanical engineer and, as such, he was able to open a saving account – life got better! Then in 70s, with the design of the Cray-1 which employed liquid cooling, he said he had become a plumber and, as is often the case with plumbers, he said he became "financially comfortable!"   Comforts that were well earned and well deserved!

Speaking of Seymour's innovative designs, I had the good fortune to experience them both as a user and as a manager.

As a user, I first encountered one of Seymour's machines in the early 1960s when I was a grad student at the University of Texas in Austin. The machine was the CDC 1604. From there I went to Los Alamos where I had my 2nd encounter with a Seymour machine – the CDC 6600; then the CDC 7600; then the Cray-1. One of the things that I appreciated most about Seymour's designs was the ease with which one could comprehend and apply them. As I observe the architectural complexity that many of our colleagues are faced with today, I am EXTREMELY thankful that I worked in the "Era of Seymour!"

I got into management in the early 70s and by the mid-70s, we at Los Alamos had acquired an acute appreciation of short vectors versus long vectors. Simply put, we knew that Los Alamos models required a machine that could achieve high performance on short vectors. So when the Cray-1 design was first unveiled, Jack Worlton and others at Los Alamos were on it like stink on a skunk! Consequently, serial 1 of the Cray-1 was installed at Los Alamos in 1976. Shortly thereafter and as a result of our experience with it, Seymour decided to incorporate SECDED into the memory system. So he cannibalized serial 2 and used parts of serial 2 to build serial 3 which was the first machine with SECDED. Ironically, NCAR ended up with serial 3 and Los Alamos with serial 4. That came about because NCAR was also an early adopter of short vector machines. NCAR financed the purchase of their 1st Cray-1 and, thus, CRI received payment-in-full for it. That payment put Cray Research in the black – financially speaking. Since Los Alamos already had serial 1 in operation, serial 3 went to NCAR, then serial 4 went to Los Alamos to replace serial 1.

As many of you may recall, NCAR was the only site to host a Cray-3 and that occurred during my tenure there as Director of the Scientific Computing Division. Both Seymour and I relocated to Colorado in the late 80s – he established Cray Computer Corp in CO Springs and I joined NCAR in Boulder. Livermore had an option to get serial 1, but withdrew its offer, circa 1990. Consequently, Gary Jenson -- who headed our operations section at NCAR and was very active in the Cray Users Group – suggested that we meet with Seymour and ask how NCAR might help CCC. So, top mgmt at NCAR went to CO Springs and posed that question to Seymour. He thought for a moment and then replied, "How about testing and showcasing the Cray-3?" Further, to facilitate transport of models to the Cray-3, he offered NCAR use of idle time on the Cray-2 at CCC and there was quite a bit of idle time at nights and on weekends. So not only did NCAR transport models to that architecture, a lot of science was done as well. It was a heck of a deal for both organizations! The Cray-3 was installed at NCAR in Oct 93 and decommissioned in Mar 95.

Here is a photo taken in the NCAR machine room while the Cray -3 was in residence there. As you can see, not only was the Cray-3 the first computer to operate at a 1 nanosecond clock, it was also the world's first desktop supercomputer! The entire machine is under the Plexiglas canopy – it would literally fit on a desktop!