An interview with
AUGUSTIN DUBRULLE

Conducted by Thomas Haigh
On
30 and 31 July, 2004
Arroyo Grande, California

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

Augustin Dubrulle discusses his career as a producer of software libraries for scientific computing and numerical analysis. Dubrulle grew up in France, studying mathematics at the University of Lille and serving in Algeria before accepting a job in 1962 with the IBM Scientific Service Bureau in Paris. He describes the work of the service bureau and the state of scientific computing and numerical analysis in France during this period. In 1967 he moved to the IBM Scientific Center in Paris, but soon left for White Plains, NY, on secondment to work on a conversion of IBM's Scientific Software Package (SSP) for PL/1. Dubrulle was part of a small group, which in 1968 moved to Houston. The group's leader, Ed Battiste, left IBM to found leading scientific software firm IMSL. Dubrulle describes the origins of SSP in IBM Boblingen, its reception by users, and various updated and improved versions, including PL/Math and SL/Math. In 1970, Dubrulle moved to an IBM development office in Menlo Park, California, produced a mathematical software package called Math-Basic for the System 3 minicomputer, and gained permanent residency in the USA. He then worked for several years in the IBM Advanced Systems Development Division in Palo Alto on projects concerning scientific computations, including a stint in the Future Systems organization and a development for the measurement and modelling of computer performance. He then transferred to the Palo Alto Scientific Center to participate in a research project of computational geophysics for oil exploration. Upon completion of this assignment, he switched his interests to vector computations in support of IBM's 3090VF vector supercomputer, for which he produced in 1989 a version of EISPACK. Dubrulle, who earned a Ph.D. in the mid 1980s under the direction of Cleve Moler, also discusses his relationship with prominent members of the mathematical software community, including Cleve Moler, Jack Dongarra, and Jim Wilkinson. After leaving IBM in 1992, he worked on a mathematical software library for Hewlett Packard's Wide Word processor, which eventually became Intel's IA64 architecture. He transferred to Convex, then in the process of being acquired by HP, where he continued to work on software libraries before retiring in 1998 when HP lost interest in this area.

HAIGH: SIAM oral history project. Interview with Augustin Dubrulle, this interview is being conducted by Thomas Haigh in Arroyo Grande, California, on the afternoon of the thirtieth of July 2004. This is the beginning of the first session. Tape 1, Side 1.

Thank you very much for agreeing to take part in the interview. To begin with I wonder if you could describe your early life and upbringing in general terms.

DUBRULLE: The most important part of my early life was during World War Two. I was ten at the end of the war, it was rather miserable times but on a shoe string I got a very good education, primary education, which is amazing considering that we cannot get that anymore nowadays with enormous amounts of money, and after that then I went through the usual baccalaureate track, which at the time was two exams over the last two years of the lycée and corresponded pretty much with what's called here an associate degree. After that I went to a prep school for the so-called "Grandes Ecoles."

HAIGH: So how old would you have been at that point?

DUBRULLE: When I went to prep school, nineteen. I spent a year there and then I spent four years at the University of Lille in northern France, where I prepared a double master's degree in physics and in math.

HAIGH: So when did you first become interested in science and mathematics?

DUBRULLE: Very early. My parents and my teachers forced me during my secondary education to learn Latin---that was the big thing in France. My two foreign languages were Latin, which is dead language, and German. To me Latin was really like a millstone around my neck, but to maintain a B average I worked harder in science and math, which I liked. That's how I got into it. I liked mathematics particularly because it required, in my view, less hard work than physics. And it had a logical beauty to it, and it was controllable. So that's how I got interested in mathematics.

HAIGH: And was it a subject you were particularly good at in your earlier education?

DUBRULLE: Yes, yes. I was not a genius but I was good at it. At the university, as I followed a physics component for my physics master's degree, I worked much harder than I did in mathematics, but I thought it would be useful just to understand how things worked.

HAIGH: And how did the French educational system work at that point, did you make the decision to specialize in science at an early age or was it made, as in the American system, after you arrived in university?

DUBRULLE: You could specialize much earlier in France. Even though I had to take Latin, which I took for six years, there was a track which was some kind of a strange mixture of humanities and science, it was called the C Section---nothing medical there. And it was, usually, it was the refuge of the oddballs because other students would go either the purely scientific/technical or humanities tracks. From a status viewpoint humanities ranked higher than anything else in the French system. You had to take Latin, Greek, things like this. At the time the French system was extremely selective.

The baccalaureate was very selective. I don't remember the percentage, but now sixty percent of the people of age, I think, in France now have their baccalaureate. Not in those

days, it was something more like six percent, and so it was very selective. It was based on stuffing into you a lot of knowledge, you didn't have much time to let your mind wander. For many people it killed creativity. A lot of people who were very good students at the time, or considered very good students, didn't really achieve very much in life because they were good students for the rest of their lives and had to be told what to do.

Later on at the university it was little bit different, it was, what a friend of mine called, an abortive system, that is, you get your baccalaureate, you get to higher education---that opens the door to higher education---and then you're pretty much on your own. And at the time everything was decided upon a yearly exam. Each unit of the master's degree was obtained by passing an exam at the end of the year over everything you had learned during that year, which was a little bit hard. A lot of people were held back, doing two years, three years, to get one certificate. But it was not totally uniform. In Lille where I was it was very demanding on science and math, and not in the humanities, but in science and math it was pretty demanding.

HAIGH: So, how did you make the decision to go to Lille?

DUBRULLE: My father died when I was seventeen, and at the time we were living at about seventy miles east of Paris. But our families, our relatives, were all in northern France, and that's why we moved to northern France, where Lille was the bigger university town.

HAIGH: Was it considered a prestigious university?

DUBRULLE: It was considered pretty prestigious. I believe that in science and math it was about second to Paris. Actually a lot of the professors lived in Paris, they'd take the train, go to Lille, give their lecture, go back to Paris. Couldn't see them. They did nothing like counseling.

HAIGH: So, during your undergraduate education did you come into close contact with any of the faculty or was it a less personal kind of experience?

DUBRULLE: Yes and no. There was one who was helpful. I was on a scholarship in my second year at the university, working towards the master's degree, I had a choice of working on either my big year of physics or my big year of math. They were both pretty hard, and since I was on the scholarship I couldn't miss a year. I couldn't fail, or I would lose my scholarship for a year. So I asked a few professors to give me advice on that and only one give me the right advice, which was to go for math first. But other than that, it was an anonymous system.

HAIGH: And how much exposure did you have during your first degree to numerical analysis and computation?

DUBRULLE: None whatsoever.

HAIGH: Was it a subject that just wasn't taught at that time?

DUBRULLE: It didn't exist. I think there might have been at the time one university in France that was dealing with computing and that was Grenoble, I think. I don't know of any other.

HAIGH: And actually just to get the dates here, what year did you enter university and what year did you graduate?

DUBRULLE: I entered the university in 1954 [*that was rather prep school*], and I graduated in 1959.

HAIGH: And when you graduated did you have your master's degree?

DUBRULLE: I had both.

HAIGH: So you graduated with both?

DUBRULLE: Yes. There was no in between at the time. You've got to realize that between that time and now there have been maybe ten reforms of the system in France.

HAIGH: So, at that point you would enter the university and if you were on course to continue with your studies you wouldn't get any degree until a master's degree?

DUBRULLE: That's right.

HAIGH: And what proportion of students obtained a master's degree at that point?

DUBRULLE: I don't know. It's difficult to figure out because, for example, I knew one fellow student who had been drafted because he had failed for two years on the big year in math. That was the end of his deferment, and he came in the third year for the written exam in his uniform. So there were guys who were taking, instead of taking four years for a master's degree, some were taking five, six, eight years. So it's kind of difficult to have a good idea, it cannot be based on one class.

HAIGH: But was obtaining a master's degree something that people routinely did or was it something that you did only if you knew that you wanted to go on and have an academic or technical career in this specialized area?

DUBRULLE: The latter.

HAIGH: So, at the point that you made the decision to do this double master's degree in physics and mathematics how did you visualize your future?

DUBRULLE: Well, at that point I was thinking of teaching in a lycée, but I was open to about anything else. Except that I didn't know very much. For a time I considered continuing for a doctorate to teach in a university, there were special doctorates for teaching in the universities. Then I approached some of my professors and they didn't have any idea about research, they didn't have, they were not the kind of people who were spending their summer vacations doing research. So in general the subjects for the dissertations were their own, from when they got their own dissertations, which was not too exciting. They were not getting you started for something really new. The whole scene was sedate.

HAIGH: And was there any kind of thesis or research project involved in your degrees?

DUBRULLE: No, it was pure studies. What you could do, and what I did, was to take units: you could spend one year on optional certificates, and I took applied mathematics, which included very new concepts, like distribution theory, which I felt was really exciting. It was based mostly on Laurent Schwartz's newly-defined theory, which started being published in 1948, so it was fairly recent.

HAIGH: But there was nothing about numerical methods and –

DUBRULLE: Oh, no, no, no, that didn't exist at all.

HAIGH: I know at least in some places in America and Britain these methods were studied prior to the introduction of electronic computers, although, obviously that made an enormous boost to the field.

DUBRULLE: Well, England was light years ahead, okay. There was Turing, there was Wilkinson, the Pilot Ace and things like this. I mean there was nothing like that going on in France.

HAIGH: Did you ever use a desk calculator?

DUBRULLE: No. [Laughter]

HAIGH: Okay, so through your entire education there was nothing at all do with computers or computation?

DUBRULLE: No, I used a slide rule in physics.

HAIGH: So, what happened when you graduated from the university?

DUBRULLE: I got drafted and I spent twenty-eight months in the Army. I was discharged as a second lieutenant.

HAIGH: And was this during the period of the Algerian War?

DUBRULLE: Yes, yes. The atmosphere there was a little bit like Iraq, with a big difference: it was a colonial war, which made it even worse. It was a low morale war because everybody knew how it would end, and it was an anachronism. It was ridiculous. When everybody was getting rid their empires, and France was clinging to Indo China and Algeria, so it was not a good war.

HAIGH: Was pretty much everybody drafted?

DUBRULLE: Yes. At the time France was maintaining about five hundred thousand soldiers in active duty, draftees, year after year.

The only good thing I got out of that experience is that I became proficient in the use of firearms, and officer school taught me quite a bit about leadership. I should say that I got classes on leadership and in other things that were very good. Unfortunately, at the time most of the French Army didn't follow those principles.

HAIGH: And did that experience change anything about your career plans or how you thought of yourself?

DUBRULLE: I felt that my life should be as exciting as possible, that was the idea, because I had good friends who got killed there, didn't come back. Some got badly injured and crippled there. There were not too many though. In eight years of that war there were only seventeen thousand dead, which is not enormous. It was much less than during the war in Indo China and the Vietnam war. Still, when the guy with whom you had dinner last night is dead, was killed in an ambush, that makes you think about life.

HAIGH: And what did you do after you were discharged?

DUBRULLE: Well, before my discharge I fired letters to IBM, Burroughs, and Bull, and also to an outfit that was doing some research on the diseases affecting coal miners. That had nothing to do with computers, that was more like statistics. And I got very encouraging letters, they were saying that they were all interested in hiring me. France at the time was coming out of the post World War Two depression, only then, and so there was a lot of hiring.

HAIGH: IBM, Burroughs and Bull were all in the computer business. Did you apply to those firms because you wanted to work with computers?

DUBRULLE: I wanted to know what it was all about.

HAIGH: Yeah. So at the point when you were applying did you know anything at all about what a computer was and what it did?

DUBRULLE: No. But that was no problem. [Laughter] Actually there were a lot of people in my situation.

HAIGH: Yeah, so what was the attraction of computing for you?

DUBRULLE: It was new. It was new and very likely there was some interesting work to do there. Every time there is something new there is some interesting work to do with it.

HAIGH: And were these companies looking to hire mathematicians, in programming jobs or general computing jobs, or were they just looking for mathematicians for research type positions?

DUBRULLE: Everything.

HAIGH: So they had an idea that a mathematician would, was someone who would do well in a variety of different computer jobs?

DUBRULLE: Yes, yes. But at the time---that was very funny---the outfit in which I was hired had people like psychologists, anybody… but mostly engineers. So, anyway, I got interviews at Burroughs and IBM, and finally I skipped Burroughs and I decided that IBM was the place. I got six months of training at IBM to learn what a computer was, half of that training was about the old machines that worked with connection boards.

HAIGH: Punch card machines?

DUBRULLE: No, no, no, connection boards.

HAIGH: The most sophisticated punched card machines, such as the 400 series tabulators, had a “plug board” front panel that you would program by wiring.

DUBRULLE: That’s it. Of course it worked with punch cards, but it was connection boards. Tabulators, sorters, and wiring, that was about half the curriculum.

HAIGH: What year was it by this point?

DUBRULLE: It was 1962, after I had spent twenty-eight months with the French Army. The second part of the curriculum was the 1401, so we were learning how to program the 1401, and the usual case studies had to do with inventory, how to make deductions for social security in the payroll, and all that kind of stuff, which was not too exciting.

HAIGH: So, you were being trained in application programming on the 1401?

DUBRULLE: A little bit, yes. But after those six months of school there were many possibilities. You could go and be a sales rep, which was not my bag really, I didn't like that too much. You could be the technical guy in a branch office, that is, in a sales office, which was much more system oriented. And there was another possibility, which was a center of research that was near Nice, in southern France. So I asked to go and visit there and I didn't find it too exciting. Finally there was another possibility in Paris, which was the service bureau, the scientific service bureau. That really appealed to me because the thing we were doing every day was solving problems. Typically engineers would come with big files of equations and the task was to find numerical solutions to complex equations. So I went there, and efficiency was the name of the game, that was how we were making money. There was a lot of pressure, and a lot of pressure to learn very fast, because we didn't know anything really, and we had to learn on the job.

HAIGH: Can you describe what the service bureau was like?

DUBRULLE: I know only the scientific side of the service bureau.

HAIGH: So it had a business side but that was separate?

DUBRULLE: It had a business side which was at a different location in Paris. Where I was was Place Vendome, which is very near upscale places. IBM, I think, is not there anymore, but that's where the IBM France headquarters was. The scientific Service Bureau was about one hundred and twenty people, altogether. At the time it was composed of several departments. The very biggest one was analysis and programming. Another one, very small, was methods, which was made of people who were advising customers and programmers and analysts about methods to solve numerical problems. There was a customer service department, which wasn't very big, which was holding the hand of customers who were working on site instead of trusting the programming to IBM. They would be given cubicles where they would work, do their own programming. They went on the machine, and they were paying for machine time. The big thing then was machine time. At the time the machine was a 7090, which was an interesting machine with a small memory, 32,768 words of 36 bits each with double precision accuracy. An hour of 7090 at the time was six hundred bucks, it was very expensive, it was more expensive than human time. Then there were administrative departments for personnel, accounting, etc. etc. There was this very small group that was doing commercial applications, I don't know why they were there, they should have been with the other outfit but they were there, and then there was operations, and that's about it. I'm forgetting something there.

HAIGH : So that was a hundred and twenty people just within the scientific service bureau?

DUBRULLE: Yes.

HAIGH: So would that have been the largest operation of its kind in France at the time?

DUBRULLE: Yes. Now it got reorganized, after a while. So I got there and I started programming in Fortran II and also in assembler. You had to know assembler, you had to know the system, because to develop your program the only thing available was the dump. The dump was a printing of all the memory contents and you had to pour over that and find ways to get rid of the bugs. So you would get the dump and get stuck with that

for a few days, and then find something, hopefully, and do another run and so on and so forth.

My first programming was something for Schlumberger. It was to reconcile different measurements of cores being drilled in oil prospecting: measurements based on physical and chemical properties, and another set of measurements based on electrical properties. It was pretty boring. I had little jobs in between, and after that I got another job which was to develop for a civil engineer a program that was really on matrix computations, matrices of order 500, maybe more, which had to do with structural analysis. He didn't have his design very well set, and I had a very difficult time with that, but I learned a few things.

After that I decided that I had had it with programming for customers, I asked to be transferred to the Department of Methods, and I thought that numerical methods looked interesting. So I moved to the Department of Methods and I was working with two other numerical analysts, with a few more years than I had in the business, like two. They were not really up to snuff, they were still learning. And for a year I did some development about everything numerical analysis: differential equations, matrix computations, exponential approximation, which taught me really a few things about numerical instability, polynomials, all that kind of stuff.

HAIGH: And where were you learning these things from?

DUBRULLE: Books, experiments, papers. Of course I had to learn English.

HAIGH: There was no literature in French?

DUBRULLE: No. During that time, I discovered the book by Jim Wilkinson, "Rounding Errors in Algebraic Processes", which I read and which I found very interesting. It made me really understand what numerical instability was. Because at the time floating-point arithmetic was not understood by almost anybody, there was always something mysterious about it, something unpredictable. And I read Jim's book and I thought it was great, it was so simple. So I was happy, and then I started learning about the eigenvalue problem because we had a lot of eigenvalue problems to solve, and what was used at the time was a combination of power method and inverse iteration. There were various devices to accelerate the iteration and so on, but the software looked like big contraptions that were very inefficient. Most of our customers had at the time large matrices, over three hundred, and we were looking if we could compute thirty or forty of them with not very good precision.

HAIGH: Now at this point when someone came along with a job that they needed programming, would there have been an extensive library of subroutines within the center to deal with common problem types?

DUBRULLE: That's what started the mission of the Department of Methods, and as I was learning things I was plugging some holes that were in our library, differential equations and so on and so forth. We had a subscription to the Communications of the ACM which had the rubric of numerical algorithms, we were getting the computer journal which was big at the time, we were getting *Numerische Mathematik*, which was also very good, and Mathematics of Computation which was a little bit behind the times. But in some of these journals there were algorithms that were not really sound or proven.

An author would write a paper on, say, Runge-Kutta methods for solving differential equations, and he would write that this method and that method didn't work on his problem but this new one did. But the new method often was unstable and the field was kind of chaotic. So, to go back to the eigenvalue problem, our eigenvalue solvers didn't work very well---actually no eigenvalue solver in France worked well. A lot of customers that we had were going to the CDC service bureau in Paris, on the false premise that with forty-eight bits on the CDC 3600 you had more precision. Unfortunately there was not the double precision accuracy of the 7094, and there was rounding before normalization. But people didn't understand those things, so they went to CDC, and then they came back to IBM.

I came across a technical report from Stanford out of the dissertation by one Beresford Parlett---he's now emeritus at Berkeley, I know him very well, he's a good friend---which was using a brand new method called Laguerre's method, which he had devised for his dissertation. I was just amazed, you would feed the thing with a matrix of order 200 and get the 200 eigenvalues to good precision, which was unheard of. So I worked from that and I wrote programs for the symmetric case.

I came across another article in the computer journal by J.G.F. Francis about the QR algorithm. Wilkinson was in the acknowledgement, and that got my attention. I studied that very much in depth and I think I might have developed the first French Fortran program on the QR algorithm. It was amazing, it was even faster than Parlett's which was pretty fast at the time. So I thought it was great and I got more and more interested in the eigenvalue problem.

[Start of Tape 2, Side B]

HAIGH: So, you had mentioned that CDC was a competitor. What other main service bureaus or computation centers were you aware of in France at that point?

DUBRULLE: I think at that point Control Data was the main competitor for scientific computing. There were not too many service bureaus. Actually the French service bureau was a little bit of a novelty, because it had been ruled in an antitrust suit in the U.S. that IBM shouldn't have a service bureau, that it violated the antitrust laws. But apparently IBM France, although being a subsidiary of IBM, was not subjected to the law because French law didn't care. But I think that, yes, Control Data was the competitor.

HAIGH: And were there any main academic or commercial computer installations within France at that point that you were corresponding with or you were aware of interesting work being done at?

DUBRULLE: Yes, yes. There was mostly the one which was at a close suburb of Paris, called Montrouge, and it was the research center of the Electricé de France. They were doing good work. I don't think they were doing very original work but they were doing good work. Keep in mind that at the time, we were not doing very much original work, we were all learning. There was also the Centre de l'Energie Atomique, the atomic energy center, but there they were a little more secretive, they were working on De Gaulle's nuclear force, so they didn't talk too much. Actually Electricé de France was also working on that because that was a great way to hide defense spending, and there were other outfits doing that kind of thing. But the center in Montrouge was very good.

And I met interesting people who would help me later on, they included Jim Wilkinson. So, anyway, when things were good and nice, the whole Scientific Service Bureau was reorganized. Some bean counters looked at the figures and said, "Okay you are in the Department of Methods with all those people that are kind of overhead. We cannot assign revenue to those people who give advice to customers, build things for the library, who knows. We've got to do something about it." So they decided to make managers out of the people in this department, and then to slice the analysis and programming department into units, each unit going into each guy in the methods department depending on specialty. At the time the guys with more seniority than I in the methods department had moved to the IBM Scientific Center in Paris.

HAIGH: And what was being done at that center?

DUBRULLE: Hard to tell, I don't know if I can. It was run by somebody who wanted it to be not a place where you do a lot of useful work, but some kind of a show place, and the show was based on business cards---we have so and so who is a graduate of Ecole Polytechnique, we have so and so from Ecole Normale Supérieure, and so on and so forth. The manager of the Scientific Center had made a name by doing a statistical analysis of De Gaulle's speeches, and he had written something about the words that were coming up with the higher frequencies and were supposed to reveal something about the guy. In the Scientific Service Bureau reorganization, I was given a group of about twelve people, it was a group for numerical analysis, statistics, and linear programming. There was a group for civil engineering, there was a group for something having to do with opinion surveys. Those were the main activities really, they were the money makers. The only problem that I saw at the time is that I had to keep my job as a numerical analyst and be a manager at the same time, which meant that I did most of the managing and dealing with clients during the day, and did the numerical analysis at home at night.

HAIGH: So the idea was that you would simultaneously be managing a group of programmers in an area and also supporting them with superior methods and subroutines?

DUBRULLE: That's right.

HAIGH: Why did they come up with that arrangement? Did they have the idea that people with more mathematical sophistication would inherently be operating at a higher level somehow?

DUBRULLE: Nope, no, it had to do with the idea that there were those pointy heads on whom you could not pin a revenue. Now if you made them, if you made them heads of programming groups then you could.

HAIGH: There just doesn't seem much necessary association between managerial ability and expertise in numerical methods.

DUBRULLE: Well that's right. Absolutely.

HAIGH: So, how did you personally feel about the different roles? You had been doing hands-on programming of scientific computations, then you had moved more to examining methods, and now you were working in a managerial capacity. How did you feel about those different jobs initially?

DUBRULLE: Well, I was still dealing with studying and developing numerical methods. The management role… well, I didn't dislike either. The only thing is that a day is only twenty four hours, so I did that for a year, and then I was getting a little frazzled. So I went to the director of the institute and I said that I wanted to leave, I wanted to be transferred to the Scientific Center, and he said no you've got to stay here because we are going to get a 360, a new machine, with the new arithmetic and I want you to stay until the installation in a steady state. So I said okay, so I stayed there until April 1967.

HAIGH: So you were there for five years?

DUBRULLE: '62 to '67… not quite, four and a half. When I was there I managed also to go to seminars at the Sorbonne. There was a very interesting guy at the time who was bridging numerical analysis and very theoretical mathematics, his name was Jacques-Louis Lions. He was a great guy, he was a genius, and he seemed to be the only academic in France who was very much alive and had plenty of energy. He made a name for himself in the U.S. and everywhere in the world, and he was a very accessible man. But his numerical analysis was really more like applied mathematics. It was very general, like the use of variational methods in spaces of distributions to solve partial differential equations, and everybody was saying that it was useless. It was not useless, because if you could understand it, when finite elements came into the picture, it made it so simple. So it gave me an idea that, hey, after all there is a world out there, and I started going to conferences and things like this.

In 1966 there was a very memorable conference in eastern France, in Besançon. The head of the computer science department in the university of Besançon apparently had good political connections and got oodles of money to invite an enormous number of stars, of numerical analysis stars. Many of them were coming back from a congress of mathematicians in Moscow. So, I went there and I met---now I can drop names---Jim Wilkinson, Alston Householder, John Todd, Olga Tausky, Faddeeva, the Russian, Lanczos, whom some people call "Lanczosh," Fritz Bauer from Germany, maybe a few others, but it was incredible, it was star packed. On the train to Besançon was a friend from the research center of Electricité de France, his name was David Feingold. He had worked in the U.S. for more than a year with Golub and Varga, and Wilkinson. We were on the train, and he said, "Hey come with me and I'll introduce you to Jim." So we go and, well, I'm introduced to God. And God is a very nice guy, he seems to have a good sense of humor, and with Feingold acting as an interpreter, we start having a conversation. Jim is very much interested in my field experiments with Parlett's method and the QR algorithm for the eigenvalue problem. He's asking a lot of questions. He's really interested because he wants to compare notes with his experiments at the National Physical Laboratory. So I give him figures, and we talked all the way to Besançon, it was a lot of fun.

I attend the conference in Besançon, and I said to Jim that I could send him more data, and he said "by all means do that." Back in Paris, I sent him the data, and he answered very nicely. And then I go back to the grind, and I ask for a transfer to get into the Scientific Center. In the Scientific Center one of the missions was to help customers solve their problems, but nobody wanted to do that. They were so afraid of failure.

HAIGH: So, in 1967 you did eventually switch to the Scientific Center?

DUBRULLE: Yeah, that's right.

HAIGH: And was the 360 running smoothly by that time?

DUBRULLE: Yes. In between I had a lot of problems because everybody, or almost everybody, was using unstable methods. The fact that there was not automatic accumulation in higher precision, as in the 7090 hardware, created a lot of problems. Some people were getting different results all the time. So there was a long line everyday at my door for a while. People were really bitching and I had to explain to them what was going on, which was not very easy. But the 360 was running smoothly by then. One of the ideas was, well, you've got to use better numerical methods, you've got to pay attention to things, and I know you don't have the time to hit the books and the journals, but that's a fact of life. Maybe you're getting results that differ from what you were getting before, but you get them faster, so why don't you use higher precision? That doesn't guarantee that stability is going to come back but maybe you're going to get better results. So after a while things just quieted down and I moved out in April of '67. At the Scientific Center I kept helping IBM personnel solve their problems, for interesting problems.

HAIGH: Can you give some examples of what some of the most important customers would have been for the Scientific Service Bureau as a whole? You've mentioned a couple that you worked on personally. But what would the biggest repeat customers for the whole operation have been?

DUBRULLE: Civil engineering. And that was going to the civil engineering department in Turin also. Linear programming was a big money maker. Oil companies too for problems of mixtures between gasoline and various fuels. That was a big money maker. But those guys didn't really rely very much on the Methods department. They were doing their own stuff. They would come and take the machine for a couple of hours and run things. They were paying for the machine time. Nuclear energy, and also defense applications had a lot of eigenvalue problems. I never knew exactly what was going on, I don't know if it had to do with neutron transport.

HAIGH: So were there some problems that you worked personally on that you knew the mathematics but didn't know what the project really was?

DUBRULLE: Yes. One big project I worked on was an earth-dam model. I didn't know the mathematics because it was so complicated, a set of partial differential equations, over a domain that was made, that was totally inhomogeneous, made of many, many different porosities, and so on and so forth. It was almost impossible to find out if there was a solution, and if that solution was stable. The only thing I could do was to rely on the engineers who had taken the problem to us, who said, "oh we built models and they're holding the water."

HAIGH: And IBM was trusted by the French government to work on sensitive topics?

DUBRULLE: A few things. For example, for a while there was a project that had to do with building the hull of the first French nuclear submarine. There were rumors about it: that finally the hull was too small, the reactor was too big for the hull, and they had to redo everything. I don't if it's true, but that was the rumor. There were some projects in defense but usually we didn't know it was really for defense because it was coming either

from Electricité de France or Gaz de France or some other outfit that was working under other colors.

HAIGH: And was IBM a prestigious firm to work for at this point?

DUBRULLE: Yes, definitely. At the time there was an aura about computers. Even if you were an operator who was just sticking the reels into the tape units, wow, you were really somebody. Yes, IBM had a lot of prestige. Actually the headquarters where the Scientific Service Bureau was, was on the Place Vendome, had big windows, and from the street people could see inside. At the bottom of the windows there was always a display like a reproduction of the Pascal calculating machine and things like this. And another window was about Descartes, and another window where people would look and see the operator go and put their reels in it and then push the buttons. They could see the entire machine room. The machine room originally had a 7090, then it was 7094 that was replaced by a 360, a 1601 that was driving a Calcomp plotter, and a 1401 or 1410 to feed the 7090 or 360, and a lot of tape units, and of course console and card hopper and all that kind of stuff. So people would look at that and be very impressed, and the funny part is that there were U.S. tourists who would go there and say "Ah, IBM, I work for IBM". And then there someone would give them a tour.

HAIGH: Did people see IBM France as being an American thing or had it taken on the local culture and blended in with the French surroundings?

DUBRULLE: I think French mostly considered IBM as an American company, and everybody of course was joking that at IBM you had to wear a uniform, which was pretty true. At the time people were dressing up generally way better than they are now. And there was also in France the mystique of the cadre. The cadre was at the time a yuppy, a young up and coming higher-education person. So they were people who were buying a certain car, which was at the time the Peugeot 404, and so on and so forth. These people were dressing pretty much like IBMers, and they were buying Dunhill cigarette lighters, and they had the right fountain pens, they had the right anything, and they were eating at the right places and seeing the right movies and buying the weekly l'Express. So in a way there was not that much difference, but, nevertheless, people were kind of joking about IBM, and they were considering IBM more like an American company.

HAIGH: So you had been talking about your move to the Scientific Center.

DUBRULLE: Yeah. In the Scientific Center I worked on a few problems for customers. Usually they were people who had small machines like the IBM 1130, and they had to solve big problems on those machines, and they didn't know how to do it. So I worked on that kind of thing, using secondary storage and so on, and that's when I started thinking of devising methods---or tweaking methods---so that they could work on hierarchical storage. Again, I didn't have it figured out at the time, but I started thinking about it. So I worked on that and one day, a guy I new from the U.S., who was working in White Plains for the mothe company called IBM World Trade . . . IBM World Trade included practically all IBMs outside the U.S., and it was headed by Dick Watson, A.K. Waston, who was the brother of T.J. Watson, Jr. This man was working there and was trying to get some cooperation from World Trade countries with developments going on in the U.S.. I had answered his call a year earlier when I was at the Scientific Service Bureau concerning a development called SSP, and that must have been 1965.

So that man, his name was Rainer Kogon---we are very good friends--- had studied at Rice University. He had been raised until age eighteen or nineteen in Europe, his mother was German, his father may have been Swiss French. He spent the war years in Switzerland, and then he came to the U.S. Actually he was part of the committee who later designed PL/1, which was supposed to be Fortran VI, and became PL/1. Anyway, Rainer one day came to me and said that there was going to be a project in White Plains, New York, and "would you like a two-year assignment there, to work on it? It has to do with the conversion of SSP to PL/1." I said that sure I'd like to go to the U.S. I did actually. In joining IBM I kept an eye on the opportunity to go to the U.S. and to see what the country was like.

HAIGH: So up to this point you had never visited America?

DUBRULLE: No. I didn't speak English. So I said okay, and, I think it was in July, we started the process. I didn't have a passport, and it was impossible for me to get something like a green card in that short a time, as I had to be in White Plains right away. So I get some kind of an eighteen-month visa, and that allowed me to go to White Plains to work for IBM as a trainee. So I went to White Plains---at the time Christiane [*Dubrulle's spouse*] was pregnant, and I went ahead in August, on the fourteenth or fifteenth of August---and got busy getting to know people, starting to learn English, finding a flat, buying a car, and so on and so forth. When I started working on that SSP PL/1, I didn't know zip about PL/1, of course. I didn't find it that great.

HAIGH: So, by this point had the compilers PL/1 it been publicly released?

DUBRULLE: I think it had been released, in 1965 or, 1966, I believe. That's when I was first exposed to it in Paris, and I looked at it at the time and found it a bit ridiculous. There were … I had big problems with customers at the Service Bureau, who took the opportunity that you could cram as many PL/1 statements as you wanted in a punch card so that they would pay less, because fees were based on how many punched cards a program used. So they were cramming as many statements, PL/1 statements, as they could on a card because there was a semicolon at the end of each statement. And it was a big nightmare, you couldn't read the darn thing.

I thought that, well, PL/1 didn't seem to be different from Fortran except that there were things that we didn't understand about it, which were strange. Those were the defaults variable definitions. PL/1 had a lot of defaults, and you had to be very specific about your definitions otherwise it would take the default and do certain things that were nonsensical. That's all I knew about PL/1, I'd never practiced it. So, anyway, I got to White Plains and in the project there I was supposed to be in charge of the numerical analysis part. Ed Battiste was in charge of statistics, and we hired two girls, one from Smith, and the other one from New Rochelle college, to work with us. They didn't know anything about programming. There was a guy who was working part time for IBM who was a former drag racer, Robert Tapia, who had done some programming for IBM, and was also associated with the project. We did the a translation of SSP into, from Fortran to PL/1. We had very little time to do it, something like eight months, seven months, so it was not very good, the quality wasn't very good. I was not too happy about it, but anyway, meanwhile I was discovering America, which kept me really busy.

HAIGH: So, what was your initial reaction to America?

DUBRULLE: Surprise. Before I went to White Plains, New York, and I had asked people what's White Plains, and they said well it's almost like a suburb of New York City. When I get there it's not like a suburb. First of all between New York City and White Plains there is the Hutchinson Parkway, it's beautiful, it's so green, it's like country. It doesn't look like suburban New York, suburban in a sense of suburb in Paris, which is solidly built up. Then I get in White Plains, no tall buildings, the only tall building is maybe two or three stories, barely, it's the Sears building, and there are squirrels in the streets, and when you get out of White Plains you see raccoons. Very surprising. The people were incredibly nice. At that time Christiane was pregnant, she had had a little bit of English in school but no English speaking skill at all, so she had a terrible time buying pots and pans and things like this, but the people were so nice. She was going to that hardware store and she would try to say "I want a pan," and the guy, it was a really old man, would come with one and say, "something like this?" She would say "no, no," so he would run back and get something else, and finally she would find the right item. People were so helpful. And that's not the kind of thing you see very much in France. In France when you walk into a store, in a way you are disturbing the people, you are disturbing the retail clerks. So, anyway, we liked it very much. While the product was in testing we took a big trip along the coast, driving to Key West, and stopping along the way and meeting people, and so on, which was great. So we liked America right away.

HAIGH: And did you notice any big differences between IBM America and the people you were working with in the U.S. compared to those in France?

DUBRULLE: Yes. In France it sounded more hierarchical. Here in the U.S., if you have an idea, speak it and somebody will listen. People might not think it's a good idea but at least you have a chance. And if people think it's a good idea they'll say so. In France it's different. Another thing, in France you don't ask questions, most people don't ask questions. You have a meeting, a technical meeting, you're presenting something in France, people in the audience don't understand, they behave as if they understood, because they don't want to ask questions, because that would make them look ignorant. In the U.S. people say, "I don't understand, please explain," or "I don't know, please explain." So that's another big difference. Overall it was way more open. Also at that time, and that has changed, at that time although I was fairly junior guy in some ways---I was certainly a junior in the U.S., because I could barely speak English although I was working hard at it---I knew what my boss was doing every day, I knew what his boss was doing every day, and I had a very good idea what my boss's boss's boss was doing. That standard has disappeared nowadays, but anyway, that was surely different from what was happening in France. So that's the kind of thing to which you adjust very well. It was a very comfortable environment, so I found the adjustment from this viewpoint easier than I thought.

HAIGH: And the group you were working with in White Plains you said that they were part of the IBM World Trade Corporation?

DUBRULLE: No, they were part of the IBM Data Processing Division, that was a group called Industry Development/Scientific Applications. There were several departments in there that mirrored pretty well the latest organization that I have known in the Service Bureau. There was optimization and numerical analysis, statistics, simulation, and so on.

HAIGH: So, by that point was scientific support within IBM was being handled as a specific industry group alongside other industries? Would there also have been say an industry group for the oil business, or something like that, at the same kind of organizational level?

DUBRULLE: A little bit, but we were a little bit not as high on the totem pole, okay. We were cross industry applications, which means it was difficult to pin a figure related to revenue on us, if you see what I mean.

HAIGH: Yes.

DUBRULLE: Okay. And for oil, or whatever, that's easier. In a way, that was the problem in the trade I was in, and it would get worse and worse with years, or maybe I got more conscious of it with years. At that time I was extremely busy, being acculturated to America, learning the language, learning also about a new side of communications. You don't communicate with Americans the way you communicate with the French, fortunately. So, after the release of SSP PL/1, we were told that our group would move to Houston, and that Ed Battiste would be the manager of the group, I was still on temporary assignment and since I was still theoretically an employee of IBM France I could say no to the move to Houston.

[Start of Tape 2, Side A]

DUBRULLE: So, in 1968, around September, we were told that we would have to move to Houston, Texas. I could stay in White Plains, but then I would be assigned to a project that didn't have very much to do with mathematical software. So I said I wanted to move to Houston, because Texas looks like a different place and I want to know about it. So I moved to Houston and one of the first notable persons we met in Houston, when we were apartment hunting, was Richard Nixon. The campaign was in full swing and we were in the lobby in the hotel to meet Ed Battiste and all of a sudden the doors opened and Richard Nixon walks in, at the time there was not that much security. So we moved to Houston and got together with Ed, and then we got in our new digs on Fannin Street, and that was the building where the IBM Scientific Center of Houston was. We were one story above, I think.

HAIGH: Now was it just this relatively small team that had been working on SSP that moved to Texas or was there a large migration of scientific staff within IBM?

DUBRULLE: No, that was only the team that had worked on SSP. So, we get down there and then Ed and I sat down and we said we were not too happy with SSP PL/1, although Ed had bought a personalized plate for his car which had "SSP PL/1" on it---that car was a lemon anyway, it was a big Chrysler New Yorker that was always in the garage. So we sat down and we were not too happy about SSP PL/1 and we said well, you know, we should do something better, we should write a better SSP because SSP had flaws.

HAIGH: What were the specific issues with the PL/1 version?

DUBRULLE: Too short a time, not enough attention to careful implementation of algorithms. It was still SSP, and we felt that now things were changing and we could do better.

HAIGH: And was the PL/1 version of SSP every widely used?

DUBRULLE: I don't think so. It was widely distributed because sales reps were handing them out like candy…

HAIGH: So that was part of the campaign to persuade people to switch to PL/1, something you could point to in promoting the language ?

DUBRULLE: That's right, that's right. The big idea, the big strategy was this people will use PL/1 and say, wow we can use single language for scientific and business applications. But we didn't buy that, Ed and I. Because we knew that there were millions of lines of live Fortran sitting at JPL and other places, and JPL was not about to switch to PL/1. To translate these millions of lines into PL/1 would be impossible. So we considered doing a better SSP. How we were going to do a better SSP was an open question. Mind you at the time SSP's design was about four years old, it had one design defect which had to do with the storage of matrices.

HAIGH: So, maybe this would be the good time discuss its origins .

DUBRULLE: Yeah, the design was four years old. I'm not sure, but I think it started out with IBM Germany, with a team in Boeblingen headed by Dr. Albrecht Blaser. The principal numerical analyst in there was Ulrich Schauer. I don't think Schauer had a formal background in numerical analysis because he was older than I. Actually I don't know, I don't know what was his formal background.

So the idea started there and it was a very good idea, in my opinion, because it made it some kind of homogenous package for its documentation, documentation was uniform, presentation was uniform, subroutines had a certain regularity in their calling sequences which was new at the time. Most libraries were collections of programs coming from here and there and elsewhere, which had nothing in common. There was the SHARE library which was …, to which contributions were made by IBM users and it was a repository more than anything else. It was really inhomogeneous, and there were big programs, programs that were not that good.

So SSP had that uniformity about it and uniformity of documentation, and documentation was given a good emphasis because documentation is a very important feature of mathematical software. You get a program, if it doesn't describe precisely what it does it's of no good use. So, anyway, the design went on, there were calls to some IBM outfits to contribute. For example, Jim Cooley (of the pair Cooley and Tuckey who devised the Fast Fourier Transform) contributed Fast Fourier Transforms to the package, and that was excellent. There were other things that were not that excellent. There was a naive blunder about the storage of matrices. But overall, for the times, it was a good effort.

Well, what happened is that Ulrich Schauer coming back from a party at this manager's, wrapped his car around a tree and spent months in the hospital in very bad shape, skull fractures and things like this, and the project was delayed because people didn't know where he had put his notes. The thing resumed, and I assume it was in 1965 that Rainer Kogon, whom I mentioned earlier, came to me in the Service Bureau in Paris and told me about SSP, and asked me if I wanted to get an advance release copy to test in a customer environment, which I did. And he asked me also to review the design, so I reviewed the design and I sent a letter to the developers in Boeblingen. I told them that, the package looked pretty good except for the blunder in storing matrices and a few other items, and

there is no program for the general eigenvalue problem, but that I could provide one program to computes eigenvalues. And they answered, thank you, and as to the storage mode of matrices, maybe it's a blunder but that will remain." I had a few trips to Boeblingen to get acquainted, and that's how I got to know Albrecht Blaser and Ulrich Schauer.

HAIGH: I have a few more questions on the early years. One of them would have been how widely the package was used when it was released to the public?

DUBRULLE: I don't know exactly how widely it was used when it was released, but I know that it was widely used, and it was used by other manufacturers because it was pretty much portable.

HAIGH: So the code itself was all written in Fortran?

DUBRULLE: Yeah.

HAIGH: And it was a library so to use these routines people would link that Fortran in when they compiled their own programs?

DUBRULLE: Yes, yes.

HAIGH: And do you know would users look at the code inside the routine or would they just treat it as a black box?

DUBRULLE: Black box, yeah. And there was quite a bit of use for the statistical part.

HAIGH: And are you aware of any cases in which installations made their own changes to the library?

DUBRULLE: Yes, yes. I don't have them in mind now, but I knew some. And I know, for example, that there was a famous statistical program which had been put in there by Ed Battiste, which was unfortunate, for Ed was very much criticized for it, which was the random number generator called RANDU. It was made to exploit the architecture of the 360 to avoid a floating-point division by using integer overflow---automated integer overflow---to do the division, and that was clever. But the random number generator had the unfortunate property of generating triplets that were very close to a plane. Some people found that out and RANDU became a big joke. Kahan massacred it, of course, as usual, and Knuth mentioned it in his book on---I think it might be---Seminumerical Algorithms. So it became a big joke and poor Ed was feeling terrible. So, some people replaced the innards of RANDU by something else, and still called it RANDU, but RANDU is still around. The other day I made a search on the internet and found that it's still used somewhere. It's incredible. Actually there are still some SSP programs that are alive, that are still in use.

HAIGH: So do you know if any pieces of the source code found their way into other packages?

DUBRULLE: Yes. I know that's true. Actually I found that not too long ago on the internet, but I don't remember which package. I know, for example . . . One time I took a class at Stanford---we could attend Stanford classes through television at IBM---and I took a class, I don't remember what it was, unfortunately, but the software that was handed out contained an SSP routine, and I thought it was kind of odd.

HAIGH: Was it credited?

DUBRULLE: I don't remember. It was in the early 80s.

HAIGH: Anyway, it's interesting that you mentioned that the code was used by other manufacturers. Now did you, or anyone else at IBM at that point, view the code as something that was the intellectual property of IBM?

DUBRULLE: In a way yes, but we didn't care. Actually IBM didn't really care. IBM at the time was focusing on the hardware, and SSP was only a nicety. My view is that IBM believed that it was providing so many more services than the other manufacturers that it had the customer married to IBM for life. So and it didn't matter if people were taking SSP as SSP was not, to IBM, a big deal. What was a big deal to IBM was COBOL and commercial applications. Tthat's where IBM was making big money. So IBM didn't get too excited about that.

HAIGH: Do you know if the code was even copyrighted?

DUBRULLE: I ought to know because I looked at the codes many times.

HAIGH: I believe at the point the law was that unless you included a copyright message in the statement it was automatically in the public domain.

DUBRULLE: Yes, I think so, yes. I don't think there was a copyright statement at all. That's too bad, you know, you don't think of history. I had all the manuals of SSP, SSP PL/1, and SL Math, and when we moved I had to get rid of them. I had too much stuff, and I said well after all, you know, IBM must have all that stuff.

HAIGH: So, a couple more questions related to SSP. You mentioned you had a statistical component and to keep matrices. Were there any other main areas that SSP covered?

DUBRULLE: Yes.

HAIGH: Aside from that what else was include in the scope of the package?

DUBRULLE: There was symmetric eigenvalue problem (the Jacobi method), differential equations (Runge-Kutta), linear equations, of course, including linear least squares solutions, which can be considered almost a separate topic. Polynomials, elementary operations on polynomials, polynomial economization---that was the Lanczos method, taking a polynomial expanding in terms of Chebyshev polynomials then truncating what looks small to a certain criteria. So there was polynomials, matrices, ordinary differential equations, Fast Fourier Transforms, and various things. There were some programs for Fourier series, that were different from the Fourier Transform, of course. Polynomial approximation, roots of polynomials also. That's about all the subjects that were at the time fairly popular. There were a few people trying to compute the coefficients of the characteristic polynomial of a matrix to compute the eigenvalues---there were still people who were a little bit behind. But in a way all the popular topics of the times, except partial differential equations, of course, which is in itself really a separate topic.

HAIGH: And do you know if there had been more than one release of the Fortran SSP?

DUBRULLE: Yes, yes. There wwere three releases all together and they came very quickly one after the other. Usually they consisted of the addition of a few routines because, as usual, there were always dates, deadlines, that were dictated by reasons

having nothing to do with the product itself but having to do with marketing. IBM always had that idea that software was there to "drag along" some hardware, and IBM was focused on hardware. So, if at some point it was important, if it looked important to drag along some hardware, well the software had to come out at that time. Which I think was a big mistake because it was imposing very artificial deadlines that sometimes impacted the quality of the products---"you have to start now and end then." And then the whole development process was a bit painful because you had to do a business case, you had to show that you would follow the IBM standards, you had to follow test procedures that were set in concrete, and if you didn't follow the test procedures you had to submit a deviation from internal standards, expressing why you were deviating. And then you had to discuss that with the business people and marketing. Making the business case with the people in the business office was some kind of a dance. You would make a list of justifications for a product and the marketing guy would say that's not enough and that you should consider other elements, and you'd consider that too. You'd go back and the guy would say that there's not enough emphasis on this and that, and after a while you thought "why doesn't he do the bloody thing himself, you know, instead of making me do it the way he wants it." That was part of the game, but a lot of time was spent back and forth doing those things, and when you would start the development you were really squeezed for time.

HAIGH: So the later releases were just basically to add routines that hadn't been finished in time to make the initial one?

DUBRULLE: Yes, that's right, essentially, yeah.

HAIGH: And you've mentioned Kahan and some other numerical analysts had issues with some of the things the group did. Was there any kind of systematic feedback that came in from ordinary users of the packages? Not necessarily numerical specialists but ordinary engineers and scientists who were using these routines?

DUBRULLE: Yes. But usually it had to do with misuse most of the time, misuse or unstable problems. When it had to do with unstable problems and not with the software, it was sometimes very difficult to explain that to customers, and also it would take time. You have to take the problem and look at it and show the instability.

HAIGH: And was that something that you were dealing with personally while you were in the Service Bureau?

DUBRULLE: I was doing that in the Service Bureau all the time, all the time. Not only there were questions of problem instability, but there were also algorithms developed by the customers themselves based on some algebra book that were completely unstable.

HAIGH: So were there any insights that you got from this exposure to ordinary users that later in your career shaped the way that you approached software development?

DUBRULLE: Yes, definitely, yes.

HAIGH: And what were they?

DUBRULLE: They were that you've got to be nice to customers, they may be up to something, but you shouldn't be their patsy. You shouldn't be their patsy, but you have to tell them that very nicely, so that they are not offended. It gives you the idea that a piece

of software is not just a piece of software, it's a service. It's also smart, in a way. If you don't know about some problem and then you have a black box to solve it, that black box should be as reliable as possible. So of course at the time of SSP things were not that reliable. Wilkinson's book, "The Algebraic Eigenvalue Problem," came out in 1965, and 1968 was only three years later. Usually when there is any breakthrough in numerical analysis, for it to trickle down to common use---not even common use, to let's say commercial exploitation of the ideas---takes more than ten years, I think. When I did my first experiments, when I developed my first program for the QR algorithm, that was 1964, and I think that the QR algorithm started being in use, in wide use, around 1970 maybe. The article by Francis(?) was published in the Computer Journal in 1961, '62. So, anyway, when I say 1970 I should say 1971 when Wilkinson published "The Handbook for Automatic Computation: Volume 2, Linear Algebra." So that was ten years, essentially. Software is a service, and a piece of smarts, and it shouldn't be taken very lightly.

HAIGH: And was there anything in the approach to documentation of these packages used trying to make sure that users were applying them in circumstances that were mathematically appropriate?

DUBRULLE: That we couldn't do, I think. That we couldn't do, because problems come in so many forms. In a way one can say that in one sentence, "use that only on a well posed problem." Well you say that to the user, and he said "what the hell, how do I know my problem is well posed?" So there was always implicit in there the limitation that you were in finite computation, finite representation, and that entailed some problems.

HAIGH: Another question, not so much about SSP itself. I wonder if you could describe Ed Battiste and what your relationship was with him, and how it developed?

DUBURULLE: We were good friends.

HAIGH: So what was your first impression of him when you initially met him?

DUBRULLE: He was very congenial, he had a good sense of humor, he liked to laugh and to joke. He played a lot of jokes on me when I arrived because I didn't understand English. He would stick his head in my door and ask "would you like to have lunch at Mother's---the IBM cafeteria---or at the pizza place on Mamaroneck Avenue," and he would say it extremely fast so that I would not understand. And I would pick up "lunch" and "Mother's" and I would think that he's asking if I want to have lunch at Mother's. So I would say "yes" and he would laugh his head off and go. I thought that was an easy joke but, you know, that came with the territory.

We got along pretty well, I can say we were personal friends. But Ed, like everybody else, had his quirks. One of his quirks was to ask my opinion on something and I would give him my opinion. If he liked it, he'd "say oh great!", but if he didn't like it he would start arguing. I would say "you asked for my opinion I'm giving you my opinion, that's it." No, he would argue, and at times he would get pretty hot under the collar. I think his father was Italian and he always said that he was hot-blooded because of his Italian descent, but otherwise technically we got along. Actually in general we got along pretty well.

HAIGH: Do you know what he'd been doing at IBM before this SSP PL/1 project?

DUBURLLE: I don't know, that's a good question. I think he might have been working in a branch office, maybe something like this. Because when I met him he was thirty-nine, I was thirty-two, so he must have been somewhere before. But he was with IBM because he had come out of the IBM university resident program. That was a very nice IBM benefit. If you worked for IBM you could apply for the program and if your evaluation showed that you had the potential to go to a university on a Ph.D. program, IBM the first year would pay seventy percent of your salary, and the following years sixty percent of your salary. That was an incredible benefit. So, Ed took advantage of it and got his Ph.D. in statistics at University of North Carolina at Chapel Hill. So he must have been with IBM a substantial amount of time before that to get on that program.

HAIGH: And had he been responsible for the statistics in Fortran version of SSP?

DUBRULLE: I don't think so. He might have contributed to it.

HAIGH: But you don't think he was the main person?

DUBRULLE: No, no, he was not the main person. I think that for the Fortran version the main people were in Germany.

HAIGH: So that brings us back to Texas. What happened to the group after it moved there?

DUBRULLE: Well most of the group moved there, and what happened is that Ed and I were making plans for doing a better SSP. And then we submitted the plans and then the word came down from White Plains that it had to be in PL/1, again. Ed got furious. We had a big powwow with some managers in White Plains, who came down to Houston, supposedly to listen to our case, which was the case for Fortran. We spent two days discussing that and at the end of the two days they said "Well, no matter what, the decision has been made for business reasons to stay with PL/1." So Ed was really furious, and I wasn't too happy either. Ed got on the horn and called upper management in Westchester County to give them a piece of his mind. Then he hung up the phone went home furious, and then didn't sleep all night, thinking about everything he had said on the phone [Laughter]. The next morning he came back and started feverishly writing memos to try to kind of soften the blow. So we said, "okay we are going to go with PL/1," and we did.

HAIGH: Do you know if this pressure to write exclusively in PL/1 was affecting all areas of IBM application programming?

DUBRULLE: I don't think so, I don't think so. Linear programming was a different animal, because first of all it was not in Fortran. There were a lot of things that were not in Fortran, like the simulation programs. There was GPSS, there was CSMP, CSMP was about continuous simulation with differential equations, GPSS was discrete simulation. Those were not written in Fortran, except maybe for some subroutines in CMSP for the integration of ODEs---ordinary differential equations. So those were not really affected by that. IBM had in its sights a few customers who were willing to do the experiments of going PL/1 for everything, science and administration. One of them was Dow Corning.

We did the design for PL/1 and we said well we are going to try to do the best we can and we'll do this, we'll do that, and we'll work in connection with the Scientific Center,

namely. There were two numerical analysts there, Bill Timlake, who was an alumnus of the ETA in Zurich. He had done his Ph.D. under Henrici, and the other one was Stewart Lynn, who was the manager of the Center. So, at some point we said we've got to hire a consultant to check our design, and Bill Timlake recommended Cleve Moler. We were now in 1969, around May or June, May I guess.

HAIGH: And by this point what you're working on is this improved SSP in PL/1

DUBRULLE: Yes. So Cleve came. We were fairly impressed with him, there were not too many people at the time who seemed to understand software as well. He was also, he still is, a fine numerical analyst, and he had good connections. He had those summer conferences at the University of Michigan, where big stars were going year after year, including Jim Wilkinson.

Cleve came and spent some time with us and he was pretty serious and did a good job, made a few recommendations. We talked a little bit and he asked if I was doing some research and I said "Well I don't have the time to do very much research but I worked on that algorithm for the eigenvalue problem. I wanted to publish it in Numerische Mathematik and with the Algol code, as usual in the Handbook Series. And when I asked IBM for a clearance to publish, some lawyer said 'No you cannot publish that because there is a program in it.' I said to the lawyer 'Well it's only for communication it's not really code,' and he said 'No I don't buy it.' So, I thought 'Okay, I'm going to change the paper and maybe I'll submit elsewhere.' Then I go to the library and I find in the latest issue of Numerische Mathematik, oh maybe not the latest, but a recent issue, an article by Jim Wilkinson with Martin, on the same algorithm, the implicit QL algorithm for symmetric tridiagonal matrices, and I think that I don't have a chance.

[Start of Tape 2, Side B]

DUBRULLE: And then coming out of that funk a week later, I compared to the two algorithms and mine is thirty percent faster. So I show that to Cleve, and Cleve said "Hey that's really good, er, you've got to publish it, you've got to show it to Jim, you've got to come to the summer conferences in Michigan," and I said "Okay." So I wrote to Jim to show the algorithm to him, but there was a typo, unfortunately, in the notation, and Jim answered "I see what you're doing," er, "it's much better and I'm looking forward to seeing you in Michigan." So, I go to Michigan, I see Jim. Householder was teaching there too, and Henrici was teaching there too. I don't remember the other people, and I had a great time, and then I realize that the summer conferences at the University of Michigan were not really the kind of summer school that Cleve had mentioned. Some people thought that it was some kind of summer school where you go to learn something. No it was a thing where most of the guys who were there were high specialists in their trade and they were meeting there to talk to other specialists and to the master's . Well, it was terrific. So, I show the algorithm to Jim. He finds it good and thinks he'll put it in the handbook. He suggests that meanwhile I should publish it in Numerische Mathematik. I tell him that IBM doesn't want to let me, the IBM lawyer doesn't want me to put the Algol code. Jim said it didn't matter and that I should just publish the formulas. So I had a one page paper in which I wrote "the following is an algorithm that is more efficient than the one blah, blah, blah, and I thanked Wilkinson." I think it's still the shortest article published in Numerische of Mathematik in all its history.

HAIGH: So interesting that IBM had no objections to competitors using SSP but was worried about algorithm published in a scientific journal.

DUBURLLE: Yes, because in between, there had been a judgment about software unbundling. I forgot to mention that, thank you. So, IBM couldn't give away any code anymore, but the code that was out there was public domain.

HAIGH: So the unbundling decision filtered rapidly through the whole of IBM in terms of a new attitude to software as being more than just a free adjunct to hardware?

DUBRULLE: That's right, exactly. So, I spent quite some time in Michigan with Jim. I used to give him rides to his hotel, so we got to talk quite a bit and, as I could speak English, or some approximation thereof, that made communications much easier. Jim was a very interesting guy. Of course he was very smart, but he was very generous. He was not clubbish, but he was also very competitive. When he started talking about some technical subject, he would take no prisoners, which was very good, but he was very nice about it. So, we hit it off, and when I went back to Houston, and I was pretty happy with that. IBM gave me the clearance, I submitted the paper and it was accepted.

HAIGH: Was that your first published paper?

DUBRULLE: Yes. I had not had the time before. It took me a few years to get to that. I guess maybe I was too dedicated to my job and not enough to myself. I had to deal with a lot of things because also when I moved to Texas I had to relearn English. I don't know if you've traveled to Texas but… only the accentuated syllables are pronounced. I found Texas very interesting. We did the new SSP PL/1, that is, PL_MATH, which took us until the summer or the fall of 1969, and we passed the IBM test. Then we sat down again and said okay now we're going to do it in Fortran, and we're going to do it even better.

HAIGH: Right. Now in what specific ways was the super SSP superior to the original SSP?

DUBRULLE: The one we were thinking of in the future?

HAIGH: Yes. As I understand the story, you had implemented the straight conversion of the original SSP into PL/1, and you had wanted to do a much improved version of SSP, and you had received permission to do that, but only in PL/1. What were the improvements?

DUBRULLE: We put more functions into it. We had the complete set of routines for the eigenvalue problem that didn't exist in SSP. SSP had very embryonic capabilities in that area. Ed also implemented improvements in the statistical part, which I couldn't talk about, because I don't know. Everywhere we could find a better algorithm, a more recent algorithm, we put it in.

HAIGH: So the main advantage was just that there were more functions that represented a broader and better selection of numerical methods?

DUBRULLE: And a better quality. We got rid of certain functions that were notably unstable. In ordinary differential equations we got rid of at least one, it might have been

Milne's method, which is a multistep method that had been known to be unstable for some time.

HAIGH: And I think you had said before that when this package was actually produced it was called PL-Math, is that correct?

DUBRULLE: That's right. It was called PL-Math. Before that, several names were tried but they were all trademarks. One of them was NPL, and somebody mentioned that there is the National Physical Laboratory, so finally we ended up with PL-Math. So then it went all out, Ed and I said "there is one thing that's missing. It's proper technical support," and instead of publishing a new version of the package and going through the whole set of procedures of IBM it would be much better to have a maintenance program by which periodically we would replace a few subroutines and send a notice to the users and to replace some subroutines. And now, how would we get the expertise to make choices on algorithms and things like this? Well, we could have a board of advisors, like academics, who would point at new algorithm better than anything before and that we could use. Ed suggested we should have a hotline for support--- you have a problem with anything, you call on the phone you get help right away. And there was a list of things like this.

HAIGH: So was SSP ever unbundled or did it remain free for its whole existence?

DUBRULLE: Remained free, totally free.

HAIGH: And the new PL-Math package was also free?

DUBRULLE: No, there was a license, I think there was license.

HAIGH: So when Ed was talking about a hotline support he was also imaging that users would actually be paying for the package.

DUBRULLE: Yes. Anyway, the license fee was that you were paying so much per month.

HAIGH: Yes. It would make sense for it to be a subscription type service where you actually got something in return.

DUBRULLE: Yeah, that's right, and that would be real service, complete service. So, Ed as manager of the group, made a proposal. Some people from White Plains came, people from the business office, other people, and they listen very politely, and they said "well we can do this, we can do that, but no hotline, we cannot retain a board of advisors, we cannot do that." So Ed got angry and told me that he was very unhappy. One day he said he had met Chuck Johnson, a former IBM applied science member. Applied science was for some time at IBM a bunch of guys who were highly educated in mathematics or science, could crack problems, but were working in a sales environment to assist sales reps. Chuck apparently, from what I know---that's indirect---had married a woman with a money and had then retired. He wanted to create a business. He was thinking of creating a software business, so Ed of course told him about the idea that IBM didn't want.

HAIGH: Do you know, by the way, at what point the applied science operation was disbanded?

DUBRULLE: That was certainly before 1967 because it was already defunct at that time when I arrived in the U.S.. I didn't know it before because there was nothing similar in Europe.

So, Ed and Chuck talked together. Ed told me they were going to start a new company and he'll be the president, Chuck will be the chairman of the board. He asked me not to tell anybody until he'd be gone. He also said that if I wanted to join, Icould be be director of the numerical analysis department. I said that, well, I was still on that Mickey Mouse visa, couldn't leave IBM, I couldn't even have a regular job. I was still an employee of IBM France, and IBM France was starting showing signs of discontent about my temporary assignment being extended, extended, extended on the demand of IBM U.S. And besides, Christiane didn't like the climate of Houston, although we had a good time---it was a party town. So, Ed went, took along with him our secretary, a programmer of ours, one of the two girls we had hired in White Plains, Nancy Bosten, and took along a few people from the Scientific Center.

HAIGH: Do you remember the names of the women programmers?

DUBRULLE: Nancy Bosten, she ended up marrying Ed much later after Ed was widowed. She was younger by seventeen years, I guess, but anyway . . . So Ed took along with him people from different areas of the Scientific Center, to the point that some people in the Scientific Center got really annoyed and said that it was reaching a point of questionable ethics and legality as well. But IBM didn't do anything about it. And so I stayed with IBM and Ed went and that was the start of IMSL. Ed stayed in touch for a while. The guy who went with Ed, who was, he was something like vice president for research and development or whatever, was Olin Johnson. He's still teaching at the University of Houston and can be found on the Internet. He's not been with IMSL since 1974.

When I was in Houston I got a call from Palo Alto, more exactly Menlo Park, California, saying that I might be needed on a job to develop a library for a new machine called the System 3 Model 6, which at the time was IBM's first answer to the Digital Equipment minicomputers. It was a small computer for the desk, with a screen and a drawer in which there was a memory, a big disk. There was of course a printer, and I think there was a tape punch but I'm not sure, and it had new little cards.

HAIGH: And was that aimed primarily at the scientific market or was this just one application for it?

DUBRULLE: It was just one application for it, yeah. So, I said yeah I'm interested in going to California. Now that I know a little bit about Texas, why not California? So I sent a letter to IBM France and I said my assignment should be extended by another six months and see what happens then. So I got a letter back from the manager of the Scientific Center. He was not too happy, but he said okay. I went to Menlo Park in California, which is not far from Palo Alto, and I met the new people. What a change! Now we are talking about January 1971. It was nothing like what I had known at IBM before, it seems that there was a drastic change of culture and business culture also, and business vision. It looked like all of sudden IBM had been invaded by accountants, that is, if you know about financial matters you can run any outfit to develop any applications in any field. The guys were nice, but they had an eye focused only on financial things, so

when I got there the department manager asked to tell him about my trade and about me. I told him about by trade, about mathematical software, and when I was finished he said "I don't understand. You developed a piece of software that solves equations, fine. It's like building a table, and then three years later it might not be a good looking table, but it's still a table, you can use it as a table, and it's as good as any other table for function." So I started over again telling him that software was not the same kind of thing, that depending on the state of the art you build tables that are not quite level or that can break, and that progress should make you build a better table . He said, "I cannot understand that, I cannot understand that." Well, his background was in business, but he was leading that group dedicated to the development of some scientific software for the System 3, Model 6. Well, you know, incomprehension was not unusual in that kind of trade. So, he gave me the problem at hand, that it was to design this math library for the System 3 Model 6, and it had to be in BASIC.

HAIGH: How did they decide that?

DUBRULLE: Ah, that was my question. Why BASIC? The answer was "Because when we were not looking, DEC invaded the work place and uses BASIC." And when I asked, why BASIC, why does DEC use BASIC, I was told that the logic is that people in college learn to use BASIC and when they get a job in industry they don't have to learn any other language, and they don't want to anyway. They resist learning another language because it's too complicated---Fortran is too complicated. So, it had to be BASIC. I looked at Basic. The IBM Basic had no subroutine capability, so I said that we couldn't do the project. Well, I HAD to do it. I asked for the specs of the DEC Basic. In DEC Basic there was some kind of subroutine similar to Fortran's but not quite, so I aked why IBM didn't have the same kind of thing. The Basic developers said that we had to use the 'minimum' BASIC because we wanted a maximum audience and some people might use versions of Basic that are not quite up to the latest. So I went back to my office and I wrote a long memo to the department manager and his boss, the Development Director in White Plains, to tell them that the project is not feasible. And I get a call from White Plains which says that there is a subroutine capability in Basic, it's just like Goto, but it can be used. And then I get another note from the Director of Development -

HAIGH: Maybe you mean the Gosub instruction.

DUBRULLE: That's the Gosub, you're right –

HAIGH: It's like Goto, but when you've finished running a subroutine you can Return and that's the only difference.

DUBRULLE: That's right, that's right, but that didn't fit very much the needs of a mathematical library. You know your stuff, no doubt about it. Does anyone use Basic anymore?

HAIGH: Yes, but not as a general purpose programming language. Visual Basic in the 90s had the same reputation that Cobol used to have as the language to write a custom business application in. But usually you were just writing snippets of code in a nice window driven environment. No more Goto.

DUBRULLE: But to me Goto and Gosub were not that different anyway. So then I get a call from the Director of Development in White Plains---that's still Industry

Development/Scientific---and he says that I must develop the package, to which I finally agree. So, I built a contraption. My design was to have a set of stand-alone programs with the computational kernels. There was some input and output flexibility. You could get your input from tape, put your output to tape, or get your input from the keyboard, put your output on the screen, or tape or whatever, or the printer. What you could also do is take that kernel and merge it with your program. Of course I had to avoid conflicts in variable names, so I had to name all my variables with special characters like pound or such. So, when I presented that contraption, I warned, "here it is, but trust me, I've been getting a little bit of experience in these past few years, and this is going to be a flop," and I remember writing "flop" in capital letters in a memo, because usability was so impaired. The answer was "do it." So I said okay and I made a design based on the methods we had used for PL-Math.

HAIGH: Now was Basic at that point able to handle the mathematical side of it?

DUBRULLE: Sort of. The BASIC semi-interpreter (it was not quite an interpreter, I don't know how DEC handled the thing but IBM handled it different way) would take the Basic source and translate it into some kind of pseudo code that was then interpreted, not compiled. Yes, you could do about anything, almost.

HAIGH: Did it, for example, have floating point?

DUBRULLE: Oh, yes. Yes, it had floating point. Now the people that developed the BASIC translator interpreter had overlooked a few things. Like in floating point. If you have integers, which are representable integers in the machine, and if you multiple those integers as integers, the result is a machine integer. The floating-point version of the multiplication of those integers should give you an exact result in floating point. Well they had overlooked that kind of thing. So I went into a few difficulties, which were difficult to trace. I think it had to do with quadrature, where there were powers of two involved, and I had to fix that and there were a few quirks like this that I had to fix.

HAIGH: So that would be something like two times two equals 3.9999999999?

DUBRULLE: Yes, about. And then I called the people who had written the compiler/translator/interpreter and they very patiently started explaining to me that in finite-precision arithmetic things were not that simple, which was ridiculous. Anyway, so I finish my design, I submit it with an estimate of cost, time and resources. Then I'm told that meanwhile IBM Germany had been doing the same kind of proposal and were set to start the project. The department manager is a little bit disappointed because it's one less item on his resume, but he didn't know what to do. I told him to ask for figures, how much would is cost to have it done by IBM Germany. So he came back with the figures. Well their cost was twice mine, and in time it was about one and a half the time allocated with hypothetical resources. So, he said he wanted to do it in his shop, and finally got it in his shop. I pointed out that one of the reasons it can be done that cheap is because I was in charge---you know, being very modest. I added, that there was aa question about extending my temporary assignment once more. Paris said no, and invited me to go meet the president of the French sales division who was on rotation assignment in New York before becoming president of IBM France. I went there and the man said that I had to return to IBM France. I asked why and he said that I was needed you there. I asked, "For what?" Well, there was nothing. I said goodbye, went back to Menlo Park, and I reported

that there was no deal, that France didn't want to extend my assignment. So, the department manager, Cliff, said that he'd try to hire me and I say okay. But he warned me that it was going to be no picnic because IBM has a complete hiring freeze this year. IBM was going through rough times, and even research didn't hire. He said that he didn't know if it was going to be successful but he was going to present it as a quasi-internal transfer from IBM France to IBM US. Meanwhile I would start working on the project. I said okay. So, I started working on the project. In July, I guess, I went to IBM France to resign, shipped the stuff that I had in storage in Paris since 1967, that had been four years, and went back to Menlo Park. When I went back, my application for employment was still in the mill, but I had burned by bridges with IBM France. Finally I got hired into IBM here in the U.S.

HAIGH: Oh that seems a nice point at which to conclude the first session of the interview and we will continue tomorrow. End of Session 1.

[Start of Tape 3, Side A]

Augustin Dubrulle interviewed by Tom Haigh, Session 2. This is the thirty-first of July 2004, this interview also is taking place in Arroyo Grande, California.

HAIGH: So picking up from yesterday's session. I believe that you had described the beginnings of the project to construct a BASIC library for the small IBM minicomputer that was under development in the early 1970s. At the end of yesterday's interview you had described the process by which you became a permanent employee of IBM U.S. I don't believe you had told us whether the library was ever actually finished or not.

DUBRULLE: Yes it was finished. It was finished on time and within budget, which was quite a performance because as usual I didn't get the personnel that I had asked for, in quality or quantity. But it worked out. For about a couple of days after that, I was a hero. Actually I had to design a special test plan because of constraints of time. I couldn't follow the standard test procedures of IBM and I worked out a different test plan that allowed some kind of pipelining of testing the different units of the product as there were developed. That released of a lot of money and time and I got a Division award for that. So, the project was finished on time, within budget and for a few days, as I said, I was a hero. Then a week later, all of a sudden I was surplus: we'd done the project on mathematical software, so now what? Well we're going to do something on database software. I didn't want to go into that, I had many friends at IBM who for local and emphemeral reasons had become jacks-of-all-trades. Finally they ended up not being specialists of anything, losing their initial specialty, and after a while they didn't know where to go, didn't know what to do. Even though they were senior people, they were not anymore useful to the company than, let's say, entry level programmers. So I wanted to stick to my guns, and we got into some kind of arrangement that I would do special assignment for the department manager who had in mind to develop a gigantic system for engineering applications around a database which was described at the time as a daisy. There was a kernel at the center of the daisy, which was a database, and to that kernel you could hook something like engineering computations, personal management, and other things. It was a big contraption. So there was some kind of a task force, and I sat on it in an advisory capacity, and I think it was going nowhere. So I started looking for a job somewhere else within the company. I knew that that department was going nowhere.

HAIGH: And did that group ever produce a database system?

DUBRULLE: No, no. I think they produced some kind of database system but which had nothing to do with the original purpose. I don't even know what happened to it. So at the same time I applied for the so-called green card, permanent residency, which allowed me to be gainfully employed by IBM. There were a few problems. First of all, the INS lost my file so I had to reapply. There were little details in which IBM was involved also, and I had to look into international immigration laws and things like this. I had to make a case in support of the paperwork. I became a lawyer in that area for a short time. It was interesting. So after that. I spent about six months reviewing proposals for software development which were not very good, which the Director of Development in that area rejected systematically anyway, because his dream was the paperless office or something that would bring in billions of revenue.

HAIGH: And what was the date by this point?

DUBRULLE: Well, I finished the product by the end of 1970.

HAIGH: Did it have a name?

DUBRULLE: The name, oh, Math-Basic.

HAIGH: And do you know if it was ever much used in practice?

DUBRULLE: I don't think so. I think it was used but not much. Actually I should do a internet search to find out, but I don't think it was used very much. What I think happened, because it was a bit of a contraption, people used it mostly in one of the modes that I had provided, which was merging routines into the user program. But I don't think it did very much good.

HAIGH: Did it have any modes where it could be used interactively?

DUBRULLE: Pardon?

HAIGH: Did it have any modes where it could be used interactively? That being the main advantage of Basic.

DUBRULLE: Yes. You could enter data with prompting from the keyboard, or from a tape, and also write the data out on the tape and show it on the screen. Some programs were supposed to work together, so the output of one would be the input of the other, and so on. Anyway, I wasn't too happy about that product at all, but in a way it was my ticket to the U.S.

HAIGH: And during this period of the late 1960s, and early 1970s, were you aware of any other significant libraries that were being produced by equipment manufacturers, whether elsewhere in IBM or at any of your competitors?

DUBRULLE: Well, I knew that there was some work going on with EISPACK, but EISPACK was totally different.

HAIGH: Right, that was being produced at Argonne.

DUBRULLE: Yes, and it was a very narrow, the topic, it was a single topic: eigenvalue problems.

HAIGH: Let's say CDC or Univac or anywhere else?

DUBRULLE: I don't remember, I might have been aware of it at the time, but I don't remember.

HAIGH: So in general this wasn't an area that any of the computer manufacturers were putting significant resources into by that point?

DUBRULLE: I don't think so. But in research centers and such there were developments. There was something going on at the Boeing Aircraft Company. One of the principals in the development was Chris Newbury. I looked him up on the Internet recently, couldn't find him, but the Boeing library was something at the time. There was also at Bell Labs the PORT library, which was intended to be a model of portable subroutines. Portability was a big issue. Actually I started working on the idea of portability when I was in France, that was already a problem in, let's say, the mid 60s, but portability was a big thing at the beginning of the 70s. They are the two libraries that are really, that are the most remarkable in my mind now. And well I suppose I don't have to describe the project that produced EISPACK, it was the NATS project.

HAIGH: The other interviewees have given a lot of detail on that.

DUBRULLE: Okay, and it was funded by NSF.

HAIGH: So did IBM ever produce a significant library available in Fortran after SSP?

DUBRULLE: No, . . . yes and no. Meanwhile, very few people knew it in the U.S., but IBM Germany had produced a library called SL-Math, which was a version of PL-Math in Fortran. And we didn't know about it. It had been produced really in a hurry. Actually I got the news because when I got a phone call, I think it was in late 1970, I got a phone call asking me to go to the SHARE meeting which was, I think, in Orlando, Florida, early 1971, to announce to IBM users that SSP wouldn't be maintained at all, and would be be replaced by SL-Math. So, I did that and I gave all the good reasons for it, that SSP was obsolete and so on and so forth, and I'd never spoken before such a hostile crowd.

HAIGH: Had you visited SHARE meetings before?

DUBRULLE: Yes, one time. I think I was in White Plains, I know it wasn't Houston. So, anyway, I did that and I was not very well received. People objected mostly to the fact that they had to pay. And there were a lot of old timers, later on at night---at SHARE there was an open bar at night called SCIDS---a lot of people there told me their war stories, how SSP was great and so on and so forth, and it was a shame. They were about the same people who years before were telling me that SSP was a terrible thing. I think it was later on, I think it was when I was in Florida at SHARE, that I was approached by somebody from the Harvard MIT computing center. She told me that they had bought SL-Math and that they were not too happy about it because they had picked thirty subroutines at random and they were all buggy. And it was so bad that Garrett Birkhoff of famed Birkhoff-MacLane, had gotten into the situation and had tried to contact the people who had developed the programming in Germany.

HAIGH: Yes. Do you know what their names were?

DUBRULLE: Nope. I don't know what their names were, they were not the original team headed by Albrecht Blaser. But at the time the development team had been disbanded, the status of the program was maintenance. It was maintained by I think the same kind of people who were maintaining operating systems. They didn't know anything about it. Birkhoff had tried to, had gone to higher levels of the IBM organization in Germany and had got nowhere. It was very frustrating. So, instead of flying back to California I flew back to Boston and I had a meeting with Birkhoff, and I told him that I would do whatever I could. He told me about the situation, he was incredibly patient and courteous, very gentlemanly. So, I tried to find an outfit here in U.S. who would take the thing and fix it up, and I could get nowhere. I tried to pull all the strings that I could find, even an IBM fellow who was at the IBM Scientific Center in Palo Alto, and was an alumnus of Harvard. And finally, almost independently, somebody in research took the job. His name is Fred Ris, a fine numerical analyst who had gotten his Ph.D. at Oxford, with Leslie Fox, and who became later a principal in the group that defined the IEEE arithmetic. So Fred Ris was in a way in that IEEE group the IBM representative, and he was very active, very technically active. Anyway, Fred took that up and I thought, well, things are in good hands. I don't know exactly what the audience of SL-Math was but I think it was used, and after that it was a reasonable product.

HAIGH: By the way do you have a date on when SL-Math would have been first offered as a product and when these fixes would have been made available?

DUBRULLE: It might have been in early 1972.

HAIGH: And do you know how long it was offered as a product for?

DUBRULLE: I think maybe through the 70s, because I know that in the 80s IBM developed another really high powered good library to which I contributed. It was designed for machines with hierarchical memories.

HAIGH: Okay. So to return to your career at IBM after the completion of the Basic library?

DUBRULLE: So after that I switched divisions. I was working with APL because in the department where I was there was an APL account that belonged to nobody, nobody was billed for it, and it was available. So I used it to do some research that had to do with mathematical software and numerical analysis. So, by using APL I got in touch with APL people. I knew already some APL people from IBM research including Ken Iverson, but then I met some people who were in Palo Alto working in an APL group for a division, that doesn't exist anymore, called ASDD, which meant Advanced System Development Division. It was some kind of half-way house between research and development, it was advanced systems. So I went to work there for the APL group. I wrote two conference papers there to put some features relevant to floating-point arithmetic into APL. I liked APL because it was fast and you could develop programs very fast, very quickly with it. Of course APL was for a lot of people like evil. You would say APL and they'd get into a fit and say, "Not APL, Pascal, Pascal. APL is terrible." I didn't care, anything that worked well was fine to me. Programming languages are only tools to me, actually. I'd gone from Fortran, PL/1, Basic, APL, who cares. So I joined that group and everything was fine until the division was completely reorganized. I'd been working there for almost

a year, it was very pleasant. The division was reorganized to host a gigantic effort called Future Systems, the software side of it. I don't know if you've heard of Future Systems.

HAIGH: The existence of the project is quite well known. I believe IBM still guards many of the details because it didn't have a very happy outcome.

DUBRULLE: Sure. So, anyway, it was really ambitious and the idea was to build up a new system, a new family of systems, hardware, operating systems, software, compilers, everything. In parallel, on the east coast, there was another task force to develop the floating-point arithmetic. So I was assigned to meet with that task force on the east coast and discuss things on floating-point arithmetic. But there in Palo Alto I was asked to look at the possibility to make a design to develop a package of elementary mathematical functions that would be used by all compilers. There was the idea in computer science of building a universal compiler, and the universal compiler would have a front end, a core, and a back end. The front end would change depending on the language, there would be a front end for Fortran, one for PL/1, etc. There would be also a back end for the same things that would be useful for the analysis of code and diagnostics and so on, and in the middle there was the real compiler, translator, whatever. So that was an enormous thing. The idea was okay: you have a universal compiler, you have a universal math library of elementary functions. PL/1 was the big headache because in PL/1 you could call the sine function and give to it a complex argument, which would give you something like cosh and sinh---I mean the hyperbolic functions---but you had also cosh and sinh functions. I thought that more than accuracy, consistency among all possible combinations of functions should be maintained, and that was the headache. So, I worked on that and I worked also with the taskforce for floating-point arithmetic.

HAIGH: And was there a team of people working on the math library for this system?

DUBRULLE: For the math library, at the time I was the only guy, I was working on design ideas and things like this. Also later, when things would be sorted out a little bit, I was to make a request to staff a group to work on it. It was very interesting. At some point I had found some ideas to reconcile the different functions and maintain consistency. I was about to get in touch with Kahan and Jim Cody, with the idea that I'm making that kind of design and I should get those guys, who were supposed to be the top dogs in the area, to look at it. That was an entirely new thing for me but it was very exciting, it was hard but exciting. I had read all the books I could read on the subject, and I gained a lot of knowledge on that. And then when I was about to start and make my proposal on the staffing and so on. Decimal arithmetic had been one of the options and there was a lot of resistance from the engineers to decimal arithmetic because it was unfamiliar and the task force on the floating-point had come to the compromise at some time of having the two arithmetics residing in a machine, decimal and hex, basically.

HAIGH: So they were both going to be supported on the hardware level in this design?

DUBRULLE: Yes. And I'll tell you where the decimal idea came from. So, well then came the news that decimal arithmetic had been killed. Which in a way changed my work. A year later the whole thing was disbanded. Now where did the decimal arithmetic come from? It came essentially from Kahan, shortly before the assembly of the task force Kahan had spent a sabbatical in Yorktown, IBM Research, and he had looked into those questions. He had concluded that, well, if we're building the best system, the smallest

granularity is binary, and we can design a "perfect" binary arithmetic. He had started working already on what would become the IEEE standard, and he had all those things about unbiased rounding, directed roundings and so on. And he asked if binary is the favorite choice of the numerical analyst, what should be the next favorite choice besides binary. Hex is not too good, but IBM hex arithmetic is clean at least. It's better than the binary arithmetic of Cray, and that of CDC. I don't think it was Cray but CDC that had roundings pretty much haphazard. Anyway, Kahan proposed a decimal arithmetic. He pointed out that the big advantage of decimal is that there is no need for input/output conversion as people can read decimal. Of course it takes a little more space for storing floating-point numbers, it's not as efficient in a way, it takes more space per digit, but it's the good compromise beside binary. So that was the idea of using decimal.

So the whole thing was dead by now and of course I was looking for a job.

HAIGH: In what year was the project cancelled?

DUBRULLE: I think the project was canned in something like in '75. So, I was looking for a job, again, and finally I found a job still in a different area. I found a job in a shop that was doing computer performance measurements, hardware performance measurements. I liked very much the manager, he was a good department chair, and he was a veteran of supercomputing, he had participated in the STRETCH project, and he was the kind of guy who was really excited about fast computers and such. And he said that the general mission of the department is performance measurement, and that I had to write a job description that would fit in there. So I set out to concentrate on the performance measurement of computers for scientific applications, of course. He said, that it was a good idea and asked about the first thing I would do. I proposed to put together a set of programs that do scientific calculations of very different kinds. There will be something about graph handling, shortest path, longest path, whatever. There will be things heavy on computation, like a solution of a three body problem, and so on and so forth. So I developed that set and it turned out to work pretty well and it was really hitting all the functions that are used in scientific computing. We had a computer with probes in the hardware that didn't intrude and we were taking measurements. That was really interesting. And there was a fellow who had written a set of APL program to collect the measurements which was really neat, I was quite happy.

HAIGH: What was the name of the group that this work took place within?

DUBRULLE: I think it was System Performance Measurements and Modeling.

HAIGH: And where were they based physically?

DUBRULLE: They were based in Palo Alto. After that they moved to, I think it was Cupertino, a place called the Pruneridge, and then after that they got evacuated to the big new development center of the General Product Division which was south of San Jose, which is called Santa Teresa, but before IBM went there it was called Coyote Gulch.

HAIGH: And how long did you spend with this group?

DUBRULLE: I spent one year with this group. I failed to mention one thing, before I went to this group, after Future Systems, I spent a little more than a year with a group, in the General Product Division, doing nothing important. So the group moved to Santa

Teresa. Now we are in 1976, the manager said that if I wanted to stay in my office in Palo Alto that'd be fine. He was the old-fashioned manager, you wouldn't write to him activity reports. He would come and talk about what happened last week or last month or whatever, and that was it.

Well, I got evicted from Palo Alto because another outfit was moving in. So for two weeks I commuted from Los Altos, which is very close to Palo Alto where I was living, to Santa Teresa which is close to a thirty mile drive, and at rush hour it was not no fun. So, when I was told to go down to Santa Teresa told the manager that I liked very much working for him but that the commute was a problem, and he said that he'd been thinking about that and that I'd done good work for him, but that there were places at IBM where I would be more useful to the company. And I said "Okay, like what?" and he said, "Well, the Scientific Center in Palo Alto."

I'd tried to get in there, but Scientific Centers were always under budgetary and head-count constraints. Wherever they were, to whatever division they were attached because they were passed along among different divisions, they were always considered a little bit as overhead because they were spending money and one couldn't measure the revenues to the company that they were providing---same old question. So I had tried and it did not work. My manager promised to work for my transfer, and it turned out that after two weeks of my commuting from Palo Alto to Santa Teresa I got the job at the Palo Alto Scientific Center. It was to work initially on a project on geophysics for oil exploration, computational geophysics, and that big project had to do with using seismic experiments to determine the shapes of the geological formations of an area. From the shape of these geological formations and their nature you could infer a probability to find oil. So it sounded very interesting. It had to do, a lot to do, with wave scattering, wave equations and solving those equations. So, of course, I signed up and I got on the project. There were two of us on the project. The goal of the project really was to work as a computational geophysicist for a few years and at the end to write some forecast of the computing needs of the industry. Computational geophysics was using a lot of megaflops. So, I had to establish some credibility, because people in the field were asking whether I was working for, say, Amoco, and were wondering why IBM could be interested in exploration geophysics. To establish credibility you had to publish. So I worked on that project for, until 1982, roughly four years, '77 to '82, and I think I wrote three papers on geophysics subjects. I used the first year to learn about the trade, learn the vocabulary, the jargon, and so on, to get along with the other guys, and after that I published the three papers. We then wrote the forrecast report, and I stayed with the Scientific Center.

I was then offered several possibilities for another project, one of them had to do with artificial intelligence, there was another, I don't remember what it was. I declined and instead suggested that if there was any kind of supercomputing coming down the pike, in any form, I wanted to be involved in software design and numerical methods. It turns out that actually there was some supercomputing coming down the pike. There was some project design on vector computing, and as usual IBM had two competitive designs. In the meanwhile I had written a technical report in a real hurry, in 1979, between writing for the geophysics project, I'd written a report about matrix computations in hierarchical memory machines. That was a collection of ideas that I had thought about during the 70s. The report is not very good. I reread it not too long ago. Because I wrote it so fast it's not

very good, but the ideas are there, the problem of computing efficiency with hierarchical memories started out in the early 70s.

[Start of Tape 3, Side B]

DUBRULLE: When I'd been working with the System 3, Model 6, that was a machine with small memory. The secondary memory---a disk---was not very fast, so we had the usual problem about programs being slow because of memory access. That was not new to me. In the early 60s when I joined IBM and worked with the 7090 we had that problem. Secondary storage was tape. At the time a lot of people were saying that if you want to multiply two matrices that reside on tape, one must be the matrix itself and the other must be in transposed form, because people had their mind on finding the elements of the product via scalar products, and scalar products address the row of one matrix and the column of the other. And then one day I suggested that if the matrices are both stored column-wise, one could do a multiplication without access problem. One could write the mathematics of the multiplication in terms of the columns of the matrices. And I did that for one customer at the time, and the guy was ecstatic because to him it was a big problem to transpose the matrix on tape. I didn't think too much of it but some people thought I had invented something terrific, which was not true of course, it was just mathematics.

So, anyway, when I worked with the System 3, Model 6, I knew about that kind of problem. So I wrote a short article for the *IBM Journal of Research and Development* about the importance of memory hierarchy and how to use it based on the System 3, Model 6. And the same year Cleve Moler published an article like this about Gaussian elimination.

HAIGH: And what year was this?

DUBRULLE: Oh, these articles were published in 1972. Cleve's was published a few months before mine, I think, but they were addressing different things. Later on I talked to Cleve, when he was spending a year at Stanford. We talked about these kinds of problems and other problems. We used to go and have a sandwich at the Town & Country Village in Palo Alto---it had a great deli shop---we would buy a sandwich, and walk around talking about numerical analysis. Technical conversations with Cleve were always interesting. So I had written that report now in 1979 about matrix computations and –-

HAIGH: And was this 1979 report ever published?

DUBRULLE: No. It was an IBM Scientific Center technical report, it was available to all but I don't think it was widely distributed. Anyway, some time later I was told by somebody (and I don't know if it's true) that with all its warts the report had found its way up the decision process about choosing between the two super computers under development. They were vector computers, one was Cray-like and the other one was a machine with cache, which was not Cray-like. Finally the machine with cache won because it was way cheaper, it didn't have the interleave memory but there were good indications that you could write good software to get good performance with it. I was told that my report had been used to support this idea.

HAIGH: Did that machine ever make it into production?

DUBRULLE: Yes.

HAIGH: What was it called?

DUBRULLE: It was called 3090VF. There was a main machine, the 3090, and on option you could buy the VF that went with it---VF meant Vector Feature. The VF was not an outboard vector processor, no, it was really an integrated design, but it was detachable, which was interesting. So the machine had a cache, and you could do vector and scalar computation. It was a good scalar machine too. There was a compiler for it, a vectorizing compiler, which was taking certain programming constructs and automatically translating them into vector operations, and it worked pretty well.

One of the underlying ideas in the 1979 tech report was that when you write a program for hierarchical memory you can put certain characteristics relating to the hierarchical memory into the mathematics very simply. For example, let's suppose that you are working with matrices. Matrices were a big case because they were the kind of problems that were causing losses of performance when the memory was misused. A matrix in Fortran is stored by columns, so if you address the elements column-wise, usually what happens in the cache is that you get a chunk of the data set when you call one element, while a certain numbers of other elements are present in the fast memory. It does not cost much to access theese elements. So what you have to keep in mind about the hierarchy of the memory, since Fortran stores matrices by columns, is that you must think "columns." So to think columns, you write your mathematics instead of representing the matrix by A, you expand A in terms of its columns, then you work the formulas, and you get an algorithm that works on the column of a matrix. The principle is simple, but to apply it to a complex algorithm sometimes is not simple. So I had to work with a machine that was a great platform to show that this problem of formulating architecture into the mathematics is really the way to go.

We're in 1982. So, anyway, I'd signed up for that thing, for a supercomputing project. We had some kind of contract with Cleve going on. Cleve visited Palo Alto and we talked about research, and I showed him a few things about my work. He said he remembered in 1972 that I was thinking of going back to school. Actually I wanted to apply for an IBM program called University Resident Program to go to Stanford. Gene Golub had told me that a recommendation by Wilkinson would get me directly on the Ph.D. track at Stanford. I would have had to do maybe a year of residence and then after that it would have been's all dissertation. Well, the third line manager above me was dead set against it, against scientific computing, that he thought was of no interest to IBM. I was pretty upset, but anyway, and I couldn't really quit, I'd taken on a new mortgage. So Cleve asked if I had thought about going back to school and I said "Well if I can find a way to support myself." He suggested that I could come to the university of New Mexico, and I said "Okay." So I went around to find a way and my second-line manager who liked me found a project, some IBM assignment for me in Albuquerque which would allow me to satisfy my one-year residency. I did it, and then when I came back to Palo Alto, I started working on the dissertation.

HAIGH: What year was it that you spent in New Mexico?

DUBRULLE: School year '82-'83. I went there in August '82, I think, that was a semester system, and once a month I would commute back to the Palo Alto Scientific

Center on weekends to take care of some other business that I had going in Palo Alto. Well, it was quite an experience because I was forty-seven. There are things you don't take that easily at forty-seven, but it was very interesting. I talked to Jim Wilkinson about it and he thought it [*going back to school*] was a good idea. He said it would take me out of the grind and give me maybe a different perspective, a breath of fresh air. Well, it was really hard work.

I came back to Palo Alto and started on my dissertation. The first subject didn't work very well. Then I went to the second subject, which had to do with the bidiagonalization of matrices as an image-compression device. That was one of the things that Cleve Moler had been puzzled about for some time. There was group doing big-time image processing in Palo Alto, and the head of the group was an IBM fellow, Harwood Kolsky. He was a great guy, an alumnus of Los Alamos, who had been associated with a lot of historic supercomputing projects. Since there was more engineering than research going on in his group, he proposed to support a good part of my dissertation work on his project, which was very nice. So, anyway, I finished that in 1986, just before I got to be fifty-one.

HAIGH: And how would you describe Cleve Moler as a dissertation supervisor?

DUBRULLE: Hands off, really hands off. He would come once in a while and ask what was going on, and I would tell him, and he would tell me to keep going and that was about it. Hands off---that's not a criticism I make---but definitely hands off. In the meanwhile I'd talked a bit to Gene Golub at Stanford about . Actually Cleve left the University of New Mexico to go to Intel before I finished.

It's when I came back from my dissertation excursion, that was 1986, that I really got involved in supercomputing for the 3090VF. At the time the Palo Alto Scientific Center had started a new group which was called NIC, Numerical Intensive Computations. The manager of that group---he was second line manager---was a good friend of mine. I'd known him in Houston. He had a Ph.D. from Rice. He was looking for projects because he was in a hurry to get some funding, and to get some funding you had to show projects. All the proposals he was getting were pet projects of people who had written the proposals before and had been turned down because they were of little interest to IBM. So, one day he said he needed a project to start the group and that project had to have some kind of resonance in the numerical analysis community, and it had to last less than a year, and it had to produce good results for performance of the 3090VF.

So I went back to my office and I thought about it for a couple of days.: how about EISPACK? I could take EISPACK, fix a few things that needed to be fixed because of age---historical circumstances---and then do my thing on the subroutines that were really requiring heavy computation for the general eigenvalue problem. So I made the proposal and the manager was quite happy. Then I took my dog-and-pony show to upper management, made presentations, and so on, and they thought it was a good idea. I started out on that work all by myself. Of course I had to make choices here and there. I decided to retool algorithms only for the long precision version of EISPACK. EISPACK has two versions, for short precision and long precision. I decided to work on the long precision version because very few people use the short precision version, and when they use it they use it on small matrices. If they use it in small matrices, well, there are not

going to be too many cache faults, given the size of the cache. The problem was really with long precision, and that's what I was going to address.

So I worked on the long precision version and the first thing I did was to take the eigenvalue solver based on the QR iteration for general matrices. I thought "Okay I'm going to apply the famous principles. The matrix elements must be addressed through the compiler, not in Fortran necessarily, but through the compiler, so I must know what the compiler does when it vectorizes Fortran constructs. The transformations that are applied to a matrix in eigenvalue problems are called similarity transformations. You have to multiply the matrix, in a way, to the left by the transformation matrix and to the right but the inverse of that transformation matrix. Of course things don't happen exactly the way I said it, you actually don't use the inverse, and you don't multiply by a matrix, you perform transformations. It turns out that for maximum efficiency I had to use vector operations for the right transformations, and scalar operation for the left transformations. That was giving me maximum efficiency, which ran counter to what a famous consultant had told IBM before, that everything had to be vectorized.

HAIGH: And who was this famous consultant?

DUBRULLE: I won't name him, he's a good guy. But you visit a company for a couple of days and you're given a massive problem and you work on it for two days, three days, and you cannot go to a level of detail that will give you something perfect.

HAIGH: Now had there been, that you were aware of, earlier projects to rewrite EISPACK for other vector machines such as the Cray?

DUBRULLE: There were earlier projects but they had nothing to do with hierarchical memory. The latest project when I started that I think it might have been something like EISPACK 3, by Cleve Moler and Dongarra. And there were interesting things in it like a function for computing Pytagorean sums by Moler, which I had generalized a few years before, which I had generalized and analyzed in a paper. So I started with that, and I got results that were amazing to the point that some people in the Scientific Center--- not the significant people---who were in the business of so-called vectorizing customer programs, found suspect. For example, they were vectorizing with great pain some loops in their programs, and they were going around proudly claiming to have saved twenty-five percent of the running time. I was getting fifty percent. If you were comparing algorithms from the original EISPACK and my EISPACK, they didn't look alike at all. They were the same numerical methods, but their implementations looked completely different.

HAIGH: So, did you achieve your goal of having this work done in the compiler or were you rewriting the actual Fortran code?

DUBRULLE: I would rewrite the Fortran code. I wrote the Fortran code for the compiler and for the machine, if you see what I mean. The most interesting thing is that every month about a thousand lines of code went through my hands, either brand new or modified, and the productivity was incredible. But why? Because I'd written the mathematics in detail before starting the programming. The programming was just a formality. I think the project lasted less than a year. I couldn't believe how productive it was, which showed several things. One, that the idea of inserting machine architecture

ideas into the mathematics worked very well. Two, that it's really economical to use a grossly overpaid programmer, a grossly overqualified programmer, to do the job. And three, that testing didn't have to be done by an independent organization. So, I was pretty happy with that, and when I presented my results nobody wanted to believe in the thirty lines of code per day that worked, nobody thought it was true, but it was.

HAIGH: And was that version of EISPACK widely used by customers of the machine?

DUBRULLE: I don't exactly because we didn't keep track of it. While I was doing that for the double precision, for the single precision there was still some work, a little bit of work to do, not as deep, which had to do with inserting compiler directives, things like this, and rewrite a few loops for vectorization. And this I subcontracted to one of Cleve's former Ph.D. students from the time of the University of Michigan, Alan Cline, at the University of Texas at Austin. And he worked remotely from Texas, using telephone lines, on the 3090VF that was in Palo Alto. It worked pretty well.

HAIGH: Do you know if there was any influence of this project on the subsequent LAPACK project?

DUBRULLE: In a way, in a way. I'll come to that. Meanwhile, IBM had developed a library called ESSL, the Engineering Scientific Subroutine Library. Quite a bit of this design had come from Yorktown research, and one of the principals in these activities was Fred Gustavson. So having finished EISPACK, I compared some of my routines to those in ESSL, and low and behold, mine were way faster. So I got in touch with ESSL and they were very glad that I would give them those routines.

HAIGH: And did ESSL work across the range of IBM mainframes?

DUBRULLE: Yes. Except that the first version, for the 3090VF, here and there had pieces of code in assembler. When we finished this project, Alan Cline put the whole package in NETLIB, the scientific routine repository at Oak Ridge. Meanwhile I had kept in touch for some time with IMSL. Ed Battiste had left by that time and there were a lot of stories, but there was a good friend of mine, Richard J. Hanson, whom I'd known since 1969. We had met at the University of Michigan. He's a fine numerical analyst, he co-authored a very good book on linear least squares problems. A serious guy. So I told him about what I had done and he said he was interested in it for the IMSL library for hierarchical-memory machines. So he got some of the codes and incorporated them into the IMSL library….

HAIGH: By that point was there any concern within IBM about having its code incorporated into rival commercial products?

DUBRULLE: Nobody cared. You make it public domain because maybe everybody will benefit, but you benefit too, so it's no big deal. That's free market. IBM's mind was still on the hardware side. The hardware was paramount even though starting in the late 1970s a lot of people at IBM were saying that software was the future. But, you know, the executives looked at the revenues and saw that the hardware was really bringing more revenue. So, I did that [*EISPACK project*], and then I did some more work on these ideas, and of course I communicated with Jack Dongarra and other people, Jim Demmel and the LAPACK people. I was very much interested in LAPACK. Then I looked at the its design and there were a few things I thought that were not optimal. LAPACK is using

blocking methods, I think that was Jack's, Jack Dongarra's idea, and the way data were being blocked was not optimum for the non symmetric case. I had a conference paper and a technical report that pointed that out, and I had performance measurements to verify it. And there were a few other things, so I communicated all that to Jack and actually those things, at least one of those things, went into LAPACK---the blocking scheme. When the first version of LAPACK came out, I got a commemorative mug and a note from Jack, and also a reference in the preface of the user's manual.

HAIGH: Now through this period you've mentioned your contact with some of the well known people in the area. I think you've also mentioned participating in one of the Michigan summer schools. Were you also active in any of the professional groups, such as ACM SIGNUM?

DUBRULLE: I was a member, I felt that going that track would not give me… I mean the brownie points that I could acquire by doing so would not serve me very well. Because first of all I was really older, even in the 70s, and in the 70s I had not started really doing a lot of publishing except two papers about APL and a paper about virtual storage, my note in *Numerische Mathematik*, and there was a publication co-authored with Jim Wilkinson, but that was not a big record. On top of it I was in organizations that didn't care a wit about that.

HAIGH: Right. So those groups were more for people with academic careers?

DUBRULLE: Yes, yes.

HAIGH: And I think you had mentioned that you did attend John Rice's first mathematical software conference.

DUBRULLE: I attended three mathematical software meetings. And besides that I participated in task forces on the future of mathematical software. One, I think it was in Granby, Colorado, I think it had been called by Wayne Cowell of Argonne. Yeah, I participated in those things and I participated in discussions and it was interesting because that's in those meetings that we saw a lot of new software companies come out such as IMSL, NAG---at the time Brian Ford was the president---and others. I was looking for libraries a while back, I mentioned the PORT library at Bell Labs, and the Boeing library, but there was IMSL and NAG. Yeah, I did those things. I didn't consider them as a service, I think they were exciting forums for ideas.

HAIGH: Alright. So what happened then after the 3090VF version of EISPACK?

DUBRULLE: Okay. So I communicated with LAPACK and did some of that, and continued a little bit on the same ideas for other algorithms. I did also a few odd jobs like looking to arithmetic systems, and designs of small supercomputers that never came to production.

HAIGH: Within IBM?

DUBRULLE: Within IBM, yeah. Everything within IBM. They were confidential projects. Oh, and I tried to reformulate some customer programs that had been written for Cray, and a few other such things. I worked at a problem for the University of Washington, Seattle, in the medical school. It was about digitized tomography and it was to take multiple images of a heart and reconstruct the three dimensional image of the

beating heart. It was an interesting project, actually the mathematics, the numerical solution was not difficult. It took me more time to look at the formulation of the problem that had been done over there than to formulate the numerical solution. So I did odd jobs like this, and then IBM showed these signs of deep illness. IBM had been ailing for some time, but after 1990 it got worse and worse.

HAIGH: Now in the late 80s did IBM retain a significant share of the high end scientific computing market at that point or had it become quite marginal?

DUBRULLE: I think IBM still retained a good part of the market, but there were several problems. People were abandoning mainframes. They were abandoning mainframes for workstations. IBM had the stellar, in my opinion, workstation, the RS/6000, which had been originally designed in 1981, but had not been released because some were afraid that it would advantageously compete with the mainframes. The mainframes were raking in so much money. So finally when IBM released the RS6000 it was after Apollo, it was after Sun, it was after everybody, and there was some catching up to do. But still the mainframes were bringing in good money so the IBM sales division pushed more mainframes, which didn't work. It looked like there was a big disconnection between the top executives and the rest of the company. It looked like they didn't know what was going on in the market. Well, anyway, things went from bad to worse and in 1992 we were in the division called Technical Computing, we in the Scientific Centers, and the president of the division told us that something had to be done about the Scientific Center, that she had too much red ink in her ledger and that the Scientific Centers were part of the red ink.

[Start of Tape 3, Side B]

DUBRULLE: So finally, to make a long story short, the Scientific Centers were closed. At a time I think there were something like six Scientific Centers in the country with various fortunes. At the time everything closed down, there were only two left, which were Palo Alto and Cambridge, Massachusetts. The manager of the center, the old time manager who had been manager for thirty some years, Horace Flatt, a great man, had retired a few years earlier and gone back to Texas where he was from, and had been replaced by Howard Brauer. Howard was a very good guy too, and when the news came that the centers would be closed he put all the resources of the Palo Alto Scientific Center at our disposal to find jobs either in IBM or outside. So, I fired out some resumes to Cray, to a certain number of companies, and, oh, to Thinking Machines [Laughter]. Anyway, I won't say too much about Thinking Machines, and other places.

Well Thinking Machines said I was too expensive, Cray, maybe the same thing, I don't remember what else. Anyway, I was going nowhere. Oh I got a proposal from the Theory Center at Cornell University. The Theory Center was supported by IBM funding. That was a big cut in pay but if I were taking retirement at that time I would have an IBM pension which is nothing really to write home about, but that would have been a good complement. Now, about going to Cornell, I was thinking that if things got worse for IBM and IBM ceased to fund the Theory Center, I'd be stuck in Ithaca, New York. The best thing around was a shotgun manufacturing company, a famous one actually, which was going under. As to the University I had no record of teaching and that would make it very difficult to get into it. So I decided to stay in Palo Alto, and then we got a tip that

HP laboratories, which is the research division across the street from the scientific center, might be interested. So, a few of us at the Palo Alto Scientific Center sent resumes and four of us were hired. Two were hired, I was one of them, to work in a group that was designing a new chip called at the time the Wide Word chip, and it was based on ideas developed at Multiflow and Cydrome. So two of us went to that group which was headed by, guess who, a former IBMer, with whom I'd crossed paths in the days of Future Systems on the task force on floating-point arithmetic. Bill Worley, a super smart guy, was the director of the HP laboratory that was designing the Wide-Word computer. So I got in there, and the Wide Word computer was way way underway at the time. I had a hard time to catch up because I had no idea about what Wide-Word computers were, what the idea was.

HAIGH: Did Wide Word refer to 64 bitness?

DUBRULLE: Yes, but I think it was, I think it was VLIW technology, you know, where most of the work is done by the compiler, essentially. There were things like predicated instructions and speculative execution. I'd never heard of that before I went to HP, so I hit the books and I learned about it and I talked to people to get an general idea. My job there was to design a mathematical package that would efficiently use the Wide Word computer. It turns out that a lot of the work that I'd done for the 3090VF would work very well on the Wide Word chip. Wide Word now is called IA64, and was transferred from HP to Intel.

There was another IBMer there in that group, his name is Peter Markstein. Peter was one of the architects of the RS/6000 working under John Cocke who was the father of the project, and Peter knew about hardware, architecture, compilers, elementary and mathematical functions. He knew everything, and he was the kind of guy who turned ideas into programs in no time, very smart man, older guy. So I started working on that project, and finally I came to the conclusion that the best thing to do for everything concerning matrix computation---which was a big item for Wide Word because it would be a showcase of performance---the best thing to do was to use LAPACK. I'd remove all the stuff that made it portable, even to Cray, simplify a few things, suppress subroutines calls, for example, to elementary matrix operations, and put that in line, and rewrite that for the Wide Word chip and its compiler. I found out that things were, in a way, much less difficult than I thought. I started working on that, and I developed also new algorithms, or new modifications of existing algorithms. I noticed that some of this would work also for the current HP library, you know, for the hierarchical memory. The HP mathematical library was not great, and that would have been a good thing to incorporate these ideas into that library. Well, HP was not impressed. In short, HP didn't care. So it is then that Bill Worley told me that Convex had been contracted by HP to write mathematical libraries, so he suggested that I could transfer my work to Convex. I find the idea cool and I called Convex.

HAIGH: So Convex was producing the mathematical libraries for the then-current generation of HP machines?

DUBRULLE: Convex was still building a library for the current systems, yeah.

HAIGH: I know HP ultimately bought Convex. Were they independent companies at this point?

DUBRULLE: Yes, yes. So everything was doing well, and then when the design of the Wide Word chip was finished the group was practically disbanded. Bill Worley wasn't in charge any more. A new guy came in, a hatchet man, in charge of redistributing the members of the groups. Bill Worley wanted to be involved in the development itself of the chip. Well, the decision was not to do that, that after having done the research, the design would be passed to the development division that will just pass it along to Intel and that was the end of it. So then the new lab director told me that I didn't have a job any more, that HP didn't care about a mathematical library.

So I was looking for a job again, firing out resumes. HP was not interested in numerical analysts, anywhere. I was getting on with Convex extremely well, they were a great bunch of guys, you know, a small company, very aggressive, and really bound together, it was great. When I mentioned that my days with HP were numbered, the manager of the compiler group, and the numerical analysis, mathematical software group, advised me not to leave and to wait because there was something big going on, and that I could still work with them. I could not tell anybody, except that I had found something. I said okay. The big news was that HP was buying Convex. So I kept working for Convex in the same capacity, doing the same kind of work, designing algorithms, running some experiments, starting working also on numerical methods that I had not touched before, like iterative methods for gigantic systems of linear equations. Anyway, there is one thing that made me nervous: there were two possible reasons for HP to buy Convex. One was that HP was interested in the scientific market, really interested, and Convex was a credible entry point. Convex commanded much respect in that market, in supercomputing and scientific computing, and that would have been good for HP. The other reason could be that HP wanted the exclusivity of the Convex designs, and didn't give a hoot about the scientific market or supercomputing. In which case HP would do what it had done with Apollo, what it had done with other companies it had bought: keep them more or less alive for about two years and then dismantle them. Which is what happened to Convex. Convex kept the name for a while. It was the Convex Division of HP, then it became the High Performance Division of HP, and then it became nothing. All the people associated with scientific computing and supercomputing were just told to, practically, take a hike, or put in conditions that they had to take a hike, which they did, and which I did. That was in 1998. I hung it up then. I took retirement, which is good thing to do.

HAIGH: And would you say that the job market for people with experience in mathematical software libraries in general was contracting during the 80s and 90s?

DUBRULLE: Yes, enormously. HP is not a research company, it's an engineering company, there may be a little bit of research going on but it has to do with engineering. And their mode of operation is this: if they need ideas, if they need research and so on, they hire help. If they're interested in Topic A, they look at the list of universities that are working on Topic A, they hire the specialist, or if it's a small company which is good at it, they buy the company, or they buy its principals. HP would maintain groups of people who could move from one subject to the other without going too deep into it, and in a short time could understand what's going on but without being real experts. So, mathematical software, well, leave it to Berkeley, leave it to whatever is competent, and the day there is a need for a library, hire those guys or buy their products or whatever. The job market for such activities is awfully shrinking, so the only forum where

mathematical software is alive and well is the big engineering organizations that solve, for example, enormous systems of partial differential equations.

HAIGH: And so those would be within end user organizations that have a need for that specific code?

DUBRULLE: Yes, and people who need proprietary codes.

HAIGH: And do you think that the success of systems such as LAPACK undercut the perceived need for computer manufacturers to produce their own libraries?

DUBRULLE: Of course, yes. Because, well at worst it gives them good conscience, because they can say "Oh look, there is LAPACK, why should we do anything?"

HAIGH: Right. Well, moving to conclude then. I wonder if I could ask what you think your biggest regret would have been over the course of your career. Is there anything that with hindsight you wish you would have done differently or you think should have come out in a different way?

DUBRULLE: I think what I might have done is shop around in France among the universities to find a good subject for a doctorate, with a good adviser. I don't know if it was quite possible but –

HAIGH: You mean back, back when you were still living in France?

DUBRULLE: Yes. It's not that it would have given me really great additional knowledge but it would have been one item on my resume which would have opened the door to places like research centers here. In the old days things were different. As Wilkinson put it in an interview, he said "I don't have a doctorate, and it's very likely that after the war if I'd returned to Cambridge to the academic world," which was the plan interrupted by the war, "I don't think I would have gone for a doctorate, no need for that, you just go and you research." But, he said, "nowadays people are much more hung up on the credentials and things like this," which is true. And so that would have been in that perspective, and it would have helped.

HAIGH: And reversing. What do you think the single thing that you feel most proud of, that you've achieved during the course of your career would be?

DUBRULLE: Adaptability. And having worked in numerical analysis, in some kind of off-beat paths, except for the Scientific Center in Palo Alto, and the first three years at HP labs. But the rest of bumming around and still working, keeping working on numerical analysis because I liked it, and building my jobs so that I could do that. And actually I didn't put very much importance to that kind of thing, until I got my twenty-fifth anniversary with IBM. There's a little ceremony for twenty-fifth anniversaries. You get a gift and people send you letters. A friend of mine sent me a letter that said "It's amazing that, anywhere you were, you could have kept working on the subject you liked." And I thought, "Well maybe that's true, and it kept me entertained." And I'm happy about it because I saw so many people who went working from to subject to subject to subject to subject, and not in a research position, in development positions, who by the time they were forty-five were intellectually exhausted and couldn't work anymore on what they liked, and didn't do anything of significance anymore. So, who's to complain. My career

path, if I can call that a career, led me here, to a great country, and another citizenship. And that's great.

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

DUBRULLE: You're welcome.

HAIGH: End of Session 2, conclusion of interview.