An interview with
BRIAN FORD, STEVE HAGUE & SVEN HAMMARLING

Conducted by Thomas Haigh
On
29th and 30th June, 2004
In Oxford, United Kingdom

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

© Computer History Museum
Mountain View, California

ABSTRACT

Brian Ford discusses his entire career. Born in Nottingham in 1940, Ford studied mathematics at Imperial College, London. He worked as a school teacher for several years before earning a M.Sc. and Ph.D. from Nottingham University. His work on a QR algorithm brought him into contact with Jim Wilkinson and Leslie Fox. Ford worked with the university's KDF-9 computer center, building a user advice service and numerical software library. In 1970, as the university planned for the arrival of an ICL 1906A, Ford took part in the launch of a collaborative effort with other universities to create a mathematical library for the new machine. This became NAG, initially the Nottingham Algorithms Group but later the Numerical Algorithms Group. Facing lack of institutional support at Nottingham, NAG moved to Oxford in 1973. NAG and the American IMSL were the main vendors of commercial numerical libraries from the mid-1970s on. The library was initially offered in both FORTRAN and Algol versions. In 1976 it was chartered as a not-for-profit company, led by Ford, and it has been financially self sustaining since 1980. Ford discusses the changing institutional form of NAG, the organization and development of its library, its products and marketing, the tools it created to automate aspects of its work, the sources of its code and relationship to its contributors, and its relationship to its customer base. NAG also took part in many collaborative research projects as part of the European Union's ESPRIT and Framework programs. At the time of the interview NAG employed around sixty people, down significantly from a peak during the technology boom years of the late 1990s. Ford retired as director of NAG in July 2004.

Ford was an active member of the international mathematical software community, visiting research groups at IBM and in the National Laboratories and participating in the IFIP 2.5 Working Group and the ACM SIGNUM group. He discusses relationships with many of his colleagues in the mathematical software community, including William Kahan, Lloyd Fosdick, Brian Smith, Shirley Lill, Phyllis Fox, Lee Osterweil, Françoise Chaitin-Chatelin, Christopher Strachey, Bo Einarsson, W.J. Cody, Christian Reinsch, Malcolm Cohen, and Jim Pool.

For part of the interview Ford is joined by Steve Hague, his longtime deputy. They discuss NAG's business strategy and the development of its product line over time. In the 1980s NAG added visualization products through its Graphics Library and acquisition of IRIS Explorer from Silicon Graphics. NAG also distributed GENSTAT, a statistics library, and a range of FORTRAN programming tools and compilers under the NAGWare label. This included the first successful FORTRAN 90 compiler, which Ford credits with saving the language standard from oblivion. NAG also acquired from IBM and marketed for some years a symbolic mathematics packaged called Axiom (based on the well-known Scratchpad system created by Dick Jenks). NAG also collaborates with Maplesoft to integrate its library with Maple.

The transcript also includes a short interview with Sven Hammarling, a longtime NAG employee. Hammarling discusses his own career, his work at NAG, and his involvement with the LAPACK project and the design of the Level 2 and Level 3 BLAS.

[Start of Tape 1, Side A]

THOMAS HAIGH: Brian Ford interviewed by Thomas Haigh on the 29th of June 2004, for the SIAM project on the History of Numerical Analysis and Scientific Computing. This interview is taking place at NAG Headquarters in Oxford, in the United Kingdom, and this is the beginning of the first interview session.

Dr. Ford thank you very much for agreeing to take part in this interview.

BRIAN FORD: I'm delighted to have been invited.

HAIGH: We've already talked a little bit informally about the kinds of things that we'll be covering. I wonder if you could start off talking in general terms about your own early life and education, and the process by which you first became interested in applied science and mathematics.

FORD: I was born in Wilford, Nottingham on September 28, 1940, at the height of the Battle of Britain, and suffered bombing and related strong influences. For example, when I took my sister to see the primary school that I was going to, I watched bombs falling on the factory where my father was building radar. I was four and my sister was two. I went through the state education system in the United Kingdom, went to the local primary school, the school I'd been walking to when I saw the bombers. Then I passed what was then known as the eleven plus exam, which enabled me to go on to secondary education. In Nottingham we had a well known boy's school called the Nottingham Boys' High School, and when I took the eleven plus I was interviewed by the school, I came third in the county in the eleven plus for possibility of going to the Boys' High School. Indeed I was interviewed twice because I was accepted for Dame Agnes scholarship there but was refused on both occasions because I was a Methodist and the places were only available to Anglicans. So I went to a direct grant school in Nottingham where I was pushed forward in doing O-levels in the fourth year rather than in the fifth year of the school. Then I specialized on the science side. At this time I was intent on becoming a strike pilot with the Royal Air Force, and indeed did the examinations, etcetera, to go to Cranwell, but I discovered I was too small to be a strike pilot; I would have to be a bomber pilot. I was unwilling to do this, and so changed my career direction and applied to do mathematics at university. My head master refused to sign my form to go to Cambridge, because by then I'd become interested in the anti-apartheid movement and also CND. He said if I got a major open scholarship I could go up to Cambridge, but in the event I only achieved in the exams for Cambridge an exhibition, so the forms weren't signed and I decided to go to Imperial College, London. At that time I had no desire to ever come to Oxford.

HAIGH: And during your time at school before going to university, had you focused particularly on science?

FORD: Yes, the channeling in the fourth form specialized us in science and indeed I did pure maths, applied maths, further maths, physics and technical drawing at A level. And I also had an active church involvement from being very small, and through this thought quite seriously about becoming a Methodist minister, this is all relevant much later in my life. The upshot was that I went to do mathematics at Imperial College. This is what I could do, this is what interested me. At Imperial, I specialized in applied mathematics rather than having a strong emphasis on pure mathematics. My tutor was [Walter K.] Hayman at Imperial , of memorphic functions . I was taught quantum mechanics by Abdul Salem who was the famous Nobel prize winner for the omega minus particle .

At university I was Secretary to the Union at Imperial College. I just avoided doing what was called the call-up which meant you had to spend two years in the Armed Forces. This had been one of the reasons for the interest in the RAF – I wanted to be an officer and fly rather than pen pusher for that period. That was not an issue, but it meant once we got to Imperial that we were, as a year, four years younger than the year ahead of us and this led to a step function in the life of the college. As Secretary to the Union I was also an active student politically. This was the time of Sharpville, I was involved in the march on South Africa House in London ,in protest. It was also the time when the Campaign for Nuclear Disarmament (CND) was very active and I became passionately involved in CND and had the pleasure of working with Canon John and Diana Collins who led it. And on the Easter March, in 1961, I met Martin Luther King who had just come from his work in Selma, Alabama, and also met Winnie Mandela, Nelson Mandela's wife, so it was a very exciting time. I was involved in the student Methodist Society in the University, so there were these twin tracks, and I was unable to differentiate whether I wanted to continue as a mathematician or to go into the Church. But the Church would not take me until I was 25 , so for four years, having got a reasonable degree, I went and taught in two state grammar schools, in Erith in Kent, and in Worsley Wardley Grammar School up in Lancashire.

HAIGH: So you hadn't at the point considered the possibility of staying on for a master's degree?

FORD: No.

HAIGH: Why was that?

FORD: I was just basically gaining experience of life, thinking in terms of going to the Methodist Church as a minister. And doing this I was offered a place to teach at Manchester Grammar School, as well, but again, decided to stay purposely in the state system. I was still radically involved with anti-apartheid and CND, which took a lot of time. By then I was a Methodist local preacher. I've always led church services since I was 16, until the current time.

HAIGH: And was that kind of involvement with religious and political causes unusual among mathematics students of the period?

FORD: Yes, almost unique, certainly at Imperial College. I had a friend who was a chemist who shared similar interests, but we published a newsletter in the College which actually got into the national press because it was felt to be somehow divergent with the atheist or agnostic view that emanated from Imperial. So I loved the kids while I was teaching but I found the staff petty, and I realized that I was in grave danger of not using my brain and certain native abilities. So whilst teaching in Lancashire, my first wife was doing a sociology degree in the Manchester University at the same time, I approached Robert Sack who was a professor in Salford University about the possibility of doing a master's degree by research with him. All he wanted me to do was to invert a seven by seven matrix, which didn't seem particularly helpful, but this got me started. I realized that I had made a career mistake. I'd been elected a Methodist delegate to the British Council of Churches Youth Conference in England, and the World Council of Churches Youth Conference in Canada, and been on television with well known prelates in religious broadcasts. I realized that I had that social concern, but my ability was such that I should go into the universities or into industry and develop my abilities there as a mathematician, perhaps as a manager, rather than taking the course of going into the church and being a priest. So having made that decision –

HAIGH: Was that because you realized that you'd be a better mathematician than you would be a priest?

FORD: It was because I was interested in serving the community and I was interested in using the abilities I've got to the greatest possible extent. For the community's sake, for my sake, but also as an influence, as a form of service.

HAIGH: So you more came around to the view that in general mathematics would be a more practical way of helping people?

FORD: Yes. As a mathematician in a society in which computers were just beginning to be recognized, I realized that I had a mathematical ability to be useful. On the one hand, the insights for applied mathematics which is taking a problem, expressing it as mathematics and then solving that mathematical model, on the other. Which gifts were, if not unique, certainly more rare. Because of that I determined that what I had to do was to go and continue my education.

HAIGH: And can we get the dates for these things as well?

FORD: By all means. So I taught from September 1962 to July 1964, in Kent, and from September 1964 to July 1966 in Worsley Wardley Grammar School outside Manchester. I met with Robert Sack in the spring of 1966, and by the autumn of 1966 I'd got a place on an M.Sc. course at Nottingham University to do a course in quantum mechanics and computing. From Abdul Salem I'd become fascinated mathematically and intellectually, philosophically, with quantum mechanics, and computing was a vital tool as an applied mathematician that I wanted to develop.

HAIGH: Yes. Now at that time in 1966, when you made the decision to return to study, had you ever used a computer?

FORD: I'd used hand machines at Imperial. We used them every day, every afternoon.

HAIGH: Yes. And desk calculator type machines?

FORD: We used desk calculators as well, so they're the classic hand machines. I was taught by a man called Sidney Michaelson who helped build an early computer at Imperial College and later became well known in the United Kingdom as the Professor of Computing Science at Edinburgh University, and by various other people. I was nearly sent down for having a CND banner on the CND Aldermaston March at Easter 1961 by Professor P.M.S. Blackett (Dean of Science at IC), who also became President of the Royal Society, but that's history now. I'd seen the machine that was being built at Imperial College at that time and I met Bickley, the blind programmer at Imperial who had strong connections with NPL, but at that time didn't forge that link.

So I then went to Nottingham and got a prize for the best M.Sc. there, and this was fascinating because this led me to teach quantum mechanics in the mathematics department and work on my Ph.D. there. Succinctly I started the M.Sc. in September 1966, and finished it in June 1967, and published a paper which involved computing at that time. I now knew Algol 60 fluently, FORTRAN was regarded as somewhat alien. At Nottingham we had a KDF-9 which was based on the design model of the Pilot Ace, initially designed by Turing, but actually built at the National Physical Laboratory by Jim Wilkinson, and other people there, with whom I would have contact later. So I did the M Sc, and I then went to Loughborough University for nine months where I worked with a man called Steven Barnett on control theory algorithms and wrote my first algorithm. Indeed I had the pleasure later on of meeting Olga Taussky at Caltech and JPL for lunch. She actually was kind enough, this was some fifteen years later, to remember that paper and the algorithm which was the first one of its kind she could ever remember seeing.

HAIGH: Now the M.Sc. was a research M.Sc.?

FORD: That's right, by examination, research, and I published a paper from it.

HAIGH: So what did you research?

FORD: What I researched was hydrocarbons in quantum chemistry and the parameterization of the different chemical bonds. Having been to Loughborough, and finding it rather difficult settling with the people there, I then went to my old supervisor George Hall, and George said, "Brian why on earth did you ever leave?" I said, "Well because I didn't think there was a place for me." "But we'd love to have you at Nottingham". So I left Loughborough and went back to Nottingham, and set up a twin activity there with a man called Eric Foxley, who ran the computing center. I started developing the services there, and in the mathematics department I was teaching applied math in quantum mechanics.

So we're now into 1968, and Nottingham University needed a numerical library on the KDF-9, and I collected some algorithms from the National Physical Laboratory and some algorithms from Harwell. I put these together in a coordinated library loosely based on the modified SHARE classification. There was a place in the computing center where people could come and collect a copy of the documentation they wanted for the algorithms they needed. So there was written documentation, there was an example of how each algorithm was used. There was virtually no test software at all, but it was a coordinated collection in available areas: linear algebra, nonlinear optimization, quadrature, approximation, random numbers, that people could have access to.

HAIGH: I have a few follow up questions on those things. When you returned to Nottingham you had mentioned that you were teaching in Mathematics and working in computer center. Were you also working on a Ph.D. at that point or was that later?

FORD: I started my Ph.D. at that time. The Ph.D. always got pushed into a corner for reasons which may be clear already. I've always tried to do too much, and I finished the Ph.D. ultimately in 1973, but that will recur once NAG started.

HAIGH: Alright, so we'll return to that topic later. And I believe from what you've said that it would have been during the time you were working on your master's thesis that you first wrote a computer program?

FORD: The first program I wrote was with Robert Sack in Salford, so that would have been in 1966. That's the first high level language program. Before that I'd tried machine coding but that was basically for fun.

HAIGH: And was that while you were a school teacher or was it earlier during your undergraduate or teaching careers?

FORD: No, that was while I was a school teacher.

HAIGH: Oh, so would you have had access to a computer to run the program on?

FORD: Well this is how I made contact with Robert Sack.

HAIGH: So just for fun at home you had been writing machine language programs and then you wanted to see if they worked?

FORD: That's correct.

HAIGH: I imagine this would have been a slightly unusual hobby, during this period.

FORD: It was actually. My wife, who was a sociologist, thought it very strange. I just needed a different sort of intellectual challenge while I was a school teacher. The rock climbing was one outlet, the activity in the church was always another, but I realized that I was not using my brain, and I wanted to use my brain. And there was a route there too, because from that and the work with Sack, I then tried to write my own QR algorithm. I'd heard about QR and I saw a copy of Francis' paper on the QR algorithm, but it didn't have a code with it so, on the basis of reading that, I started experimenting with a code. And whilst I was doing my M.Sc. at Nottingham I worked with George Hall for fun on trying to develop a QR algorithm as well. He had another student who was doing that full time, but I developed my own too. So reverting again to Nottingham, we now have a situation where Eric (the Director of the Cripps Computing Centre) is using me to build a library for the KDF-9. I persuaded him to start a numerical advisory service and the idea of this was to tell the university that there was someone who could help them formulate their problems to solve them by computing. This was one of the earliest numerical advisory services, I believe, in the world, certainly in Europe. I also built this library in Algol 60 with routines which I got from other places, but then separately prepared documentation , and wrote example programs for them. So people began to get used to the idea of having a collective set of software in library form available for solving their own problems.

HAIGH: Now when you arrived there was there already an informal collection of routines that was available to users?

FORD: A man called Henry Neave had actually developed a random number generator on a paper tape, (we mainly used paper tape on KDF-9). There was a drawer containing about six routines when I arrived, and very quickly I was able to get a library of some 50 to 70 routines in linear algebra, nonlinear optimization, quadrature, interpolation and curve fitting, and random numbers put together so that people in university could use it.

HAIGH: And do you have a sense of whether other computer centers' collections might already have been more extensive, or do you think that this effort to collect useful routines was something that hadn't been done at other places in the UK at this point?

FORD: At that point I was only aware of Harwell and the National Physical Laboratory. The National Physical Laboratory was clearly the national center, and my guess at that time probably the world center, because of the work of Wilkinson and his team. People working in non-linear optimization, people in curve fitting, people working in quadrature, people working in ordinary differential equations. There had obviously been the work at Cambridge on EDSAC 1, but that was very much a Cambridge thing. What was happening with the people at NPL is that they had KDF-9s, as did people in the universities too.

Within the universities there were, I later discovered, a small number of people working in a similar way. They were usually called numerical analysts, some were actually numerical analysts, some like me were more likely numerical mathematicians. But at that instance I was unaware of that and so I built up the activity at Nottingham on KDF-9. As I indicated earlier, I developed my own QR algorithm, and what I wanted to do was to push the service in Nottingham ahead. I wanted to get proper use of the machines and of the resources so I invited two gods to come to Nottingham to speak in their areas of specialization. The first was Jim Wilkinson, who by that time had already written his book *The Algebraic Eigenvalue Problem*, and the second was Leslie Fox. So we're now into 1968. Jim came himself, stayed overnight, gave us two lectures, and I had an unforgettable experience with Jim. This was the beginning of a real friendship which only stopped with his untimely death in the 80s. I showed him my code and he said, "Do you want the good news or the bad news." I said, "Well I'll take the good news." "Well the good news is, Brian the code's not bad! The bad news is that next week I'm publishing my own QR with Christian Reinsch in Numerische Mathematik". Leslie Fox didn't come himself to Nottingham, he sent his colleague David Mayers to lecture on ordinary differential equations. Right at the end of his life, when Leslie was dying, he asked me, "Why was it Brian you liked me, but you loved Jim?" Which of course is an impossible question to answer. But part of it was caught up with the fact that Jim came to Nottingham, but Leslie sent one of his colleagues.

So we now have a situation at Nottingham where we have tens of computer users, using the library software, stopping me everywhere I go in the city. I would go to the theater, The Playhouse and get cornered at the interval to help somebody solve. At the same time, I was working on my Ph.D. which was modeling molecules, and during two periods, from 1968 to 1969, and from 1969 to 1970, I actually worked continuously for several days and nights starting on Boxing Day and working till 2 or 3rd January, running the KDF-9, calculating molecular wave functions, with code that I'd written with a man called John Packer, a program called OPEC, which was an optimization program for molecular wave functions. It was using the code that I'd developed for the library, the eigenvalue code, some non-linear optimization theory, and so on, and the warm spot was near the fixed disc of the KDF-9 in the Machine Room. I used to sleep there, because you obviously have to stay with the machine the whole time.

HAIGH: So I wonder if you can say a little bit about what life was like in the computer center at that point. For example, would users book time on the machine and operate it themselves, or was there a staff that would run the jobs for them and give the results?

FORD: No, there was a staff that ran the computers for the university, and what was unique about Nottingham was that it was a middle rank university, but the users of the computers were across many departments of the university. One of the amusing things is some of the people I worked with in the late 60s and early 70s at Nottingham, have just been given Nobel prizes. So two of my users got Nobel prizes last year and one of them, Clive Grainger from the University of Chicago, was kind enough to get in touch with me when he got his prize for the GARCH algorithms (the other user was Peter Mansfield for work on MRI (and the body scanner)).

So the staff ran the machine and you basically submitted a paper tape which you carefully constructed with splicing, and then you watched while your beloved tape was read into the machine by an operator and then, at the end of the run, they would put the output into the file. That was the traditional way of working, but at that time the importance of the resource wasn't generally recognized. So I discovered the machines were being turned off basically Christmas Eve through till after the New Year. However, because the old religious thing, Christmas Day was special, but Boxing Day wasn't special, so I would then go and sleep with the machine so I could get the runs for my Ph.D. done.

HAIGH: And were the machines usually operated during normal hours, or was there a night shift?

FORD: There was a night shift, the machine was worked 24 hours a day, seven days a week, it was just over the two great holidays, Christmas and Easter, that it was turned off.

HAIGH: And what kind of response time could the user expect between giving in their paper tape and receiving their output?

FORD: It depended on their persistence, if they weren't too troubled they might have to wait two or three days, sometimes a week. There was at least one person who was running a background job who would sometimes wait weeks. But it you really needed your output it was always possible to influence the order in which the computing was run. But I wasn't allowed to run my wave functions during the day because they took too long. I had to run them overnight or at the weekends.

HAIGH: And was there a system to charge academic users for their time on the computer?

FORD: No, everything was free. But there was resource allocation, you had to justify why you needed the time because from early on there was always more demand than there was actual resource.

HAIGH: So there was rationing?

FORD: There was rationing. And this was crucial actually in what happened in the life of NAG once it was formed. I guess we'll talk about that later. So having had the KDF-9, and it being such a brilliant machine, and being used really efficiently, we had the advisory service going and I'd been given staff to help me run the advisory service, got my first two colleagues. We're then told by the Computer Board for Universities and Research Councils, (which is a Government body, part of the Department of Education at that time,) that Nottingham was to be given an ICL 1906A . There was great pleasure in Nottingham at this because the Universities of Oxford, Birmingham, Manchester, Nottingham, Leeds and the Chilton Atlas center were all advised they were getting this wonderful new machine. It was described as being so powerful that it could afford to have an operating system, the George 3 operating system, that took thirty percent of its power because obviously all known problems could be easily solved with the other seventy percent.

HAIGH: And the KDF-9 had operated essentially without an operating system, had it?

FORD: It did have an operating system but nothing as sophisticated as George 3.

HAIGH: It would run only one job at a time?

FORD: Correct. So it was a simple sequential system.

HAIGH: So, by the point you heard this news, was the KDF-9 still considered a reasonably powerful machine, or was it something that you had very much come up against the limits of?

FORD: We were up against the limits of the resource available. We'd liked to have had a second KDF-9, the universities I'd referred to themselves had all got KDF-9's. Nottingham though had jumped ahead in its use of the resource. It appears that our library was somewhat fuller than the libraries available elsewhere. I don't mean that competitively, you know it was just a statement of fact. And of course there was major computing going on in Cambridge, but this was on the EDSAC series, which was all one-offs. And in Manchester, where they had the Atlas system. I'd used Atlas while I was up at Salford writing an Atlas autocode. So we had this group of universities with each university receiving this machine.

HAIGH: And was the reason for everybody getting the same explicitly so that they could share code?

FORD: They were given the same machine because it was a British machine. This was a British Government committee, and it was supporting ICL in the face of competition from IBM and from HP.

HAIGH: So would the 1906A have been first major computer produced after ICL was created?

FORD: Correct. They had the 1900 series, with 1902 and the 1904, but the jewel in the crown was the 1906A , which these universities were going to receive. Basically they were going to develop and polish them up, so that they'd be better machines to sell into business and into industry. So Eric Foxley, who was the director of Nottingham, was told that he's getting an ICL1906A in November 1969. He had me in, in January, and said "Brian I need you to build a numerical algorithms library for the 6A for Nottingham." I said to Eric, "No way. It's boring, I've done one library, I have got to finish my Ph.D., it's supposed to be finished next year, 1970, got a place cleared on a six months postdoc with Per Olav Lowdin in Uppsala in Sweden." I was also doing some work at this time for the Shell Mathematical Education Center in Nottingham, developing teaching material for six formers. I wrote a seminal paper with George Hall in a well known education journal about the way to teach applied mathematics, which led to strong contacts with the people at Caltech and MIT.

So life was pretty full and the thought of building a library was really rather boring, but it's useful. Eric who is a devout Methodist, said "Brian don't you think this is a service you owe the University, and owe everybody," which with my sort of nonconformist conscience is a difficult bone. And in February 1970, I was lying in a bath one Saturday morning in Bunny, in Nottinghamshire, and I realized there was an obvious way to build a library. I'd been to Oxford, whilst working on the KDF-9, and met Linda Hayes, who had developed a library on the KDF-9 with Leslie Fox for the machine at Oxford. That was linear algebra. I'd had contact with Mike Powell because my Ph.D. needed to understand how the Hessian was laid down in a nonlinear optimization program, for my quantum mechanics, and through that met Roger Fletcher and hence Shirley Lill. Shirley and Roger by this time were in Leeds, and the KDF-9 in Leeds, which was in old Methodist chapel, had an excellent optimization library. I'd heard about this awesome woman in Manchester called Joan Walsh, who was working on ODE software and I had a good random number generator which I'd developed myself from the work of Henry Neave, so we had the beginnings, if we worked collaboratively, of a good algorithms library.

HAIGH: All these things were written in Algol were they?

FORD: They were written in Algol. There was no FORTRAN at all. But, many of the scientists and engineers we were working with were working mostly in FORTRAN. At this time I'd been working with something like 15 or 20 professors in the university at Nottingham, in research areas, in partial differential equations, in civil, mechanical, whatever, engineering, and the physicists, people working on the NMR, were at Nottingham at that time. Three or four of them were there, three later moved to Oxford, and they were working in FORTRAN, simply because we were getting codes from the States. So there were those tensions coming in, and mechanizing, building the library collaboratively was a very sensible way forward, we –

HAIGH: I'll turn the tape over.

[End of Tape 1, Side A] [Start of Tape 1, Side B]

FORD: So it's now the thirteenth of May 1970, and we're in Nottingham University and I've invited Joan Walsh from Manchester, Shirley Lill from Leeds, Linda Hayes from Oxford. And we hold a first meeting about the idea of building a collaborative library, and I put a proposal to the meeting, Eric chairs it, which is as follows: that we develop a numerical algorithm library in FORTRAN and Algol 60 covering the whole of numerical mathematics and statistics; that the documentation will be as important as the software for the library; that there will be an example program which shows the use of the routine which will be available to the users; that there will be stringent test software, which is the basis on which the algorithm is selected for inclusion in the library; and that we share out responsibility for the library amongst us upon the basis of our areas of numerical expertise. The original reference papers for that discussion are in this black file.

HAIGH: Now had this meeting been called specifically to discuss the creation of the library like this or, was it a more general meeting that people were attending just to discuss the imminent arrival of these new machines?

FORD: No, it was library, it was a meeting that I called, with Eric's encouragement, specifically to create a library for the ICL1906A. The document here, I'll give you a copy, actually defines the work of the early days and the decisions we took. So the important thing is that the source code for the algorithms was only *as* important as the documentation, so that the users could use it. And the example program was basically something that if they wanted to they could develop to use as the basis for their own program. The stringent testing was vitally important because right from the beginning we were going for the best algorithms in the world. We were working with the crème de la crème. Wilkinson, and the Eigenvalue problem at that time, had no comparator on earth except possibly Velvel Kahan, at that time at Toronto, soon to be in Berkeley (due to a reference from Jim). Mike Powell, Roger Fletcher, famous for their work on nonlinear optimization, and so on. We knew that this thing could only be of value if the algorithms were very carefully selected against the criteria we defined, and then met the highest possible standards, because otherwise it would be useless. But we basically set out to build a library for the 6A. What we finished up doing of course was developing, over a period of time, the first truly portable numerical algorithms library that went to two or three hundred different platforms in its onward development. But we didn't do it for that reason, we did purely for our people on the 6A in the university.

HAIGH: Yes. And looking at the 1972 paper, "Developing A Numerical Algorithm Library," published in the *IMA Bulletin*, I do notice that these, I think essentially these same desirable characteristics for the library that you just mentioned are spelled out, and portability is noticeable by its absence.

FORD: Correct.

HAIGH: So what was it about taking this approach to a library that turned out to make it portable, or was it necessary to add portability later?

FORD: We built characteristics into the library from the beginning, in the way we coded, that actually made portability possible later. At that time, of course, nobody had thought about the issue of portability in any depth, Morven Gentleman used the term at a NATO meeting, I believe it was 1968, but this was just a vague concept at that time. We're in 1970, we're thought to be unique in suggesting a library in both FORTRAN and in Algol 60. The library is being developed within UK universities. The first appointment to NAG was made in September 1971, of my deputy Steve Hague, who I'm lucky enough to still work with, and a coordinating group developed at Nottingham, for the Library. The group were kind enough to call it the Nottingham Algorithms Group, because it was started in Nottingham, and because I was a fierce lover of my hometown, and of Nottingham Forest Football Club.

HAIGH: So did NAG at this point have any kind of legal existence or was it just an informal –

FORD: It was an informal voluntary collaboration. All of us greatly admired Jim Wilkinson, and Jim of course made his codes available to us, the Algol 60 code which was in *Numerische Mathematik*. Our first major code activity was to start translation of the numerical linear algebra into FORTRAN. So when we met in May, on the thirteenth of May 1970, we agreed that our first library would be available by the end of September 1971, and what that involved was the translation of much code from Algol 60 into FORTRAN. Although there were other numerical areas where we did in fact have FORTRAN and had to create the Algol 60. All of this was written to strict language standards, and again, we recognized the importance of this from the beginning. The important thing, for me, was that in the last week of September 1971, one of my colleagues, from Nottingham, whose name escapes me at the moment, and I jumped in a car with fourteen boxes of cards and came down to Oxford and worked with the ICL 1906A there for three days while we created the first NAG Library.

The first NAG library was successfully created on the thirtieth of September of 1971, so we were on time, which was also very important. That library for the 6A would also run on the 1900 series, we discovered. Steve was appointed the coordinator for the 6A library, so another university, the University of Loughborough, appointed another person, Richard Tallett, to do the library work on the 1900. We then came up against two machines, the System 4, and IBM machine in Cambridge. Cambridge had stopped producing their own machines, and had been persuaded or had decided that they clearly wouldn't have an Atlas, because that had been invented in Manchester and there was great competition between Manchester and Cambridge. So they took an IBM system. There was a need for a numerical algorithms library for both the IBM system and also for the System 4s. The System 4s were an ICL machine, which had badly truncated arithmetic. The truncation mechanism on the IBM system in Cambridge had been improved because of the work of Vel Kahan and colleagues in the University of Toronto, as we learned later. So in the summer of 1971, we have this discussion going on about whether the library can also be moved into the byte environments. Rather than the two bit arithmetic of the 1900 series, we have the byte words with the IBM system with different numerical characteristics.

At this opportune moment there is an IFIP Congress, in Lubljana, and at that meeting I was invited to give a presentation by Jim Wilkinson about the NAG library. Whilst I was giving, leading this evening session Vel Kahan came in and said "you don't know anything about libraries". Jim said, "Vel, listen to him.This guy's the man who's developed the NAG library in England I've told you about." George Forsythe was there, and George said "Listen to him Vel, I've talked to him about this and it's surprising how far they've got, particularly as none of these are numerical analysts that we both know." So Vel said, "Alright, I'll sit and listen but I'll interrupt the first time you start talking nonsense." Well he let me finish, and the reason I mention this meeting is not just to have met these two wonderful characters, George Forsythe who sadly soon afterwards died of liver cancer, and Vel Kahan of course we're glad to be still working in Berkeley, but it was also the beginning of the IFIP Working Group on Numerical Software. George Forsythe was one of the first professors of Computer Science in the world. His chair was at Stanford.

From that meeting Cleve and I went to Technical Policy Committee Two and suggested the formation of the IFIP Working Group. Now I should at that point leave that element of the topic and come back to it later. Bo Einarsson's involvement in the meetings and everything, we'll come to that. Back in England we were making this decision about the byte machines. I think we're now into 1972. I also at this time got an invitation, from Wilkinson, to go with him to Ann Arbor, Michigan, as his baggage porter, and then to go on and work with him at Argonne National Laboratory on a project called EISPACK. We had translated Wilkinson's numerical linear algebra into FORTRAN. The idea of EISPACK had come from a man called Jim Pool, the workers on EISPACK were Burt Garbow, Wayne Cowell, Brian Smith, and obviously a number of other people, including Virginia Klema. So in the first week I went to Ann Arbor, Michigan, and the first night I sat up developing numerical examples to break the cxinvit code of Jim Wilkinson, and next to me was Brian Smith doing the same. Obviously Brian was working on EISPACK (Brian was one of the speakers at my celebration meeting last week on my retirement from NAG. Jim Pool was also at the meeting, I'm glad to say).

We formed a strong bond, and the second week I went with them to Argonne National Laboratory as a consultant on the EISPACK activity. The idea was to get us to use the same FORTRAN code. Their FORTRAN code was probably better than ours, but for a period we continued to use our own. Much later, when LAPACK was developed, which involved people directly from NAG in its design and construction, their code and our code ultimately fed into the same source. We had wonderful collaboration with Argonne from that time in May 1971, right forward until the current day, that still continues although it's now in different generations. We were working on the NAG library quietly in the background, through 1971, through 1972. In 1972 we faced the issue of the different arithmetics and how one would handle machine characteristics. So the work started to define a model of machine arithmetic that could be encoded into numerical algorithms so that you could have the same algorithms in a high level language like FORTAN II or FORTAN IV, and Algol 60, which would work on byte and bit machines. I worked very hard on parameterization of a machine model. This, for me, was a piece of simple applied mathematics. It was a question of how you could characterize it, but it's also a matter of how you could parameterize that. At this time Kahan, for example, said it was absolutely impossible. That's important because later on I'll come back to Kahan's great leap of faith when he said at a meeting in Oak Brook, in 1975, "Well of course Brian we're all working on portable software."

So we started activities to put the NAG library onto byte machines. Peter Kemp was one of the leading workers in that area, and by this time in Nottingham, in the late 1972, we had some twelve people working full time on building the libraries.

HAIGH: So let's talk a little bit more about the very first versions of the library then. So you came out to this initial meeting with the idea that they should be something that it could be called the Nottingham Algorithms Group, and it would be –

FORD: In the first meeting we didn't think about names at all. What we concentrated on was what the characteristics of the library should be and how we could pursue them with quality, the name was something that was given to us in the third or fourth meeting by the other people.

HAIGH: And were meetings happening on a very regular basis?

FORD: Yes, we started meeting once a month, usually in Nottingham, because clearly it's in the center of England. So people could come down from Manchester or up from Oxford, and we actively worked on the issues of building a library. For example, in the folder you have there, you'll see there's a paper that describes the documentation, so that all of the routines were documented in exactly the same way. One of the big issues we had was the naming of the routines. Over the years there have been many jokes, some in good taste, some in bad taste, about the naming of our routines. It was certainly one of the hardest fought issues for the early collaboration. My concept was that the first three characters would be the modified SHARE classification index number, so for example FO4 was solution of linear equations, FO2 was the Eigenvalue problem, the S chapter was taken up with the special functions we choose to include. Fortunately they didn't go beyond a hundred so we can have S and the number, again, from the modified SHARE classification index. Where the index was undefined or unclear, for example, in the statistics area, we created our own having, carefully checked with the SHARE organization, which by this time, already in the 70's, was not strong. We found that we had to preserve the modified SHARE classification index very strongly. Later on it became an issue in the 80s, but put that to one side for now.

HAIGH: And beyond the classification system, had the SHARE library exerted any other kind of influence on the way that you had approached this?

FORD: Well we knew of the SHARE library, and the CACM algorithms, but they were so spotty. You never knew whether you were getting the work of pure genius as with some of the special functions written by Cody and Kuki, which were world class, or a curve fitting algorithm with no mathematics behind it at all, so the simplest difficult cases would trip it up. So we knew of those spotty collections, and we recognized there had been little discipline exercised in what was included, and they weren't uniformly presented. Because of FORTRAN our names had to have six characters: the first three would be the modified SHARE classification index entry, the fourth and fifth allowed a substructure, so taking the fourth character for A it could be positive definite, symmetric positive definite matrix, for B it would be symmetric, for C it would be general. Then the A, B, C, D, E, would be the running ordering of the algorithms within that subchapter, and the final letter, the A or the F, would tell the user whether it was in Algol or in FORTRAN. So SO2 special function, B the particular set of complex data rather than real data, C it's the third algorithm in that chapter, F the language it was to be called from.

Now clearly we also chose every algorithm, so we tried to have a one-to-one mapping from the problem area of the numerical mathematics or statistics onto the algorithm selection. In some areas that was easy. By that time already in the numerical linear algebra, for both the eigenvalue problem and linear equation solution, it was possible to put in just one algorithm for each problem type. In other areas like optimization theory, the E04 chapter, it was very spotty coverage. Nobody knew what the best algorithms were. The simplex method had been developed by John Nelder (and Mead,who worked with him as a statistician, both) from Rothamsted. Their algorithm was good for some very small problems and then for the bigger problems you got to the Davidon and Fletcher-Powell methods, and of course we were working with Bill Davidon in the states, as well as Fletcher and Powell in the UK. By this time, because of Jim's influence, the best numerical analysts all over the world were beginning to channel to us their latest work, using us as a means of publication within the UK community. By 1973 that collaboration , a great characteristic of the NAG Project was already very active, and over the years we've been lucky enough to work with the best numerical analysts and computational statisticians from all over the world. So there were chapters where the coverage was spotty and you basically selected a few good algorithms and tried to advise people of the problem types for which they were best.

HAIGH: And the people coming to these meetings, were they expert creators of mathematical functions or were they directors of computer centers?

FORD: By '73, we had different sorts of meetings. We had a core meeting of the leaders of the project, who were Joan, Linda, Shirley and me. We had no great concern about our careers. We weren't looking to become the great and the good, we weren't developing research departments or anything, we were basically building the library because we realized that it was a valuable and worthwhile thing to do. Then we had subject committees. We had one in numerical linear algebra which used to regularly meet at the NPL so Jim Wilkinson could be present, we had one in optimization theory, that moved between Harwell and Leeds. The ODEs were done up in Manchester, Joan Walsh, Ian Gladwell, other people. Individuals were assigned responsibility for particular chapters in the library. But we also had the beginnings of what we called implementation groups, who would take the library and get it working effectively on their own particular machine family. You'll remember that by this time we have an IBM library being developed from the University of Cambridge, the University of Newcastle, and the University of St. Andrews. We had a System 4 library involving the University of Bristol, the University of Birmingham, the University of Edinburgh, and so on. Hence we had these different focused technical activities. The directors of the computing centers were off doing their own thing, thinking about developing their centers. We were totally focused on developing a numerical algorithms library and encouraging its general use within our universities. So of course while this life was going on I was still teaching in the maths department and still looking after the users in Nottingham.

HAIGH: So unlike SHARE which was quite formal in its insistence that each member was representing one particular computer installation, this was much more an organization of individuals with related interests?

FORD: Correct. Clearly the people building the 6A library had responsibility in the 6A centers, and the people on the different machine ranges were actually preparing things for their own specific use in their own university. But the numerical analysts who were helping us, and the computational statisticians, were basically seeing us as a publication route to get their software used. Because of the likes of Wilkinson and Reinsch and Fox and Powell and Fletcher, people wanted to see if our algorithms were better than their algorithms, and so there was a healthy competition.

HAIGH: So in that sense NAG would fulfill the same kind of role as publishing something in the algorithm's section of ACM Communications or the Transactions on Mathematical Software?

FORD: Correct. Obviously TOMS had not yet come into existence, but certainly that would be the case. So we're building the library, we'd set up the collaboration with the States, we've got the beginnings of the discussion for the formation of the IFIP Working Group, and, suddenly, Nottingham University decided that we couldn't stay there.

HAIGH: Now you had mentioned that a significant number of staff were working exclusively on the library by that point. Was Nottingham paying for them?

FORD: These were all being paid for by the University with grant money from the Computer Board for Universities and Research Councils. "The Board" were desperately trying to get British universities to cooperate with one another, and saw our collaboration as the goose laying golden eggs.

HAIGH: And this was the same group that was providing people with the 1906As?

FORD: That's right.

HAIGH: So they would have an interest in having them used.

FORD: They were also funding the other machines, the System 4s and the IBMs. They were buying the computers for all UK universities.

HAIGH: So at that point no universities were buying their own computers?

FORD: No. I think Manchester was still building their own at that time. I think at that time Atlas was still being developed, but that was the last machine of its kind. So we have this galvanizing experience of being told by Nottingham University that NAG has to stop. Eric Foxley was sacked because of the poor performance of the 6A, nothing to do with numerical mathematics, just general poor performance on site. The Deputy Vice Chancellor died of a heart attack whilst talking about the NAG project, which increased the ante greatly. My baby son was born, was critically ill, but fortunately fought through and survived. But his mother… was restive. And then Nottingham says we have to leave, and, quote "if you stop this stupid NAG thing and go back to mathematics we will not destroy your career."

HAIGH: And who is that quoting?

FORD: That's quoting the Deputy Vice Chancellor and the Deputy Bursar of Nottingham University. They were sent down by the Vice Chancellor and by the Bursar, to "sort me out", and this is in December 1972. But we discovered that two universities would welcome us if Nottingham formally say that we can leave. One is the University of Oxford and the other is the University of Cambridge. And I managed to get the chairman of the Cripps Computing Center Committee to sign a piece of paper saying that we could leave the university, and took it down to Oxford. Oxford then welcomed us with open arms. Then Nottingham say it's been a misunderstanding, and they'd not intended this at all, but that was too late. So we moved to Oxford on the 1st August 1973.

HAIGH: So given that it wasn't costing them anything and it was bringing more people, more positions to the university, what was it that Nottingham didn't like about NAG?

FORD: They didn't like the fact that we were using significant machine resources, they didn't like the fact that I was clearly developing my own empire there, and they didn't like the fact that they hadn't formally agreed to any of this, and yet the Computer Board was telling them what a wonderful job we were doing. So we moved to Oxford on the 1st August, 1973, and in the event only three of us moved, and the third of the three went back to Nottingham within three months because he found the women in Nottingham much more attractive and much more available.

So Steve and I continued to run NAG from Oxford, from that time forward. Again we started to appoint people here, a man called Alan Scott, F. A. Scott, was the director for computing service, and once he'd understood that I wasn't working for him, I was working with him, life was very good. The Computing Lab was run by Leslie Fox, but Leslie never liked the computer, he only believed in hand machines, but again, he was benign and helpful, and the group developed in Oxford progressively so that by the end of 1973 the library was in use on most computers in the vast majority of British universities. And there was growing pressure from universities in Europe and large UK companies, like Rolls Royce and ICI, to have access to the library. So NAG was set up as a not-for-profit company, limited by guarantee, associated with, but not part of, the University of Oxford, in May 1976.

We became financially self funding on the 1st August 1980, and have been an independent organization within the ambience of Oxford University ever since. Once we got to Oxford, the pressure to make the library portable got greater and greater. In 1972, and in the middle 70s, there were two mathematical software meetings in Europe, similar to the meetings that John Rice organized in the States. I and my colleagues working in the project didn't have time to organize the conferences so a man called David Evans organized one for us in Loughborough, and another person, David Jacobs, organized an IMA meeting in Brighton, and the proceedings of those meetings are here. I went to the second Rice meeting, (I didn't go to the first which was purely North American, the second Rice meeting allowed some European influence) and in a dark corner I was handed, by Lloyd Fosdick, who was the chairman of the computing science at Boulder, Colorado, a code, Prettifier, Pretty Printer, in plain white envelope because neither of us were sure of the legal status of the code, (since clearly I was going to take it out of the States and bring is back to the UK). This I did, and that was our first software tool. Our second software tool was the master library file system, and the idea of the master library file system was to collect together in one place the variant copies of the codes of the library. We had what we called a contributed library, which was the original version of the library codes, and that formed the base of the data base within the master library file system. And then we had a means of putting into the data base the alternative records of code that were required for different computing systems, the intention being that by looking at the code we had a single source, so if an error was found in one library version we could make sure that that error was correctly identified, and then the correction to the library could be made in all its versions.

HAIGH: So this system wouldn't automatically generate different versions of the library but it would check them against each other, is that correct?

FORD: Once the data base had been formed we did generate the code for each library out from it. But initially laying it down it helped us to identify where change had been felt necessary in the library codes for different systems, sometimes because of infelicities in the dialects of FORTRAN accepted by the compilers, but more looking at this parameritization issue, of convergence criteria, of the impact of arithmetic on various aspects of the calculation, etcetra.

[End of Tape 1, Side B. Start of Tape 2, Side A]

HAIGH: Now to bring you back slightly to NAG's internal structure. You had said that early on Joan Walsh, Linda Hayes, and Shirley Lill had been the other three members of the four person group that was leading NAG. So if you could talk a little bit more about what that central group was doing and where the other people on it came from.

FORD: I saw myself as the coordinator of the group. Each of them had a specific numerical mathematics interest, and so they were very much involved in their own subject groups. I was obviously running the implementation groups. The core activity was being driven forward by the four of us. Joan had become a member of the Computer Board, by this time she was a full professor, and Richard Field, who's now the chairman of NAG, was actually also one of the civil servants on the Computer Board responsible for funding the NAG activity, so we were very comfortably placed with regard to one another. The Computer Board was encouraging us to drive forward, they hoped to use us as an example of how collaboration could work in other facets of university computing life. Indeed, in 1973 the US NSF paid for Joan, Shirley, Linda, and I, together with Eric Foxley and Steve Hague, to go to the NSF for a week so they could try and understand how and why we got the collaboration to work, whereas the NATS activity from Argonne, (from which EISPACK and the other PACKs came,) had not been able to create the same collaborative creative framework. Our purpose was still to develop a high quality numerical algorithms library, and Joan, Linda and Shirley still had their other jobs and responsibilities. I finished my Ph.D. in 1973, and by the time NAG moved to Oxford I'd decided that I was not a first rank applied mathematician who was going to do world class research. The Shell Center wanted me to go and work there. I did some useful work for them, but that wasn't my interest. The numerical algorithms library we were working on, and the ability to meet people like Wilkinson and Reinsch and Fox, the people at Argonne, the people at Stanford, the people at Berkeley, Marchuk who had become minister of science in the Soviet Union and wrote an ODE solver for us, this was a fascinating community and so I decided that I'd make this, at least for the time being, a career activity.

HAIGH: Were Hayes and Lill also full time faculty members?

FORD: Shirley Lill taught optimization theory in Leeds. While she was there she burnt out her first husband. Ten years later with her first round of companies outside the university she burnt out her second husband and moved on to her third. Shirley now is an extremely prosperous owner of several companies, still a close friend, and particularly of Carol and Brian Smith who've just been there. They just built stables for thirty horses, their daughter is a three day event rider.

HAIGH: And were the companies directly related to applied mathematics?

FORD: No, all her companies were involved in computing. So she did the Littlewoods catalog system for the Moores family. When she sold out, for two million pounds, about ten years ago, she then set up another company associated with the use of address codes, and so on. Linda Hayes remained the research associate and research assistant of Leslie Fox until he retired from the Computing Lab, and sadly died. Linda, of course, played bridge for England and is also a high quality county tennis player. Joan is a devout Roman Catholic, she's one of the main forces behind the continued availability of the Tridentine Mass in parts of the Roman Catholic Church in England. She reads and speaks fluently in Latin. Joan advised the Papal Nuncio Bishop Heim that Cardinal Wojtyla of Cracow should become the current pope. Joan moves in those sorts of church circles. Joan retired from Manchester University as its Pro Vice Chancellor . She was never appointed a vice chancellor, I think simply because of male prejudice. She was a little too early. So each of them has been very successful in their own way. They kept an interest in NAG: Shirley until the early 80s, Linda until the late 80s. Joan was the first chairman of NAG and indeed was kind enough to make a few comments at a celebratory meeting I had last week as I retired from NAG.

HAIGH: So an interesting group. Now would it be exceptional within applied mathematics at this point that out of the four of you there would be three women?

FORD: Absolutely, and I think the reason for that is not my particular feminine characteristics, but because none of us were actually concerned about building academic careers. All of us were interested in what we were doing, and were all willing to make the time and had the energy to have our own other commitments but still build the library as well.

HAIGH: So you feel at that point someone who was looking to make a respectable academic name for themselves would not have gone into the area of software?

FORD: I think that was true then. I think it's true now. There are very, very few people who have made such an outstanding contribution through software that they've become internationally recognized. I guess an exception might be Stuart Feldman, with the Make command, the writing of the first FORTRAN 77 compiler. I don't believe Stu has published any outstanding seminal papers in computing science, I think he's just immensely knowledgeable, and of course he's one of the creators of the C language, and has made other contributions to the Unix operating system.

But I think academic numerical analysts in the main have had to specialize in the mathematical infrastructure of their numerical analysis, and the technical mathematical issues involved there. Rather than their core focus being the creation of world class algorithms. Some of them have created world class algorithms, Linda Petzold, for example, and Bill Gear, Cleve Moler developed the excellent QZ algorithm, but whether Cleve would consider he followed an academic career or not I wouldn't like to comment. Vel Kahan has been able to spend much of his life working in algorithms, but also in the deep mathematics of the various areas of numerical analysis he's been interested in too.

We were all very competent mathematicians but basically one of our drives was to get the algorithms to be used by the user community and to present them in a form that they could readily employ them. And the issue of portability, which is one of the most fascinating, came out of our experience of going multimachine, multiplatform. The seminal meeting about portability was held in Oak Brook in 1975, and after that meeting came two major contributions which I'm sure our world has now clearly recognized. On the one hand there was portable software, the model that derived from the work on the PORT library and NAG's own work on its master library file system, and my work on the parameterization of machines, leading the activity within IFIP Working Group 2.5. The second major contribution that came to world computing out of the '75 meeting, was the recognition by Cody, and others, of the value of having a chip with perfect arithmetic, that we would all employ. So one solution of the portability issue was the software solution in the portable numerical algorithm library, and the other one was the creation of an underlying arithmetic that everybody would use. (It's worth saying, in 1975 there was not only the portability meeting, there was the second meeting of the IFIP Working Group 2.5 on Numerical Software in Argonne at that time, the first meeting having taken place in Oxford the year earlier).

HAIGH: And that would be what eventually led to the IEEE floating point standards?

FORD: Correct. That all came out of that 1975 meeting.

HAIGH: Yes. So following up a little bit on portability then. You originally had the library available in both FORTRAN and Algol, so that was a kind of manual portability. Was it difficult to make these translations between the two high level languages?

FORD: It wasn't difficult, but it was a lot of hard work. Of course we had to do all of it by hand. I mentioned the early software tools, and we had the Pretty Printer. If you knew exactly how your code was laid out you could use an editor for making automatic changes, so that was important. On the other hand we had the master library file system, which I've already talked about. We then realized that if we used a portable subset FORTRAN we could greatly aid portability in the FORTRAN community, which was by now the dominant language for science and engineering. So from our experience with the master library file system we actually put together a subset of FORTRAN which we knew would work on the different machine families to which we moved the library.

HAIGH: So would this subset help with these issues about bits versus bytes and precision and truncation and things?

FORD: Yes. What had happened was that we had a clear model of a computer platform, and we built into that model parameters concerned with word length, concerned with convergence criteria. We then embedded that into this subset FORTRAN.

HAIGH: So the subset then had more to do with the capabilities of the compiler than with the architecture that it's running on?

FORD: Yes. The architecture was handled by the parameterization, and the language issues was handled by the language dialect.

HAIGH: So when you add those two together, you've got portability?

FORD: You have the basis, the roots of portability. It was a real excitement for us. We put together our own portable FORTRAN, we then made contact with Phyllis Fox, Stu, Norm Schryer, W.S. Brown, and the people at AT&T, Murray Hill, New Jersey, and we discovered they'd got a PFORT verifier, but the difference between what we'd come up with and what they'd chosen was only one characteristic. Which we found fascinating, and of course what had happened was the people at Bell Labs had created a software tool for the automatic checking of this FORTRAN subset. There was a friendly battle between Europe and North America, the modeling by Phyllis and the people at Bell, was just so much better than I'd been able to manage working with my own parameterization scheme for the IFIP Working Group, but this was balanced by the far greater practical experience we had enjoyed developing the library. So what happened essentially was that we had a melding together of the two bodies of experience. We had characteristics that they didn't have, so we enlarged their model, and their parameterization, and with the enlarged model with its parameterization we were able to create an effective model of computer arithmetic embedded within the enlarged FORTRAN subset.

HAIGH: What date would that have been?

FORD: The late 70s, about 1977, '78. And we then had a period when we could genuinely write portable code. We could sit here at Oxford and prepare the source text of a library that we knew with systematic parameter settings would compile and run reliably on tens of different platforms and we still had access to computers all over the world on which to test the compiled libraries. For example, in 1978 I modelled the arithmetic of the BESM 6 for the Soviet Academy of Sciences at the height of the cold war, at Akademgorodok, Novosibirsk. The machine had a wrap around arithmetic which was very unclean, but we were able to, nonetheless, use the same portability model for a library for the BESM 6. And for a period of years, until the workstations came, we had a near perfect, portability mode on sequential FORTRAN engines.

HAIGH: And were there similar issues with the Algol 60 compilers?

FORD: Regarding the Algol compilers, in the Algol 60 standard, there was no specification of input or output, so it was a nightmare, and our Algol 60 library stopped at Mark 8, in 1981. We then also started an Algol 68 library. Algol 68 was perhaps the best language I ever worked in. We did a Pascal version of the library too. I was present at the meeting in Southern Germany, at which Nicholas Wirth, the inventor of Pascal gave his first description of the language. Sadly, whilst good for teaching, it never really took off. I also spent a number of years building an ADA library, leading activity in Europe for scientific ADA funded by the European Union through its Esprit Programme. But again ADA was never adopted by the universities. The next library we worked on was the C library, and C derivatives I think could yet be seen to dominant the scientific computing scene. FORTRAN is very important and the algorithmic cores, our components in FORTRAN, are the best numerical engines in the world. But they can be set in a multi-language environment where you use material developed in different programming languages, setting each part in the language best for that function. Such programs can really worked very well. But the real problem has always been the documentation and it is only recently with the coming of XML that we've been able to create an algorithmic core expressed in attenuated FORTRAN, supported by XML documentation, also using various scripting languages, so that we can now automatically generate wrappers which enable the library cores to be used in any language and automatically create the documentation to support that software, and automatically to create the test suites needed for example programs to go in the documentation and also for the stringent testing of the software. It's only in the last year, 18 months, that that's become possible.

HAIGH: Okay, we'll return to that topic later then. When was it that the last sites would have given up on using the Algol 60 version of the library?

FORD: There were Government laboratories in England using Algol 60 until 1997. There are strike aircraft still being flown in Europe which use our Algol 68 codes, they'll do that until the year 2007, and there are missiles that use the algorithms in Algol 68 as well. The ADA code is still being used in many guidance systems as well, and similar sorts of control environments. One of the things that Joan and Shirley and Linda and I realized, was that the algorithms we were developing could be used for both the efficient delivery of food in the third world, and the efficient delivery of missiles in Iraq. And, you know, we couldn't hope to influence them being used in one environment and not the other. I think experiences we had during the cold war, in any case, caused us to realize that the Western democracies had to take care of themselves. One of the most exciting phone calls I've ever received was when a missile went into the command and control center in Baghdad during the first Iraq war. Some year earlier I'd been asked how accurately an algorithm we'd developed could locate over a thousand miles. We computed it was into an area sixteen feet by four, and we were told it would have to go into an area six by two, but we were fortunate and the missile went precisely down the flue of the command and control center. That was because of work that had been done in developing quality algorithms, and on the IEEE chip, so that one could achieve that sort of precision of modeling and of control.

HAIGH: So would it be typical that you would have that kind of close relationship with end users?

FORD: Yes. We've been fortunate right from the beginning with our user support and numerical advisory desks. We've always had contacts with major research groups often because they were looking for specific algorithms to solve problems in particular areas. So I believe there is a NAG Ridge in Antarctica, and there is a mountain range (or area) on Venus that is named after our activity as well, and these are given by people who have used our software in their specifications and applications and come to us for help. And indeed, I hope without being too self serving, this is why the Queen gave me the OBE for services to science and technology because we had been able, it was felt by the Ministry of Defense, and by the Department of Trade and Industry, to have a profound influence on the solving of research and development problems. Gaining a real insight into other people's problems is the best means of developing algorithms for solving problems of that kind, and for being able to persuade users that the quality that we sought had been achieved as far as it was possible. This has meant that over a period of 34 years, algorithm developers still send us codes to include in our libraries and users with esoteric problems, the cosmologists from Cambridge, for example, Steve Hawking and his group use our software.

HAIGH: Now you had mentioned earlier that back at Nottingham before the library was even created you had been involved in establishing user service where you would assist people with their problems. So what did you learn about ordinary users and their relationship with the programs that they needed to create to solve their problems during this period?

FORD: I was encouraged to do this by Jim Wilkinson. Jim said, Brian, if you want to learn about numerical analysis, the best way is to solve other people's problems. And what I learned was that, particularly in those early days, it was best to sit down and talk with them about their problems, rather than trying to solve the mathematical problem that they'd brought to you. Because there were often different ways of formulating problems. It could be as wide as you starting with a numerical linear algebra eigenvalue problem, but you actually discover what they really want is to solve stiff ordinary differential equations, and so on. Particularly if you get into areas like optimization theory, problem formulation, to try and keep the constraints as simple as possible is actually very important. So they come to you with their problem and you together agree on a mathematical model. Then you characterize the model as being in one of the many areas covered by the library, and decide which algorithm or algorithms within the library are best suited to solve that model. Sometimes you particularize a general algorithm you have in the library to solve their specific problem.

Most of them really didn't want to program. So you're much more likely to be talking to a Ph.D. student than one of the senior professors. Some of them were wonderful programmers, extremely clean, recognized the value of high quality standards, using language subsets, using our computer model for their own codes, so they had portability across the large systems that they were working on. Others brought really filthy code, where it was very difficult to discover what was poor programming and what was actual algorithm. The main characteristic was that they're all impatient, that very, very few of them had time to breathe. Particularly at that time, university research was very time intensive. There is a lovely story told of the time when the University of Bristol had its machines turned off for three months because of a major flaw in the electrical system. A study was done on the quality of the research from the university over a two year period, and it was found that the best research was completed and later published soon after the machines were off, because during that time people would actually start to think about the problems, rather than just generating more numbers. And this is something that we've always had to watch, to try to get people to think hard about the problem they were trying to solve and cut it down to the absolute essentials. With the library one of the things we found was that the stronger we could make the contributed library, the original library, and the more carefully parameterized, the easier it was to implement it on the many different machine families we were on. By the end of the 70s the library was in use on fifty to a hundred different platforms. The users also just didn't want to spend time reading documentation, they wanted to have access to the documentation, but wherever possible they tried to work without it. So it was essential to be seen as systematic in the way parameters were described, parameter sequences were laid out in the calling sequences, because people would try to guess what we were doing rather than consult the routine documentation. By the end of the 70s, a NAG FORTRAN manual was twelve volumes, involving some 2000 sides of documentation.

HAIGH: And with the shift from just Nottingham where you're working with one site and one set of users to NAG, where you can ship to almost all of the computer centers in Britain, presumably it would become important to have the computer center staffs at the various places as a first main level of support to help the users?

FORD: Absolutely. What we've done is develop a very careful chapter introduction which explained exactly why algorithms had been selected for inclusion in the library and how problem types could be identified. It is a fact that the last four people I appointed at Nottingham, who stayed there and continued running the numerical library service we'd set up, are just on the point on retiring. They've been working in that advisory capacity for the whole of their working lives, and the broad intentions of what we're seeking to do are unchanged. Obviously the way it's done have changed because of the coming of the Web. The first time we used e-mail was in 1976 as a communication means, we were given the first contact outside defense in Europe, by the people working in the DOE because they wanted to have access. So we'd been involved with online services of one sort or another from a very early time.

But yes, you had in universities, and in large industries, and Government laboratories, you had a central computing facility. And into that facility, you could put an advisory desk, and advice on programming too, and everybody went there to get that advice. With the coming of the workstations and the setting up of project teams, of course all of that infrastructure has now been broken down, although because of the coming of the grid, I guess some of it may come back again.

HAIGH: So was that then, was that shift a challenge to NAG to work more directly with the end users rather than being mediated through the computer center staff?

FORD: Well those of us who were involved with the running of the core of the library stayed actively involved in our own university computing centers. So, for example, when I moved to Oxford, Steve and I made sure we spent some time on the advisory desk, so we kept that valuable contact. Also we'd also got sufficient reputations by then as being useful so that people made the effort to have contact with us and to find out if we had algorithms for X or Y or Z.

HAIGH: Yes. Now I read in one of the papers you gave me that the user would normally find that their routine almost but not quite dealt with their problem. Was it common early on for users or specific sites to make changes to the code?

FORD: We recognized that this was a fundamental problem. Our aim early on was just to give each site a compiled library, which had been thoroughly tested, and to allow the users access only to that compiled library. But we found we had a confidence problem, they wanted to actually see the code and see how it worked. So we provided a source text which they could inspect, but we never allowed them to take copies of it.

HAIGH: Oh, so that's I think the difference with the other libraries that I've come across.

FORD: Yes.

HAIGH: So from the beginning then it was not an open source project, except in the most literal sense, because the users couldn't compile their own version of the code?

FORD: No they couldn't. They could have access to the source code and see it, but they couldn't have access to the source code to run it. I guess, being funny, if they'd wanted to punch it up then we couldn't have stopped them. But basically on the library tape, and you will find it in the document there specifying the early library, the whole emphasis was on the semi-compiled library that we sent out, not on the source text. The free source movement for us now of course is a considerable problem but we can again talk about that later as well. We all worked in source texts as numerical analysts and computational statisticians. We developed our code that way but the code that was put out for use was always a compiled code on the computing system that everybody called, and they couldn't make arbitrary changes to it. The mistake made by SHARE (perhaps by the PORT library, who knows it's not for me to comment) was that people did change the codes, saying "I don't need that record." And then they'd come and say "I've used your library, your library routine doesn't work." This happened to me on the KDF-9 in Nottingham, and you'd look at the code and you'd look at their code and say, "well these records are missing." "Oh they didn't seem important." "But that's the control mechanism for the algorithm." "Well it doesn't work."

So having had that experience of people having access, if they wanted to, to the source, and of being able to vandalize it, we were quite tough. And this meant the NAG library developed a reputation for being reliable and robust, and accurate, and efficient, and it wasn't undermined by people playing with the code, and saying, "Oh that SHARE software's no good, you know. It never works." Which certainly is what happened. Kuki and Cody's code was in there but they couldn't tell the difference between that and other code of less quality. Also Kuki code which had been spoilt by people playing with it.

HAIGH: Yes. Now you've been calling it the contributed library and you've already mentioned that Wilkinson's code an important early element. Presumably the routines that you'd already gathered at Nottingham were a seed for the library, and you've mentioned, that by about 1973, someone who had come up with a good algorithm would be keen to contribute it to the library as a way of getting it into use. So was it the case that you were just able to sit back and wait for good contributions to come in?

FORD: Oh, no. The characteristic of all of us was to press on working. I think I've explained through the different sorts of activities they had that Joan and Shirley and Linda were energetic, and so were Steve and I, and my colleagues. Jim Wilkinson was also very energetic too. And having created the first library the desire then was to continue building the library. We've had some wonderful times, amazing times. For example, the quadrature group always met Queens University, Belfast, and so throughout the troubles we had all our meetings in Belfast. We were a force for reconciliation in Northern Ireland, for the quadrature group had a Paisleyite, two Unionists,members of Shin Fein, one explicit member of the IRA, members of the Alliance Party. We would fly into Belfast, and we'd hold our meetings, and the meeting were superb, but when lunch time came we used to try and talk about other things and we had some of the most vehement arguments and difficulties. So we learned in the end just to hold the meetings, expressly and solely talking about quadrature. That way we were able to achieve along with our advisors, people like James Lyness from Argonne National Laboratory, a world class collection of algorithms.

It's very interesting because we were looking for algorithms, and our friends at Argonne were building their PACKs. Then when the PACKs became well known, people would put together their own PACKs, as it were, to share the reputation of EISPACK and LINPACK and MINPACK. But then those algorithms were often put into the NAG library too, and so QUADPACK, for example, almost in its totality was in the NAG library (much of it before the total PACK was created), and so these were mechanisms for publication. The other thing that happened, I hinted at earlier, was that we had a pretty printer and the master library file system, we then moved on to develop tools for portability, language checking, parameter checking, the mathematical characteristics, the parameters for arithmetic characteristics, etcetera. From that we developed a set of Numerical software tools. Those tools led ultimately to something called TOOLPACK which was developed from Argonne National Laboratory, and we indeed for a period of seven or eight years at NAG, had a TOOLPACK service. So we were providing not only libraries to people but were also providing users with tools they could use to metricize their own codes, to make sure their own codes were portable across the different machine families. And indeed in the end we finished up with a colleague, Malcolm Cohen, developing the world's first FORTRAN 90 compiler, and indeed Malcolm is now the secretary of X3J3, which is working out the standards for FORTRAN 2003. FORTRAN is alive and well, and just like English it's taken on many of the characteristics, the good characteristics of other languages, and perhaps some of the bad ones , as it's developed.

[End of Tape 2, Side A] [Start of Tape 2, Side B]

HAIGH: So returning to the matter of the contributors and the origins of the different routines, is it true then that most of the routines were coming from people who were formally involved with NAG as part of these different subject area groups?

FORD: The people that had created the NAG project had interests in particular areas of numerical mathematics but, with the possible exception of Joan Walsh, none of us were internationally known or rated numerical analysts. Because of Jim Wilkinson, and his involvement, and his delightful reputation as a collaborative individual who encouraged people to work together, people were happy to work with us. Wilkinson was very supportive of people, I think particularly young people. Jack Dongarra was one of his protégés, I'm sure he's helped Cleve Moler, he certainly helped me, he helped Sven, he helped Kahan, and so on. In each numerical area of mathematics and statistics we set out to create a group, a community, who would provide algorithms to us. For example, in ordinary differential equations: Larry Shampine, Allen Hindmarch, Bill Gear, the people from Toronto for example Tom Hull, and then later we even had an input from Yanenko and as I mentioned from Marchuk. Most of the code was actually put together by Ian Gladwell, but all of these other people fed into it and we were able to build the library subject area by subject area. There was a sort of peer review going on within the group as to which the best algorithms were. So initially we started using pedigree, which is dangerous, it always looks backwards. It became vitally important to have test examples carefully selected so that you were studying specific numerical characteristics within the computational area in which you were working. Hence we not only had these communities of individuals who were providing algorithms, but within that group there would be people who developed a specific interest in preparing and holding test suites, so they could prove their algorithms were better than other people's algorithms.

HAIGH: And was that modeled explicitly on the kind of academic peer review traditionally used for scientific papers?

FORD: It was actually rooted in that, but it wasn't done for that reason. We had a slightly different set of selection criteria. The fact that the analysis was set in a proper Banach space was not of always fundamental importance to us, although of course we wanted the underlying mathematics correct. Wilkinson had invented backward error analysis to enable that sort of analysis to be done. That was universally used I think throughout our work. We also had error checking so that we tried to get some understanding of the tolerance that our results had as well. The library was almost unique in that quality too, as very little software outside the library software was developed with those characteristics.

What we were trying to do always was to get this practical understanding, in each subject area, of how we could best identify and structure the subsets of problem type. Linear constraints, nonlinear constraints, with a computed Hessian, with an estimated Hessian, and ultimately, such separation had a mathematical basis and it was important to understand that mathematical basis. But from the point of people trying to solve problems, the important thing was to recognize the subsets of problems and have algorithms available to address those. Then to have the test software which recognized the subdivisions, and evaluated in depth the specific characteristics within that subset so that you could describe to a user with the problem exactly what the characteristics of his problem were (much understandings came from that) but then also which algorithm had been included in the library to solve that problem.

HAIGH: So the creation of code and testing were both done by the collaborating centers, although different centers were creating and testing code.

FORD: What happened was that the library contributors in specific numerical areas were creating their own algorithms ultimately to the language standard and using the underlying model that we'd described. We would then take their code, ruggedize it and polish it. Commercial software can not, I'm afraid, in the end, be Ph.D. research type software, it has to have other characteristics.

HAIGH: And in this case by *we* you mean the paid full time NAG staff?

FORD: The people who understood those characteristics. Some of those were the paid full time staff, but some of them were people like Stu at AT&T, or Margaret Wright, who ran the department there, who happened to develop that interest, not only in the algorithms were they developing, but of creating software that was as reliable and robust as possible. The idea was to give these people the tools, the software tools, to enable them to do that, because the better the quality of the contributed library software that came to us the less work we had to do before it could be incorporated in the library and made generally available. Does that make sense?

HAIGH: Yes. So in some cases the code that you received would need considerable work, and in other cases the people producing it would have gotten more on board with this kind of culture of producing robust, quality, saleable software and it would mean much less work, certainly much less refinement?

FORD: Yes. I think in fact early on in the library we had a wonderful colleague, Cliff Stone from Manchester, who brought some code and said "There's my code." I said "But Cliff that's useless it doesn't conform to standards." He said, "Oh I can't write language to those standards, you know I thought that was an aim, not something that we have to achieve." And we said, "Sorry Cliff you can't work with us. You've got to have the quality Cliff, you've got to meet the standards." But when we were working in Algol 68, the first people on Earth to create an Algol 68 compiler were Sue Bond, and her colleagues at RSRE Malvern. I asked them to give me document about how to write good Algol 68, and they said "Sorry Brian it's impossible." And we then sent them some code written by people from a particular British university, some of the early code in Algol 68, and virtually by return we got a paper on how to avoid writing bad Algol 68. Because it is an art… I think it's become increasingly a science writing good code, but it's also an invaluable art. You do need those sorts of guidelines.

HAIGH: Now one of the terms that I know has been used to describe the NAG approach is this idea of the "NAG library machine." Could you say what you consider is distinctive about the approach that you adopted?

FORD: The NAG library machine, which is a paper, which I started writing when I was on a sabbatical in Chicago at Argonne, was modeled on the machine of Mayor Daley, the famous political machine of Chicago. Not the unhappy side of the exploitation of a political process, but the idea of having a disciplined coordinated team of people, with shared objectives. Not interested in making money. NAG is a not-for-profit company, so the drive isn't to make money per se, the drive is to get quality software, quality tools, used by the community at large and particularly the academic community, and to work collaboratively together. So yes, there was robust competition. Fox and Wilkinson showed that in the early days. Within the EISPACK group and particularly in the LINPACK group at Argonne, Pete Stewart, Cleve, the other people, were all first rate numerical analysts, all with their own ideas, but were working collaboratively together to prepare integrated and quality software.

You need the collaboration, nobody can afford to bring together, then and now, the best numerical analysts in the world all working on one project and pay them. Even IBM can't afford it, even the U.S. Department of Energy can't afford it. If you work together collaboratively then you can create a science that is internationally used and builds quality into the science and engineering that's done with it, and confidence into what's produced with it. It's a vital ingredient. The other thing about the machine: it used people's best abilities and integrated into an activity different sorts of ability and ability groups. So that you'd have certain sorts of computer scientists developing the tools, you'd have people who are specialists on one particular machine family, helping you optimize the library's running on that machine range. You had numerical analysts and computational statisticians who were each providing algorithms in the area of their own technical interest, encouraging them to interact with one another positively, to create the best, rather than having sometimes arid competition.

Writers -- one of the areas NAG failed is we've never got people to write books, subject books, about the use of the library in the way that other activities have. That is one area where our collaborative model didn't work, perhaps we didn't put enough emphasis on it. But when it came to writing, for example, user documentation we learned an immense about, about how you present information to people, so that they can understand it and read it. One of the things we learned, a killer, was that the desecration of the English language can be a good thing, if you want to succeed with an international community in numerical computing. The average American scientist I'm told has only ten thousand words of English, the average educated scientist in the United Kingdom would expect to have at least thirty-five thousand words of English vocabulary. In English, in English English, you would never repeat the same idea using exactly the same language (the same words) in the same sentence or closely related sentences. In American English you almost invariably repeat the same idea using the same language. The impact of that is that if someone looks at a manual, and English is their second, third or fourth language, they get a much better clarity of understanding from an American document with its restricted vocabulary, than from an English manual with its beauty of English perhaps, but bringing in a mystique which is completely lost on someone who just wants to look at the doc to be able to effectively use the software.

HAIGH: And was the documentation being produced by the same people who wrote the code, or was this something that was being handled by the full time NAG staff?

FORD: No the first drafts were invariably done by the people who wrote the code. But as we polished the source text of the library, so we unified the presentation of the documentation as well. And the longwinded explanation I've just given, was because we learned we had to go to a subset English for the documentation as we had had a subset FORTRAN for portability.

HAIGH: Yes. And I wonder if there are any specific aspects of the NAG machine approach that you can point to and say at the time we did that nobody else was doing this.

FORD: The whole idea of building a library collaboratively had never successfully been done before. We didn't do it for that reason, we did it because that was the only way we could operate, it's the best use of resources. People had tried to work collaboratively on things like this before, but since they'd not had standards they wrote everything to, what usually happened is the activity they were involved in had less and less quality built into it, so people weren't interested in using it. By having standards, carefully documented standards, which we rigorously kept to, and then later on wherever possible making machine-based standards, so stuff could be properly checked before things were allowed to go out. The contributed library would always be rigorously tested, that you conformed to standards, before it was allowed to go out from the contributed library and be implemented on the different machines. I think that was unique.

I think also the fact that we weren't financially profiting. Our friends at IMSL, who started at very much the same time, and had always been the other commercial library that we competed with, their drive was always a financial one to make money. There's nothing wrong with that. I've learned to accept the profit motive thirty odd years on, but it was very much a different drive. People cooperated with us because we weren't making money. There was a clear altruism in what we were seeking to do, which was also in the interest of the science, and clearly as far as the Computing Board was concerned was a key vehicle for them encouraging collaboration and cooperation amongst U.K. universities in the use of valuable computing resources.

HAIGH: Now this period of the early 1970s, was also then people first started talking about the concept of software engineering, at least that there should be such a thing.

FORD: Absolutely.

HAIGH: Was that a kind of discussion which you thought was relevant to NAG's work?

FORD: Very much so. As I indicated earlier, we had our first software tool from Lloyd Fosdick, and he was in a department of Computer Science, was interested in software engineering. The TOOLPACK activity involved Lee Osterweil and various other people with strong computing backgrounds, but with limited interest in numerical mathematics. We were able to be a test bed for them. For example, Stu Feldman came and spent a few months with us in the late 70s, or early 80s, I can't remember exactly when. Watching these massive bodies of code that we were handling (the routines were quite small, the largest was three or four thousand records, but if you had several hundred of them , each with associated on line documentation and the example programs with input and results, and the stringent test programs with their input and results, the need for checking that the library's been properly implemented on a particular machine range or not) these great clods of structured information were what led Stu to create the make file model in Unix and indeed his seminal paper expressly refers to his work with us, working on the library and that experience. That's the most famous example of direct connection.

All sorts of tools in TOOLPACK came from the metrification, specification, testing, checking, of the code that was required for the library. The methodologies developed there were found to be relevant in much larger application programs, in much bigger software activities. So the tools were something that were very important to us, and we've developed our own and indeed still market and support our own. The other thing we learned was that graphics and visualization were very important too. Early on, we worked with the people at the National Physical Laboratory, and with Christian Reinsch, in developing algorithms for graphics. We found that many of the algorithms which had been developed up to that time were done on some sort of pseudo aesthetic grounds, people sensing what would be a nice shape and ways of enforcing that in the algorithms that they designed. What we set out to do was to make sure that our graphical algorithms preserved the mathematical properties of the data and we worked with Curtis and Clenshaw, and people at NIST in the States, to develop algorithms which had the characteristics of preserving the mathematical property that were found in the data or computed in the data, and then fitting the best curve or surface for the data with those algorithms. And what we found is that over the years people have continued to use these algorithms some of them now twenty, twenty-five, years old, simply because they do have that accuracy. We have some lovely horror stories about some of the graphical algorithms used, for example, by Boeing and other aircraft manufacturers, people working in nuclear installations, how by not pursuing the mathematical properties they've unwittingly led themselves into difficulties.

HAIGH: I'll ask you more about the graphics things tomorrow when we talk about the expansion beyond the original library. So we've obviously still got a lot to talk about, I think perhaps to wrap up the first session there are two things that I could ask you about. So the first of those: you've mentioned Wilkinson a number of times, unfortunately he isn't around any more and nobody ever did an oral history interview with him, so if you have –

FORD: Excuse me, that isn't true. There is an oral interview by him with a man called John Nash, and the tapes are available, I believe, from SIAM, it may be from ACM.

HAIGH: Oh, that's good to know. I don't believe that's in the Babbage library of oral history, so I'll see if I can get that. [see Oral History with Wilkinson on this site]

FORD: It is somewhat, …. idiosyncratic. Jim, obviously, was a very extremely fine numerical mathematician, we've mentioned his work on Pilot Ace, we've mentioned his creation of backward error analysis as a means to evaluating the quality of algorithms, we've also mentioned his work on the algebraic eigenvalue problem. The numerical linear algebra eigenvalue problem, not the ODE eigenvalue problem, was one of the first numerical areas that was completely, as it were, laid out and solved. Perhaps one of the easiest, but, nevertheless, it was a vitally important one. I always remember being at the Royal Society with Jim, in the middle 70s, and a man called [Brian] Trubshaw, he was famous as the Concorde test pilot, coming up and saying to Jim, "I'm told that you saved my life." Jim said, "Yes I probably did actually," and this was because earlier prototype aircraft were shaking apart in the air, with the rivets coming out (and early jets like the Comet) with the vibration harmonics (eigenvalues) of the airframe being exaggerated until the thing just flew apart. Jim solved that problem.

He was so able that I think most people were somewhat in awe of him, but he did encourage you to work and was very supportive so it didn't matter that I was a nobody from Nottingham. He'd come to Nottingham to talk about his work on the eigenvalue problem because of the university audience we could put before him at that time. Having recognized that commitment to helping people solve problems, he then worked systematically to get the numerical analysis community at large to cooperate with one another. Yes, of course, there was competition, but at least it was toward a quality end.

He had some fascinating characteristics, he was the first person to teach me to drink red wine. When I went to Ann Arbor, Michigan, with him I'd never seriously drunk wine at all. I came from a Methodist family, which in the main didn't drink. But in that first fortnight the only wines I drank were Premier Cru Bordeaux; Lafitte, Latour,Mouton Rothschild, and so on. Well my taste buds, by the end of the fortnight, were thoroughly activated. The trouble was that it was at a quality of wine that is difficult to sustain! Jim loved company so we'd go out every night in Ann Arbor, for example, to the Pretzel Belle drinking. Fortunately I've got hollow legs as well, but it was great company, had enormous vivacity. He loved food. His wife, Heather, who is still alive, was a great cook. It was usual to go to their home and have a five course dinner with six or seven wines, all chosen to complement the food.

Jim lived in a modest house in Teddington, Middlesex near NPL. When he died, the value of the wine in his house, which was stored everywhere, was worth four times the value of the house. He also had an unhappy habit of being frugal with his own resources. So even towards the end of his life he would use his bicycle, (he loved cycling,) to go and buy a hundred weight of coal, and carry the coal home in a sack balanced on his handlebars, wobbling his way through the traffic. He used to make some of us feel very nervous. He came from very modest roots, was the International Secretary of the Royal Society. Probably didn't get a knighthood because of… one of his failings: he had an immense attraction to pretty women, but that's shared by many of us. He was a very kind man, he always saw the good in people. At the meeting in Ljubyljana, Françoise Chaitin-Chatelin was speaking, about the eigenvalue problem, and Vel Kahan began to argue strongly with her. This was not a fair competition in front of an audience of a thousand. Jim just quietly calmed the whole thing by asking Vel if he was quite sure he was right about one of the assertions he had made. So he drew people together. He enjoyed collaboration and cooperation with them, and what mattered was the quality of their thinking and what they produced and not the reputation they were working and speaking from.

HAIGH: And the other thing I thought I should ask you about, is after your 1973 move to Oxford, the continuation of your own life and academic career beyond NAG.

FORD: When I came to Oxford I was given Reader status in the university, and then shortly afterwards Professorial status in the Subfaculty of Computation within the Faculty of Mathematics.

HAIGH: And that must have been very soon after receiving your Ph.D.?

FORD: Correct.

HAIGH: So was that an unusual distinction?

FORD: I guess it doesn't happen very often. I mean really brilliant mathematicians obviously move very quickly. I clearly wasn't that. I fear it was probably a status thing. I'd published papers in numerical mathematics, numerical analysis, applied mathematics particularly quantum mechanics, areas like quadrature, random numbers. Then later on I came up with the idea of problem solving environments, and did an IFIP Working Conference on that in Sophia-Antipolis. I'd already run an earlier Working Conference on library issues.

Through the period I had occasional Ph.D. students, sometimes jointly supervised, sometimes supervised on my own. I guess my best known Ph.D. student is Steve Hague my deputy, who did a D.Phil. in computing science, intelligent editors, etcetera, here in Oxford. But I've had them in civil engineering, in numerical mathematics, in computer arithmetic, and applied mathematics, and in computer science. But eventually I found that I couldn't be fair to them. The business was taking so much time and I didn't have the profound depth of knowledge in a particular numerical or engineering or science area. I was sort of an intellectual gadfly: I would recognize an emerging challenge, see an interesting area developing and I would do some work in it, but then the work we were doing, necessary for the libraries to appear complete, and I would move on. So, for example, I had an interest in Tools, Methods and Languages, which was a European meeting. I did problem solving environments, which was a vital broadening of the methodology and basis for doing scientific computing. I did the creation of all the scientific background to the ADA computing language as an interest as well.

HAIGH: So were you on the group that was defining ADA?

FORD: I helped define aspects of the operators that we needed within ADA for scientific computing, and then worked them forward into creating the infrastructure. I've been principal investigator on twenty-six major programs funded by the European Union. A major program would typically have three or four million euros. Hence I've been responsible for spending over one hundred and thirty million euros of European tax payers money!

HAIGH: And were those projects related to NAG's activities?

FORD: Usually.

HAIGH: Tell me about any important ones that weren't, because I'll ask you about the NAG ones tomorrow.

FORD: Well, for example, in 1978 we started implementing a package called GENSTAT which is in statistics. {Ronald A.] Fisher had been at Rothamsted Experimental Station, the father of modern statistics in the early nineteen hundreds, and then there had been Yates, and then there was Nelder. Nelder is still alive and he has worked with us in all NAG activities since 1978. Constantly the problem was getting the software coded in a way that was portable across different machines, and user manuals that could be read by people with less than a Ph.D. in a particular area of statistics. So we had GENSTAT, and we also had Iris Explorer, which is a general purpose visualization system which had a problem solving environment within it. We proposed and received funding for a project to put GENSTAT into Iris Explorer. Now I think this is an extremely good idea, and it's still not been properly exploited. In statistics you often have massive banks of data that you wish to analyse, and you want to be able to study the characteristics that flow from that data. So you take three or four dimensions within the n-dimensional space and look at it say for a syndrome where there's a fundamental change or breakdown. In the visualization system you could do that, so we set about developing STABLE. In the STABLE project we had Albert Prat who's the Dean of Science at the Universitat Politècnica de Catalunya, a very able statistician. We had the people from Limagrain, who are the largest seed merchants in Europe. We had the people from Rothamsted, Roger Payne and his colleague. We had people from NAG. We had people from GESA, who are the electrical power suppliers in the Balearic Islands. And that was a typical European group, representatives of a small country, of different technical interests, a worthwhile project, and we created STABLE.

I've done that twenty-six times. We've developed parallel algorithm software that way, we've developed different sorts of computing environments that way, learning about open systems like libraries, and closed systems like GENSTAT, how different sorts of computing specialists and technical specialists, can get the most out of the different sorts of computing environments. And helping big industry, the financial markets, solve their mathematical problems by developing systems that are specifically engineered to solve their problem types.

Again, the great gift has been getting people to collaborate. Getting them to talk to one another, helping them to overcome the mild irritations when you meet people who you almost get on with but not quite, by having carefully designed aims and objectives and making sure there's something in it for everyone, so however you look at what you're doing, everybody can see some value for them in what's being achieved, and a place for them. Of course, the great tying qualities in Europe are bread, (almost everywhere makes its own bread,) beer,(virtually throughout Europe there is high competition where beer's concerned,) wine, (Europe is supremely good at making wine, I believe). And then excellent meat and vegetables, give us wonderful food. And by creating communities every time, (of people with common objectives who work together,) being able to set challenging objectives that fit it into the work program of ESPRIT 1, ESPRIT 2, or ESPRIT 3, the Fourth Framework, Fifth Framework, Sixth Framework. But worthwhile things for people to use in their own working environments as well, a bank here, a plastic manufacturer there, seed merchant here, and the academics, always, of course. And it was a replicatable approach, and we had fun. On one occasion a program officer tried to persuade me to buy her brother's farm, because she knew that my brothers-in-law were keen farmers. We've had marriages from our project groups as well, and all sorts of relationships.

HAIGH: So you think that those European projects have been the most important kind of activity outside NAG for you for the last few decades?

FORD: They have been important. I think the American ones we've done have been just as important. One of the most important streams in my own life has been consulting in the United States, and I've been fortunate enough to spend fifteen years working outside the fence at Livermore, a month each year. The university only pays me a professorial salary, and it was recognized by my colleagues that I kept being offered jobs to go to the States, and earn substantially more than I got in Oxford. So the understanding was reached that I could go off and earn enough money to keep bread on the family table by doing a month a year in consulting in the States. That's only part of it. The other part was access to big science, big projects, big systems, so carrying the methodology that we had in the library, and developed forward into the problem solving environments, into solving major problems, major issues in the US National Laboratories.

[End of Tape 2, Side B] [Start of Tape 3, Side A]

FORD: In Sandia Livermore, Lawrence Livermore. I was used for six years to try and encourage conversation between the quantum chemists and numerical analysts in the States, with very limited success.

HAIGH: Do you think you were successful in other areas in transplanting anything from NAG into this U.S. context?

FORD: Undoubtedly. The SLATEC library developed by the weapons lab had many of the characteristics that the project itself had. I don't think we were anywhere near as successful in getting the scientists and engineers in the labs to adopt the same programming standards and accepting other disciplines, simply because of the awesome computing power that they had at their disposal. But some of them certainly learned from our approach and the quality of the software and the programs that they developed were marked by these influences, which is why they kept paying me to go there year by year. And in a similar spirit I was able to go and work with IBM, at Yorktown Heights, and do some consulting, not a lot, with AT&T at Murray Hill, New Jersey.

The important thing was the constant refreshing of what was happening in IT generally, and in the development of systems for scientific computing, in particular. So that at Fox's wedding (I believe in 1974), Christopher Strachey told us all about workstations. You remember the VAX VMS system, we were the first people outside Digital to use the system, and in a moment of elation I described the system as an implementer's dream and found myself used on advertising all over the world. What that led to was a recognition by computer manufacturers that if they were going to have systems that were going to be used for science, they had not only to have hardware and operating systems and language compilers to standards (standards were recognized as important because of issues of portability), but they also needed numerical algorithms libraries from which the users could begin immediately to build their own programs on the new system. So at each major step of a movement through computing, people have recognized the need of having these vital components that we create available at the beginning, not only for the workstations but also for the PCs, and for networks and for the grid. So, for example, we're now directly involved with Intel and in AMD in creating the core math libraries on which the system's basically built. I stopped lecturing in the university in the late 80s, I think that the Lab realized that they needed either specialist numerical analysts or people with very specific computing science gifts. Yes I could do the M.Sc. lectures on scientific computing but that's pretty boring stuff actually over a protracted period.

HAIGH: Does scientific computing still figure in a significant way in the Oxford computer science curriculum?

FORD: Yes. Mathematical biology, computational biology, is all the rage. We have an e-science program in the United Kingdom comparable with the ASCI program from the DOE, and the NSF programs. People aren't taught as much as we would like, but they're taught a lot. One of the great things for me has been keeping in touch with the way the hardware technology was developing, the firmware was that being evolved, the tools that people needed, and the use of visualization to enable people to be much more effective in understanding their areas of research interest and the driving characteristics within them. I've been very lucky in that, because of Jim Wilkinson's interest and going over to Argonne in 1971, I've been able to continue a program of visits and friendships and collaboration throughout the U.S. Department of Energy and the major universities, and the computer manufacturer laboratories up to and including the present day. It's well worth remembering that the increasing power that has come from numerical algorithms over the period has been as great as Moore's Law, as far as the hardware is concerned. Moore's Law of course is now no longer being met, there are problems in chip manufacturing that mean that continuous growth is no longer being achieved. The real challenge, as I prepare to retire, is to ask, "What's going to come after the chip?" We started with the valves, in EDSAC 1, in very early designs of Pilot Ace and the relays, we then moved to chip based computers. An exciting question is: what is going to be the basis of the next family of computers which will not be chip based, they're going to be driven by another technology.

HAIGH: That's probably a good note on which to conclude the first session then.

Brian Ford interviewed by Tom Haigh, Session 2, 30th June 2004. Continuing from the previous interview, again in the office of NAG, in Oxford, the United Kingdom.

HAIGH: During this second session we also expect to be joined at some point by Steven Hague, long time deputy director of NAG. And by Sven Hammarling who did work on LAPACK and other projects for NAG.

So picking up where we left off in yesterday's session, you'd already referred to the initial establishment and, I think, the first meeting of the IFIP 2.5 Working Group. I wonder if you could say some more about how that group developed over the years, and what you considered to be its most important contributions.

FORD: I think the foundation of the group sprung directly out of the activities in Lublyjana, at the Working Congress there. I think Cleve decided that he wasn't interested in that sort of international group, but I certainly was and Bo Einarsson, who also picked up the challenge, certainly was too. We put the case to TC2 for the formation of the group, I hosted two meetings of IFIP Working Group 2.1 and 2.2 here in Oxford in 1972 and 1973, and through those meetings established the credentials of what we were seeking to do. We then had a founding organizing committee which met at Argonne National Lab, in Chicago, under the chairmanship of Jim Pool, and he became the group's first chairman, Bo Einarsson was the first vice chairman, and I was the first secretary. Christian Reinsch was also there, helping us draft the paperwork.

We drafted at that time aims and objectives for the Group and its initial state and we did with care. The group at its meeting in Washington the beginning of this month, reviewed those aims and objectives and made modest extensions to them, but actually found nothing that it didn't actually agree with. So it was a careful job and our aim was obviously to forward numerical computing by the provision of high quality algorithms and software, through the development of computing systems, which had better arithmetic and compilers that conformed to international standards. It was to be a forum also for the development of algorithms, so that the continued development of algorithms at least kept pace with the continued development of hardware and a meeting place world wide. So, although the cold war was already heavy on Europe and on the world, we wanted to make sure that the Russians were involved in the Group as well as people from Europe and North America, and indeed the rest of the world.

HAIGH: And who do you recall as having been the most active participants during the 1970s?

FORD: The most active participants during the 70s were the founding group (Jim, Bo, myself) driving it forward, and a group of people involved in arithmetics, particularly Tom Hull. Although Kahan wasn't a member of the group, we had an active involvement with him. People were involved in language standards. A strong interest throughout the life of the group has been the evolving FORTRAN language, and we've always had world authorities on FORTRAN within the membership of the working group such as Brian Smith and John Reid. And people working in specific subject areas, say, for example Professor Doctor Hans J. Setter, from Vienna, has always been very prominent in the discussion of ODEs with Tom Hull and with other members of the group. Also in the early days, there was this strong concern about the development of portable software. So the group had an activity, which I led, looking into the issue of parameterization.

So it was a very fertile group, it had a quiet period in the late 80s, early 90s, but its membership has continued to realistically grow. Membership of IFIP Working Group 2.5 is seen as a badge of honor, and desired by people in the world of scientific computing, numerical software in particular. Recently we've strengthened the group moving forward with the addition of people in their middle- to late-30s, people of international reputation like Bill Gropp of Argonne and Ian Reid from NAG. Bill has the makings of a world leader, in tools for the network and various uses of the grid, and he's caught up on the one hand with the big labs, on the other hand with academia as well. He is an instance; there are a number of people of that quality who've recently joined the Group. Working with the community at large, what it has done is to force the manufacturers to give us reasonable machines for scientific computing. Appreciate the scientific computing market is not one of the strong world markets in IT, the strong markets are the commercial industrial markets, and obviously the personal use market too. So that in the States there is now concern about the development of the work for the high end computing which is required for National security, etcetera.

HAIGH: And how is IFIP able to do that?

FORD: We are able to exercise an influence by agreeing on important issues such as computing languages, such as computer arithmetic. Cody was a member of the group while he was doing the IEEE work, for example.

HAIGH: And did the people on the Working Group serve as eminent individuals, or as representatives in effect of their organizations?

FORD: They actually served in three capacities. They're all eminent individuals, very carefully selected, and there were people put forward for membership who weren't elected because they weren't felt to have the necessary kudos and weight. They usually came from large organizations, but not always. So Ed Batiste from IMSL, for example, was a member. Tom Aird was a member because he was a distinguished numerical analyst, but also people represented or were known to be from specific countries. So, for example, the United Kingdom has always had just two representatives, though at the moment we actually have three, because shortly I should be retiring, and I should then become an associate member of a group rather than a full member of the group. On an associate basis we're allowed to remain members until we die. But I'm very aware of representing the United Kingdom and have used my membership of the IFIP Working Group sometimes to lobby the Department of Trade and Industry, Government laboratories, etcetera. Certainly the Russians, who've only had academicians from the Soviet Union as members of the group, which shows the esteem it was held in there.

In the States we've always made sure that the membership from the States was less than fifty percent of the total membership of the group because there is so much going on in the US, and the people have been from the big labs, mainly Argonne, funnily enough. We had to be very careful, because in our area so many world recognized people were from Argonne: Jim Pool, Brian Smith, Jim Cody, and of course Bill Gropp now as well. And we found that people were willing to receive us and talk to us because we were from the group.

We could only exert pressure since we had no formal standing. IFIP as you know is a subcommittee of UNESCO, and UNESCO is a sub committee of the UN, so it does have a place in formal structure which occasionally is used. It's helped us getting into the Soviet Union and sorting out a couple of problems with getting people to meetings. One of WG2.5 major contributions has been its working conferences. We've recognized that a new subject in the area of numerical software has achieved a certain maturity, and importance, that warrants a working conference on the topic.

For example, Francois Chaitin-Chatelin, from France, and I chaired a Working Conference on Problem Solving Environments and we and the Working Group were aware that this was the first major world meeting on this subject, establishing its scope and ground. The Programme Committee was myself as chair, Françoise from France, Ingmar Dahlstrom,(numerical analyst specializing in Ordinary Differential Equations,) Stuart Feldman (world class numerical software and computing language and software tools expert), Morven Gentleman,(similarly well known), Jan Kok (an expert in Ada), Chuck Lawson (numerical analyst in curve fitting,co-inventor of the Level One BLAS), Marek Machura,(from Poland), John Nelder (probably the world's first great computational statistician), John Rice (numerical analyst in elliptic PDEs,organizer of the first meetings on Numerical Software), Martin Schultz (distinguished computer scientist), Brian Smith (international FORTRAN expert, first student of Vel Kahan to complete as Ph.D., his particular battle honor to have been through the Kahan mill, and expert on numerical software and related software tools) and Margaret Wright (numerical analyst in optimization theory,head of the CS Group at AT&T, Murray Hill, President of SIAM, and now head of the NYU Computing Science Department). An excellent committee that organized an outstanding meeting.It was a real honour to chair both the Programme Committee and the ensuing Working Conference.

HAIGH: Let's read out the citation for the transcription, that's, "Problem Solving Environments for Scientific Computing", edited by B. Ford and F. Chatelin, and the publisher appears to be North Holland, and is there a date on that?

FORD: The conference was held from the 17th to the 21st of June 1985, and the book bears an IFIP copyright of 1987. ISBN number 0444702547. The working conferences, of which there have been eight so far, have generally achieved substantial penetration into a new area. It would be arrogant to say all. So that was another major contribution of the IFIP Working Group. The meeting in Washington we've just had confirmed the text of an IFIP Handbook for Numerical Computation, to be published by SIAM, very much along the quality lines of the famous linear algebra handbook of Wilkinson and Reinsch. We stand in that quality path, and have those quality objectives.

HAIGH: Okay. So in terms of accomplishments of the Group, you've mentioned their successful lobbying of manufacturers to get them to take scientific computing seriously, you've the academic and intellectual value of the conferences, charting new areas for the community, and you mentioned the forthcoming handbook. Are there any other particular areas of accomplishment that come to mind over the decades?

FORD: Well, as I said, the impact on international language standards, particularly for FORTRAN, the involvement with Cody and Kahan, and the development of the IEEE Standard for arithmetic, the pressure on Seymour Cray to clean up his act and give us some clean computational power at the high end, and the general continuing recognition of work for design and development of scientific algorithms. So many members of the group have also been associate editors of TOMS, and it's my understanding that that's one of the most highly regarded academic journals that ACM runs. So we've done our bit.

HAIGH: Yes. And the international character of the group, as you've said, is also the distinctive thing. It's clear that a community evolved within the United States relatively early on in this area of scientific computation..

FORD: The community developed in the English speaking world early in the development of numerical and scientific computing. People like Wilkinson and David Wheeler created an environment from which all this came. And in the States there were people like Householder and Givens, particularly working in numerical linear algebra, from the middle- to late-60s. Remember the computer was built initially for numerical computation, and sprang from our community. One remembers figures in the US like von Neumann but its appeal spread quickly to a much broader community of users as Turing had always envisioned, but the numerical community was in it from the beginning. Not only were machines being built in the United States and the UK, (the UK was particularly rich because obviously we had the Manchester group, the Cambridge group, and the people working in the National Physical Laboratory), but in Europe similarly, most countries developed their own early machines as well.

So with the creation of the Working Group Derek Dekker from Amsterdam, was one of the early members, and Hans Stetter, from Vienna, as I mentioned already, was the second, and there were others. Christian Reinsch from Munich was a founding member. And so when the Group was formed these people represented a fertile community in Europe, which had been meeting using the excellent train systems of Europe. The International Congress of Mathematicians they all attended, and bodies of that kind. Of course the Russians had always been at those meetings too, and the Russians always found it easy to have contacts in France because there were strong political ties between people in the Soviet Union and groups in France which aided those contacts. Certainly what happened with the Working Group was the drawing together of the often rather fractured nature of contact in Europe. Then with the coming of the Multi-Annual Program of the European Commission a little later, followed by the various Framework Programs of ESPRIT that acted to draw Europeans more coherently together specifically in the area in which we were working, as well as through the Working Group.

HAIGH: So then you see the distinction not so much between the U.S. and the rest of the world, but as between the English speaking countries and continental Europe.

FORD: I think that's the way many people in continental Europe felt, particularly the French. But having said that, the shared common language, the fact that there was a special relationship between the United Kingdom and the United States, which is real and had existed since the Second World War, meant there were all sorts of works inside computing that we could do together, in which it wasn't so easy to involve colleagues from Europe. What the Group was able to do was to solidify activity in the Anglo-Saxon world and with the broader European community.

HAIGH: So if you return now to NAG itself, you had spoken about the origins of NAG, it's existence as a consortium funded by the Computer Board to provide this high quality meetings to computer centers –

FORD: The Computer Board funded a support system for the voluntary collaborators who were actually working in their own time from their own institutions.

HAIGH: Yes. And with a lot of the actual work coming from volunteers.

FORD: That's right.

HAIGH: Where we left the story, I think NAG had moved to Oxford, it had begun to cover multiple platforms because there were multiple platforms in use in academic computer centers in the UK, but it had not yet begun to sell its products, or to provide them internationally, or to non academic users.

FORD: This is right. The first indication that the software would find a broader market was when the University of Braunsvig, outside Hanover, but within the old British controlled sector in Germany, following the Second World War, was given an ICL 1906A by the UK Government. It asked for the NAG library for its gift from the Government. Now this absolutely threw the Department of Education and the DTI, because at that point we weren't formed to a point where this could easily be done. I remember a wonderful night with the British Ambassador and with Maurice Wilkes, who was there to give a lecture at the presentation about the world's first university computing service, namely at the University of Cambridge. We got drunk together with nothing else to do, and Wilkes and I did a reconstruction of world computing history up to that point. At the presentation the next day I handed over a copy of the manual and made a brief speech. That was in 1973. The University of Hanover, which is the big regional computing center in that area, which is still there and we still work with, then asked for the Library too. At the same time Rolls Royce wanted the Library for work on airframes, having heard of its quality from Jim, and others who acted as their consultants, and ICI had a need for it for some of the modeling they were doing too. By this time we're at the University of Oxford, you'll remember. We had no formal standing and the fear was that obviously if a plane fell out of the air and it was found to have occurred through our faulty software, you can imagine the consequences.

HAIGH: Yes.

FORD: So a process started to decide what formal structure the NAG Group should have in the long term. And this led to a meeting with a IP lawyer, Jonathan Anelay, from Morrell, Peel and Gamlen), who I'm glad to say then advised us for twenty-five years until he moved to become the solicitor to the University of Oxford, which was a new role for him, and a new role within the University. Jonathan and I, working with the inner group of NAG, but also talking to the Registrar of the University of Oxford, and its Vice Chancellor, came up with the idea of forming a not-for-profit company, limited by guarantee. The members would be the people who'd made voluntary contribution to the company. By having it not-for-profit there were two results. One was that people individually felt they wanted to give, because money from the activity wouldn't go into our back pockets. But also it meant that Government laboratories throughout the world, if they chose, and large commercial companies, like Shell, Phillips, and in the States, like DuPont and Boeing, and also people in universities elsewhere in the world, Stanford, Chicago, University of Tennessee, that they could provide software to us without breaking various financial and tax considerations in their own countries, and their undertakings for their own taxpayer.

That has worked from that time to this day. So, for example, when speaking at the meeting to celebrate my retirement last week, I was able to list over 190 individuals who'd actually contributed software to the library over the years. So the formation of a not-for-profit company, limited by guarantee (not by share capital) was agreed. It was then agreed that the staff employed within the University for NAG would be transferred to the employment of the company on 1st May, 1976. We obviously had to prepare Memorandum and Articles of Association and all the good stuff. By this time, you're right, NAG was being funded by three sources: monies from the Computer Board for Universities and Research Councils, some research grants from the Science and Engineering Research Council, which I'd been able to get for work on tools and work on use of the technology we were developing, but also the early fees from the users of the Library outside the UK universities, either overseas universities or large industrial companies within the UK itself. Interestingly the University suggested that I remain simply in the University as a full professor, but that everybody else should transfer out. I refused this, we all had to jump in the cold water together. I think it would have been impossible otherwise. And NAG actually became financially independent of the Computer Board on the 1st August 1980, and since that time has received no external funding from any Government body as support. It's earned money through grants or through the provision of services, if you take the distinction.

HAIGH: Yes. And in order to sell the product presumably there were a number of issues that you had to deal with in terms of whether it was being sold or leased, whether there would be a single fee or an annual fee, how support would be provided. These would not have really existed in the same way when the software was being given away to academic centers. So can you talk about how you dealt with those issues for the first customers?

FORD: Well Steve and I have not had the benefit of doing an MBA. What we've had to learn is the whole business of running a company from the bottom up. It's true to say I was very much the director of the company, setting policy in all of these areas. We had a NAG executive committee prior to the setting up of the company. From the setting up of the company we had a council of management. The NAG Executive Committee was Shirley, and Linda, and Joan, and me, and two or three other major library contributors, and implementers, Peter Kemp was involved early on. After the 1st May we had a formal Council of Management which had four representatives from our members, elected by them at our Annual General Meeting, a representative from the University of Oxford, which is where we were and two representatives appointed by the Computer Board for Universities and Research Council. Steve and I were ex-officio members of the Council. From 1978 when we started our first subsidiary in North America, we had a representative from NAG Inc., on the Council too.

Independently of that I also set up with Steve, and my colleagues, a Technical Policy Committee which has met annually for the last twenty-eight years, and this is of the great and the good, advisory notwithstanding its name, pointing us in the direction of where the market was developing in terms of scientific computing, in terms of hardware and software, in terms of licensing issues, in terms of the evolving Internet and Web. The Council had various subcommittees too. So there was that infrastructure. And so, for example, the Finance and General Purpose Committee was a standing committee which we now went to for advice on organizational and financial matters, but it was made up of academics. The Staff Panel was where we considered employment issues, the Membership Committee always looked at who were the members of the company, and made sure that appropriate people were invited to join. I think we have something like 375 members of NAG at the current time.

HAIGH: So did you become a member by using the software, or was it more complicated than that?

FORD: You became a member by making a contribution to either the creation of software for NAG, or by helping implement the Library and our products on particular machine families, or being a member of the great and the good, who'd helped us either through the Technical Policy Committee or through making some other personal contribution. The customers have never been allowed to be members of NAG. Our customers had a user group for some twenty-five years, but like so many other user groups fell into nonuse, simply because the world moved on and people related to one another in different ways.

So that was the structure and within that structure we then have the issue of providing a service to our end users. I developed with Jonathan, and with some good input particularly from Jim Pool, a software license for our products and services which made it clear, for example, that they could rent our products annually, that it was like the taking of a subscription to a journal and you paid up front, at the beginning not at the end of the year. That was vital for our cash flow. Remember we had no initial capital base, we never have been given any special resources, founding capital, we had to do everything ourselves from the cash we generated. So, for example, universities were given a discounted rate, since they have always been poor, at least they have always claimed to be poor. There was an annual license fee for industry and commerce, and typically the overseas universities would pay sixty percent of that. For their annual license fee they would get a magnetic tape on which there was a copy of the compiled library for the machine they were using, together with the example programs for customer use. They would get a library manual, which described the use and calling of the software. We always had an advisory desk that they could approach for advice on the use of the software and the installation of the software, and error reporting.

[End of Tape 3, Side A] [Start of Tape 3, Side B]

FORD: So a site would order a Library, they would sign a commitment to abide by the license conditions, and it was set up so that once they licensed the product they had to make the decision to discontinue using it. That has been the basis of use of the Library, from that time until the present day. We now also have a second area of selling components of the library, either as internal or external run times, internal within an organization, or sold for export from the organization (called from within other software). We also have technology licensing so that, for example, Maple have put part of the NAG Library into Maple, the symbolic computing system, and they've paid us a substantial amount for the licensing of that technology. We have similar licenses to that with many other ISVs, and this is an important developing part of our market.

But to go back to the early days again, initially we made the classic mistake and assumed that since the product was such a world leader everybody would rush to our doors to buy it. The good news was that genuinely happened for the first few years. There was a regular flow of orders for the library service and shortly, from 1978, we also started selling GLIM for the Royal Statistical Society, a product developed by John Nelder, and GENSTAT, a statistical system developed by Nelder and his colleagues at Rothamsted Experimental Station.

So we have the licensing, we have the support. Finally we recognized that we actually needed to have sales people. They were part of an alien culture for us. The people in the office believed, virtually unanimously at that time, that we were a not-for-profit company, that we weren't out to make lots of money for ourselves, that our main drive was to get people to use our quality software in the solution of their problems, that we were a support team. We had a serious research arm and we've talked about that already, and that has continued until the present day, but this idea of providing a Library Service, that's what we called it, has also continued to the present day. With the salesmen came the whole issue of commission and company cars. John Reeves, our first sales manager, was appointed in the late 70s, early 80s. All of this is proper commercialization, and one of the weaknesses I've had as Director is not recognizing early enough the vital importance of that business side. In the early years in particular, we were so successful that we weren't forced to get these business issues into proper focus and establish a strong balance sheet and save plenty of cash so that we always have the resources we need to carry forward the development that we wanted to carry through.

At the time of going independent, in 1980, we had a staff of about forty people .The outside world generally treated us as academics. So if I visited a university or ICI, I was treated as a university professor. What happened, of course, as we started being more commercial, is that we took on other spots. I can remember the first time one of my working collaborators said, "Well Brian you're the commercial person, the salesman. And of course I'm an academic. I'm a distinguished statistician, so I think I ought to go in and see Professor Finney and introduce myself. And if you can bring the stuff in, I'd be very grateful." This was at the Compstat meeting in Edinburgh, and David Finney, who was Chairman of the Computer Board, and one of the distinguished professors of statistics in Edinburgh University. And I will never forget, it's so reassuring. I thought to myself, "This is not a time to argue, just do it Brian." So I let the young chap go ahead of me, we were both young actually, and walked in carrying a printer, and various other materials. Finney went straight round the guy as he approached him and said, "Oh, Brian let me give you a hand carrying that, we don't treat professors from Oxford in that way here at Edinburgh." But of course I wasn't a full professor in Oxford with that title (I never professed) I had professorial status, but I was glad to be a member of Congregation, and involved in the University too.

HAIGH: Okay. So to rewind and look through issues in a little more detail. You said that the model that you had right from the beginning was that they would sign a legal agreement and they would be licensed to use the software for one year?

FORD: They signed a licence. This committed them in perpetuity, paying year by year.

HAIGH: Oh, okay. So they wouldn't have to sign a new agreement but presumably they could at some point terminate the agreement and stop paying and stop using the software?

FORD: Correct. If they terminated they had to issue us with a certificate saying that the software had been taken off all their systems and out of all the programs. And this meant for a university, it was extremely difficult, and it remains difficult now because they have heritage code, which has Library software in it.

HAIGH: So when a programmer compiled their program, parts of your code would be in that program, and they would have to destroy the programs in order to be able to cancel the licence?

FORD: Correct.

HAIGH: Did you have any cases where people tried to do that and you had to be convinced that they really had followed through.

FORD: No, by and large, perhaps immaturely or unwisely, we've treated people as honorable. And yes of course our software has been pirated, as we speak we know there are four to five hundred sites around Moscow using early version of the Library for which they've never paid. And software theft is something we all have to deal with and this is one of the reasons why we've gone into various security measures with software keys and software licensing, in recent years, to overcome that. And we've had a Director of a secure lab in the States, say he's using our software without paying for it, because he knew there was absolutely no way that we could force him to. But the big labs, obviously, behave responsibly. Lawrence Livermore started using our software in 1976, and also Oak Ridge. Argonne National Laboratory was a very early user of the Library as well in the States. I've mentioned already that Germany was a very fertile area for use of the Library, and at one point every university in Germany, France, the Netherlands, Sweden and Norway, was using the Library. Now many of them continue to use the Library, but, of course, as in this country, the United Kingdom, we have many institutions, which are now called universities which don't do any scientific research, or serious scientific or engineering computing.

So you can't make those collective claims anymore, but the number of sites, in academia, is effectively unchanged. At the moment, for example, use of the library in Japan is going very strongly. I have friendly acquaintances in Academgorodok Novosibirsk, who I know are using the library, too, from questions and comments they make to me from time to time, particularly on our New Year cards. They would like to put that whole thing on a proper legal basis so they can help support the group, but it isn't easy for them.

We have found that method of licensing is really quite effective. Our auditors tell us that we charge too little for the library service we offer. Currently one of our major growth areas is in financial markets and for economic and financial research, where the high quality of our algorithms is clearly essential.

HAIGH: So can you remember at the time was it obvious that this was the way that software should be sold, or did you have a feeling that it was something that you had to invent?

FORD: We basically invented it. We asked around and got some monstrous suggestions as to price. One computer manufacturer suggested we ought to sell the library to each site for at least ten thousand pounds. We were confident that people just wouldn't have been willing to pay that sort of money. Since we had an educational intent and a support intent, rather than an economic drive, it seemed important to us to have a licensing technique which was realistic but reasonably based and we found the model of people taking out annual subscription as with journals a very natural one. Now, independently, our friends at IMSL, in the States, came up with the similar solution, starting from their rather different position. I think this is indicative that it was probably the right way to go. That was one of the better decisions we made. Perhaps the best of all was to have the cash up front, now that was made purely by financial instinct. We could not have survived without the cash being paid up front because we had no other source, but I think that only became clear to us many years later.

HAIGH: And did the license cover a site or was it per machine?

FORD: Initially it was per machine, but then it evolved to this difficult concept of a site. If you think of much larger universities, for example, they always invariably have more than one campus that makes up the total university. Clearly, as time progressed, the concept of site had to keep pace with the changing nature of computing. Also, of course, universities very quickly had more than one computer. We had concepts like the University of London, with its fifty, sixty, different colleges, the preeminent college obviously Imperial College, and the first division University College, and Kings, and QMC. I guess the subtlety of that might be wasted on our North American audience –

HAIGH: They would certainly understand the idea of a state university that might have a dozen different campuses.

FORD: Absolutely. Let's take the concept with the University of California. So we have Berkeley, but we also have Lawrence Livermore, which is run by the University of California. So we had to be very careful when it came to recognition of a site separating out the actual degree conferring institution from the organizations for which it might be responsible. Some of those might be research labs and some of them the famous huge labs. But, yes, the concept of site was a very important one to us too.

HAIGH: And was there a fixed fee regardless of the size of the computer or the size of the site?

FORD: That was the next interesting activity. Early on it was basically a charge per computer, per processor. We soon had the CDC 6600, Seymour Cray's first machine, and then quickly after that the 7600. We had 7600s in the University of London, the University of Manchester, in Hanover, but also, of course, in Lawrence Livermore, and Sandia Livermore. The beginning of the supercomputers meant that we had to have differential rates for them. Our aim was always to get the supercomputer at the apex of a computing use triangle, because below the 7600 there'd usually be serious processors in institutions going to use the 7600. And soon, the workstations under that. So you had this enormous triangle of use and it was important not to overprice the 7600 libraries so that you properly pulled into licensing everybody underneath it. For example, in Italy there was the famous CINECA centre, outside Bologna, with its associated sixteen universities each with access to a single CDC 7600. We'd pick the 7600s off like plums off a tree all around Europe in academia, and that, of course, led us on into the Government laboratories as well. Big industry was using those facilities too. And we had companies like Phillips, who actually built their own computers. Initially we had to actually help them implement the NAG library on Phillips' own machines.

So from the beginning, not only did we have this rich structure of different sorts of computers on which we provided the Library under licence. We also needed a licensing framework to enable people to implement the Library on their own systems. For example, I have a file of the negotiation with the Soviet Academy of Sciences for the implementation of the NAG library on the BESM 6, which in the end didn't go ahead because of advice from Washington and London.

HAIGH: So it was tiered according to the size of the machine?

FORD: Tiered according to the size of the machine, but it was also licensing for different sorts of implementation.

HAIGH: Now originally the software was free to the UK academic centers?

FORD: Correct.

HAIGH: But after you incorporated did you start charging UK university computer centers, or was that still subsidized?

FORD: It actually wasn't subsidized, because the Computer Board for Universities and Research Council at that time was basically paying our salaries through the University of Oxford. But from the first of August 1980, when the grants from the Computer Board stopped, we had to reach an understanding whereby the UK universities licensed the software in the same way as other people. That caused the negotiation of a license for UK Universities, which has continued to this day (now through a body called Eduserv) and for other universities elsewhere in the world, and for the other institutions. The licensing base got progressively stronger, so that we had at least hundreds of thousands of users of the Library throughout the world by 1980. And we learned the business. Steve and I and colleagues, we learned the business bottom up. We learned about employment law, we learned about lease negotiations, we learned about setting up sales and marketing programs, having the appropriate staff, running an organization, as well as running the voluntary collaboration which continued.

HAIGH: Did you find that you enjoyed those kinds of managerial duties?

FORD: Yes. If I'd been intended to be a traditional academic I probably wouldn't have left Nottingham. I was able to keep up a strong academic interest with students and some teaching until the early 90s. But the challenge of creating an organization and running it was great. The main thing was being Director of NAG, the whole time I've been here: the nature of the company has changed regularly throughout that period. Sadly in the last five years I've had to downsize the organization and make staff redundancies obviously in collaboration with my senior colleagues, and having advised the Finance and General Purpose Committee. That has been a very maturing experience. I'm now basically an entrepreneur businessman with strong academic interests.

But in 1976 I thought seriously about moving on, and was tempted to go work in the United States, and indeed had offers there. The last big job offer I had was about eight months ago, which was to go and work in New York running a software company there in the financial area, for which they offered me a salary of three million dollars for two years work, as long as I cut all my connections immediately with NAG, and here in Europe, and committed myself solely to that role. That was a sacrifice that, after thought, I wasn't willing to make. It would have killed me anyway, I think. But it's been good here at NAG. There have been offers like that progressively throughout the period, keeping me on my toes. The biggest challenge of my job is discovering actually what's happening in IT, and in scientific computing, so that one can set the work program of the office, the planning of new products, the restructuring of our activity to reflect coming technology as effectively as possible, and that's been the same throughout. And remember, the IT industry is a baby. Arguably it's only fifty-four or fifty-five years old. It's always pleased me that Martin Luther was one of the first people to use a printing press for his teaching notes, when he was a university professor, whilst building the Protestant Reformation in Germany. And there are some legitimate comparisons. We've tried in NAG to make sure that the power of this evolving technology was available to the academic community, and the research community, as early and as effectively as possible. It's always been one of our objectives.

HAIGH: And returning to licensing. Can you give a sense of what the library would originally have cost, and how that would have changed over time?

FORD: My memory is that the early annual fee would have been about three hundred pounds a year for a mainframe. And a mainframe would now cost about sixteen hundred pounds a year. Hence the increase in price has not even kept up with inflation. What we've tried to do is charge what we believe the market would bear, and we may have been timid not least because we wanted as wide a use of the Library as possible.

HAIGH: And do you have a sense of whether that amount would have been the same or less than IMSL, for instance?

FORD: Once we started selling the Library, and recognizing that the Brits are just as good as the Americans, we've always taken a very keen interest in what our colleagues in IMSL did charge. And indeed in other software companies like IBM, and in the earlier days Digital. Obviously Digital has come and gone in that time, NAG is of course a very old company in the IT industry, so many, many companies have come and gone, and some have been reborn like Cray. We usually charged about the same level as IMSL. Sometimes we've charged less, there have been products for which we've charged more because we recognize that people pay for quality, and we knew our products were better than theirs. Of course the major market in scientific computing, to date, has always been the United States. Clearly with the immergence of China and India in the next fifty years that may change. And of course we had a subsidiary in the States since 1978, because of the strength of that market. But we've had a reasonably tough time in the States and never had the success there that we hoped for. IMSL got into the vast majority of North American sites first, most companies and universities chose to specialize on one Library and the US Government whilst it did some business with us, preferred to work with a US corporation. In the rest of the world, particularly Europe, we've always been able to at least compete effectively with IMSL, and often outsold them.

HAIGH: You think most sites would tend to have one library or the other?

FORD: Early on sites were inclined to specialize on one Library or the other. Latterly I think it's become much more common for people to have both in their collections of software for use. The challenge to IMSL and NAG has always been the people who roll their own, then the book Numerical Recipes, has had an impact too. We did think seriously of creating a numerical recipe series before the American one started, but decided it would be too dangerous for our quality model to do that. I think our view would be that decision was and is still correct, that because of the dated material coming out in organs of that kind it would adversely affect our quality standing as the premier provider. And of course one of the things we've been able to do, because of the vibrant nature of our organization, and because of the work of our many voluntary contributors , we've been able to maintain a liveliness in our products and services, that sadly our colleagues at IMSL have not been able to match. Therefore, I think, they've progressively been hit by those factors. We, ourselves, saw a less wide use in the late 90s, but there is strong evidence that the use level we have now has hardened and indeed is growing slowly. Together with that, we have this selling of technology licenses, and the run times, which are complementary. So we do see a period of progressive growth at present.

HAIGH: And would you say that the user base has shifted from ordinary scientists and engineers with a problem they need solve, to specialized software developers who'd be embedding parts of this technology in other systems?

FORD: That is a fascinating question. Currently about the third of our users are in academia, but that number in real terms is probably stationary, and in percentage terms is decreasing. The end-user licenses, to individual scientist or individual university computing centers, in general it don't work anymore. There are project machines for particular activities in universities, industry, government and commerce, and there are networks, and we have to be alive to all of these different models. There are still people who sit down and write their own programs, but progressively there are research groups, or development groups, in organizations which develop software, and they incorporate our components as part of that software. The research groups, obviously, are developing their own new systems, which they want to sell on. So in numerical terms the numbers are probably about the same but it's made up very differently. One of the major differences has been that the universities have stopped using FORTRAN, and Algol 60, Algol 68, to teach people programming languages. Today's students even seem to find writing MATLAB programs intellectually challenging. So we don't have the same taught base using the software. And this has created for us a real problem: how do we get today's scientists and engineers to know about NAG for incorporation in their software in the future? That's a problem where we're addressing at the current time. MATLAB has brought a very different paradigm and Cleve Moler and Jack Dongarra, his Ph.D. student who created it, have been very successful in licensing the software. Obviously Cleve is the Chief Scientist of the MathWorks, and there is a whole commercial company that supports it. We have to think very careful about whether we seek to complement that paradigm through the provision of added facilities that the users of MATLAB can use, or whether we seek to compete with it. And I think the answer is we've not reached a hard conclusion on that issue.

HAIGH: Have there been any developments that have made it easier to use the NAG library capabilities in a more interactive manner?

FORD: We certainly try to design the software so that it is as easy to use as possible. There have been various projects looking at computing environments. For example, our colleagues in Japan have recently placed a major order with the Defense Academy there, and for the Defense Academy we have developed our own easy to use interface so that the students at the Academy will be using the NAG library, and its various products, as part of their science education. We're still basically developing software components which can be incorporated into other people's programs in many different ways.

HAIGH: And returning to the sales methods that you had mentioned earlier. Now if I understood what you said, in the 70s you and the other senior staff were mostly responsible for selling but kept a very low profile because –

FORD: It was an academic activity. We'd be invited to go and give seminars in various places, and we would give purely technical seminars. The question would come from the audience "is the library available here", and we would say "Well, no, you haven't bought a license yet, but I'm sure you're going to." And we were invited to give talks all over the world, most particularly in the United States. A number of us would go every year to make presentations at SIAM or SIGNUM meetings, or specially called conferences. But clearly that isn't an effective way to proceed and we started going then to hardware user group meetings, to Government laboratory open days, and through the whole paraphernalia, culminating in going to CeBIT [***Ce****ntrum der* ***B****üro- und* ***I****nformations***t***echnik* an IT tradeshow] in Hanover. We've been to CeBIT for the last fifteen years, at least, and this is an opportunity not only to seek to sell products to new industries, and new users, but also a meeting place for existing users and giving them the confidence that the products and services are continuing to develop.

HAIGH: So then the idea would be to go out to places where prospective end users of the library might be congregated, and then hope that they would be sufficiently impressed with the presentation that they would ask their computer center people to buy it?

FORD: Correct. In most institutions there's a purchasing agent and a technical authority, and it's always been important to talk to both. But of course it took us some time to learn that, and the sorts of software we're selling was alien to experienced computer sales people, because they were used to selling big packages usually with big commission. Whereas we've always had a regime where the sales people get a certain level of commission for recurrent income to preserve the existing base, but then also commission for the new sales they make. And one of my jobs has been year by year to confirm with the sales and marketing manager the packages for his sales colleagues. So clearly, although we're a not-for-profit company, we've had to have all of the expertise and interest and involvement and all of the facets of a commercial company. When we've been fortunate enough to make surpluses we're able to tell our using community that the surplus was always ploughed back for either additional equipment, or for the development of new products, or for the appointment of new staff. And over the last thirty years there have been resources available to enable us to do that.

HAIGH: So when the professional sales force arrived they would continue to focus on users, rather than on people in the computer centers who might have direct purchasing authority?

FORD: They would undoubtedly go to the computing centers first, to see if they're interested. Then they would often help to find those people on site who were willing to act as champions for the product.

HAIGH: And I have one last question in this area, which is on the response to user feedback. Obviously you would receive the routines from contributors. Were there any kind of mechanisms in place to learn what it was that the users would like to have more of or would like to have changed, and to have that eventually feed back through routines into the Library?

FORD: There are two aspects of that that I'd like to deal with. If an error was found in the software of a contributor, the contributor would always learn what the perceived error was through our advisory desk. We've had strong advisory desks since the early 80s, built on the numerical advisory desk that I started in Nottingham in the late 60s, and from which the whole process evolved. I was always willing to receive correspondence, telephone calls, whatever, telling us about what users wanted. Once the network was created in the early 80s, we made sure that we had facilities here. As I'm sure I've mentioned earlier, NAG was one of the earliest users of the Internet in 1977, but that was only part of what was needed. We set up a NAG User Group so that the users could provide us with information on line as to what products and services they felt were necessary, ways they'd like to see the Libraries developed, and so on and so forth. That worked very effectively as a mechanism until the late 90s when it was rendered ineffective by the rise of the Web, and the support facilities and questionnaire capabilities it enabled us to provide. What we do do is put out questionnaires to our users asking what they think about changes to the services and also asking them what else they would like to see. We have places on the Web where people can leave us advice as to what they think we ought to be including, and make comments about the products and services that we do provide.

HAIGH: As you're largely reliant on the contributors to provide new functions, what would you do if there was something that the users appeared to be interested in having that the contributors didn't happen to want to write?

FORD: This is a very real issue. From the middle 1980s we've actually had people in the office who themselves developed software in a particular area, such as Jeremy Du Croz and Sven Hammarling, Sven of course having been the replacement of Jim Wilkinson at the National Physical Laboratory before he joined us here. We started developing our own in-house software in particular subject areas and making sure we had in-house people who were experts in those subject areas. Now the voluntary contributions never stopped, but it has meant there have been natural champions in the office of specific numerical areas. But, for example, in optimization theory, there is enormous world demand for new algorithms and the willingness of people to provide those algorithms on a purely voluntary basis has at best weakened. At worst that just doesn't exist anymore, and so we are having to look actively about making sure that we are keeping a thorough coverage of all the numerical areas and statistical areas that we need in the Library, sometimes going out to get software from specific groups, sometimes setting about writing it ourselves.

HAIGH: So could you give some rough percentage numbers for how the balance between in-house and contributed code in new functions added to the Library would have changed over the decades?

FORD: Initially, clearly, all of it was contributed voluntary, so at that point it was a hundred percent. At this point there are areas of the Library where we rely on internal contribution. Certain areas of parallel algorithms, we have to develop ourselves. What has often happened is that we have used grants from the EU to create specific bodies of software, through research, that we want to put in the Library. That has particularly happened with our Parallel Library and our SMP Library. We'll talk a little more about that later. But we now also have areas where people were still making contribution to the Library in our traditional model, for example, in ordinary differential equations and in large areas of Statistics.

[End of Tape 3, Side B] [Start of Tape 4, Side A]

FORD: I don't think there's one area now where the voluntary contribution would be one hundred per cent. My guess is that there would be subject areas where twenty, thirty per cent would be now be voluntary contribution. Overall it's probably around the thirty, forty per cent mark, but that is a broad estimate.

HAIGH: So that's been quite the important shift over time?

FORD: Yes. And it means that when we have had times of economic challenge, so that we've had to make redundancies, we've had to be very careful to make sure the technical needs of the organization have been properly protected as well as the commercial business.

HAIGH: We are now joined by Steve Hague, long time deputy director of NAG. He will be present for the discussion of NAG's developments of new product areas beyond the original library. But first I would like to invite him to take a few minutes to discuss his own role with the organization, beginning very early on in its history, and how that role has developed over time.

HAGUE: My name is Steve Hague. I joined the project known as the Nottingham Algorithms Group in September 1971, and my initial title was something like Library Coordinator, ICL 1906A so, as the title suggests, the focus was on adapting the early NAG software for a particular machine range considered of importance at that time. And I remember quite clearly that my job specification said that I'd be spending lots of time reading journals and studying algorithms. As we all know, often the reality of jobs is somewhat different from the job specs that supposedly describe them. I did start out by doing some of literature research but I think, as I worked with Brian (Ford) in those early days, we both began to realize that the task of producing the software library had a certain amount of research to it, certain amount of literature research, algorithm research and the like, but there was a huge amount of other knowledge needed in order to build, document, test and support a library that could be safely used by other people, bearing in mind that you would never have any contact with the vast majority of them. That's the strong learning process – about writing software "at arm's length" - that Brian and I in particular, and colleagues who joined us over the later years, began to realize. So my role was of coordinating the assembly of materials, software documentation. We quickly realized that there was interest in NAG software on other machine ranges, not just the particular ICL range, and so that began a long preoccupation for me and for others, Brian as well of course, in issues of portability. Not just portability, but preserving quality across machine ranges. That has been an abiding concern throughout the the thirty years that I have worked with Brian, during which time the organization has of course developed and taken on different facets. My primary role has been in the technical area: coordination, planning, management. But in any business like this, there is necessarily a strong connection between the technical and business sides. We're not an academic research organization, we don't have the luxury of studying the things that are necessarily of most interest to us, or the technically most glamorous subjects, whatever they may be. We survive by being a trading company, and so over the years I've become more and more involved in the management of the company, as Brian's deputy, involved in strategy discussions with Brian and commercial colleagues, with a general emphasis on the technical management side.

HAIGH: So would you distinguish your own responsibilities, areas of responsibility from Brian's primarily by saying that you've been more responsible for the technical management?

HAGUE: I think that's broadly been the case. But as you appreciate NAG has many relationships, it has its own internal resources and its also had many involvements with other parties. Generally I'd say those have been very successful involvements: standardization activities, and collaborations, funded projects, joint product development and the like. Brian and I have been both actively involved in the wider NAG. In terms of the management of internal technical resources, yes that's been primarily in my court. We set an overall plan according to the business objectives of the company, and then that plan has to be translated into a number of technical activities and nurtured through real life, shall we say. That's where I spend a lot of my time.

HAIGH: And have you also been responsible for working with the contributors?

HAGUE: Yes, quite actively. As we began to develop a process, in the early 70s, for interaction with contributors, we began to realize some of the subtleties and complexities of that relationship. My primary role in that was organizing the receipt of material, the processing of material, and the interaction with contributors as we began, shall we say, to flex our muscles a little as an emerging organisation. We began to make more changes to what was contributed, and that itself was an interesting evolution. So, yes, with Brian I have had a very active involvement with the contributor relationship. It's changed over the years because, for example, the introduction of software tools within the NAG organization has meant that a number of things that we used to do by hand, we now do mechanically, or indeed, we can equip the contributors to do it mechanically. So ideally the work arrives closer to the NAG house style. That's good for everybody because, with collaboratively developed software, the huge problem is synchronization: they have one version over here, we have one version over there, these guys change their version, we change our version and if you're not careful you quickly have chaos on your hands. It's therefore true to say that, as my NAG career has progressed, in overall terms, I have concentrated much more on the coordination of the software activity rather than so much on the algorithmic development, an area in which we have been privileged to work with quite a number of very capable and distinguished colleagues, both NAG staff members and voluntary collaborators.

HAIGH: Well if we now move to address the topic of NAG's products beyond the original core numerical software in the library. I will leave it to the two of you how to apportion answers between yourselves. I would suggest that we deal with each of the main groups of products together because I know that in some of the areas such as programming tools and compilers, and statistics, there have been several products over time, and I would suggest that we try and deal with the areas in roughly the order that they appeared. My impression is that statistics was something that was present, to some degree, in the original library and emerged into additional products later on, so I'd like you to start with that one.

FORD: Well first of all what was our strategy? Certainly with the libraries, that was our main interest. The tools grew because of our need of them, and came the activity forward. The statistics came because of the contact with Professor John Nelder who was the first FRS who was a computational statistician, and by common consent is a world class statistician. We've been lucky enough also to work with Professor Sir David Cox who had an even bigger reputation than John, and we were persuaded by John to sell statistical products that he'd developed in parallel with the library. He felt the library should not have too broad a coverage of statistics, that this wasn't the way to do it. And I have to say, with hindsight, that that was probably a mistake. We should have actually kept up the vigorous development of statistics within the library as well as developing the relationship with him.

Beyond the tools and the statistics my memory is, in the early 1980s, Steve and I began to recognize, with others, that the company was vulnerable to a certain downturn if the library lost its appeal. Now the amazing thing is the library has never lost its appeal, but we've always feared that it might. So there was a decision taken in the early 80s to broaden the product base of the company as an act of prudent safety. There was also a technical need for these things, which justified it within the organization, and more widely within the company, and within markets, but the fact is it was done almost as a deliberate act. One of the tough experiences we've had is that in the last few years we've had it pointed out to us vigorously by a review group, appointed by our bank, that we should go back and put our main emphasis on our original products and services which were the libraries and the software tools. The statistics we've spun off into a startup company. So, as I mentioned earlier, the first area we got into was graphics and that led us on into visualization, but I think Steve had a particularly cogent description about it a little earlier, so I'll hand on to him to describe that.

HAGUE: The issue was really how to represent results. The numerical software people can often look at a table of results and derive conclusions from that quite speedily, whereas, other people who may not be of that numerical persuasion may be much more appreciative of some sort of visual representation. There's a famous sayings about a picture being worth a thousand something or other…. So we were aware of the greater demand of our users to have some sort of visual representation of results. There were technological problems in producing portable graphical software but initially the more difficult issue was one of persuading people that there were feasible ways of solving those problems It was rather like the challenge of developing portable numerical softwarein the 1970s; when Brian and I spoke about surmounting that challenge, there were at that time very important and clever people who scoffed at us. Ten years on, in the early 80s, there was a similar reaction to the notion that you could have portable graphical software. And along with a very able and farsighted colleague of ours, Professor Ken Brodlie from Leeds University, the proposal was developed for a graphical library. There were emerging standards at the time, and as is often the case, the standards were imperfect. So what do you do with imperfect standards? You can either ignore the standards and have your own, or you adopt something that has only three of its four wheels on or something. In the end what we did, which has since become quite a common "middleware" approach to this kind of issue, was to introduce a layer above the standards so that, whatever the vagaries of particular standards might be, hopefully they were could be quarantined from the high level. The notion of "middleware" has of course since that time been taken for granted as part of the "armoury" of software engineering techniques. In the early 1980s, that was not so much the case, especially in the somewhat esoteric world of widely differing graphics and plotting devices, each of which tended to be regarded as a singular case. The NAG graphics effort in the late 1970s/early 1980s led to the development of a highly regarded graphics library, which incorporated good numerics, of course! When you're fitting a curve, there's a lot of approximation work in that, and so naturally we used NAG numerics and were able to show in quite a number of cases just how important it was to get the underlying numerics right. We should add that the NAG Graphics Library product had some excellent advisors and contributors (e.g. as well as Ken Brodlie, Professor Dr Christian Reinsch of Munich), and the Library is still in active use today; despite all the razzmatazz about three dimensional visualization and animation, and the quality of that work done over twenty-five years ago, still shines through.

HAIGH: Now is this the product the one that I see listed on the document, *Important Dates in NAG Activities*, it says 1982 release of NAG Graphic Supplement, Mark 1.

HAGUE: Yes, it was initially called the Supplement. And there were big debates over whether there should be a library in its own right, should it be accorded the status of a library? These seemed very important issues at the time –

FORD: The ultimate accolade.

HAGUE: And the outcome of the debates was that, as a mark of its (by then) proven quality, it did become a graphical library. In that sense it emerged not just as aid and adjunct to the main numerical library, but it was used in its own right.

HAIGH: Was this NAG's second major in-house product after the library itself?

HAGUE: I think that's true. Though we did contribute in various ways to the statistical products, they were essentially third party products. Whereas, a graphical library was in the NAG collaborative sense, a NAG-developed product.

HAIGH: Right. And how had it been decided that everything that had been produced in the main library earlier should be one product that wasn't sold in pieces? A user site would buy the whole library. Why was it obvious that this should not be treated the same way and that instead it made sense to have it as a separate product?

FORD: We did have subsets of the library for specific systems as they became available We had, as I remember Steve, a PC50 library in 1984?

HAGUE: Yes, that's right.

FORD: We had a workstation library, which was a subset of the FORTRAN library, in 1986, so there were subsets like that.

HAIGH: But those were just the platforms that couldn't handle the whole thing. Is that correct?

FORD: Well, that was the perceived view, in reality in the end of course, both the PC and the workstation were able to take the full library. It was an introduction to the market. What we did was numerical software. Graphics was somehow ephemeral. The fact that Christian Reinsch and Ken Brodlie had developed the graphics routine, one a world class linear algebraist, the other one a very excellent numerical analyst, was neither here nor there. There was that perceived view. And I think even now the visualization community is quite separate from the numerical analysis community and other parts of the scientific community.

HAGUE: I think also there were both commercial and technical reasons for subsetting or not subsetting, as the case may be. From the technical point of view, the reason to have a graphical supplement separate was because, as a product developer, you have to make certain assumptions about the host environment in which the product's going to be installed. With the graphical library, If you put statistical and numerical routines in a library into the graphical library, generally speaking, from a system or client point of view, there's no difference because you are placing further demands on that system. You can therefore deliver it to the user as one product, and you impose only one set of requirements on the user's installation. In the graphical library, though, there were additional considerations, or complexities, one might say. Because in those days there were six, seven, eight, nine different competing plotting packages, there were any number of plotting devices, all of which all had their own applied standards, the only way you could make any sense of this seething mess was to put some kind of layer on top and then build on top of that. So there was a good technical reason as well as commercial arguments for keeping the graphical library associated but separate from the numerical library. I'm sure that was the right decision at the time.

HAIGH: And did the graphical library deliver features which competing packages were not at that time able to match?

HAGUE: I think in some respects it did do better than competing packages, and we were able to demonstrate that. In particular, fitting curves is a matter of data approximation, and we made sure that we had good quality numerics. We could demonstrate that certain other packages didn't. Not all of them. I think it was probably the first successful example of portable graphical software at a subroutine level, so that users writing their programs could incorporate a NAG numerical routine, a NAG graphical routine, and there was a darned good chance that they could move that program, which now had both numerics and graphics, from one system to another. And in those days that was regarded as something very unusual. You take it for granted now, but in those days you, if you wanted to use graphics, you tended to became an adherent of *a* particular plotting package, which worked on particular devices on particular machines, and those were the machines to which you stuck. The graphical library, in its own small way, I think, helped to break down that barrier.

HAIGH: And was it made available for the full range of platforms that the library itself supported?

HAGUE: Not entirely, but certainly for all the principal ranges. Again, it had these different system requirements from the library. With the library, it was a question of machine arithmetic, compiler library, compiler. You had all those considerations for a graphical library, and you also had plotting package, plotting device. So it was not as widely implemented as the main library, but it was on all the key machine ranges of the time.

HAIGH: And within the graphical library what would you say was the balance of code between contributed and produced in-house?

HAGUE: Well I suppose initially it was close to a hundred percent to zero, but over the years -

FORD: Hundred percent voluntary –

HAGUE: Voluntary. Later, it became a much more of a balanced activity. Very often, code for later releases was jointly developed. As our NAG staff colleagues developed their own understanding of the various areas, there was a generally harmonious and mutually beneficial evolution in the relationship from total dependence on external contributors to much more of a relationship between equals. That's the same sort of phenomenon that we witnessed over the years in library contribution. Our colleagues here are totally accepted by specialists elsewhere in the world, from Argonne, from NPL, I'm sure Brian's mentioned these places. Our colleagues move very easily in their company, and I think it's sort of a mark of the evolution of NAG.

HAIGH: So you had implied, just before the tape started, that visualization emerged as a separate product category from what we've known to be thought of as graphics.

HAGUE: Well Brian might have a slightly different take on this than I have. I see it in part as this: when technology evolves, until the possibilities become apparent, it's only the visionaries who say that "one day it will be possible to have 3-D animation and you'll need this, that, and the other." Most of the rest of us look around with somewhat more skeptical eyes (a "I will believe it when I see it" attitude), and then there are the real conservatives who say "it's never going to happen". I think in that sense the advent of the first workstations, the Three Rivers PERQ and the early Apollos (I think there was an early IBM contender too) with their advanced graphical capabilities that began to alert and excite people to the possibilities of 3-D visualization, of visual modeling. These things are so much taken for granted now, and the laptop that you have has a screen that is immensely more capable than the biggest mainframe fifteen years ago. And so we along with other people began to recognize how important visualization could be, but it wasn't immediately apparent. We were still in a state of flux. Standards were still few and far between, and pretty unstable, and so it did take some time before we decided to choose what we thought was the right vehicle at the time. The vehicle that we chose was IRIS Explorer and –

FORD: Can I jump in for a second? I think there's another component as well, which I have mentioned earlier but it's worth drawing out. So with the workstations came in the early 80s, and then with that came this idea of problem solving environments, and looking at environments for solving problems. Indeed, we had a number of grants which we may talk about later, for looking at that issue. We ourselves were therefore aware of visualization and indeed heard that Silicon Graphics had a product called IRIS Explorer, that AVS had a product as well, that was looking specifically at visualization. IRIS Explorer was basically an environment into which had been put a visualization ability. Then one of our ex colleagues, who had been one of the early coordinators, who was working for SG came to us and said, "Look you guys are very experienced at doing implementation on many different computer configurations. Would you be interested in implementing IRIS Explorer for SG into different computing environments?", and we said "Yes, we'd look at that." And the thing was so successful and the place of visualization, as Steve's described it, had become sufficiently clear that we agreed, I think, in 1994?

HAGUE: I think so, yeah.

FORD: To 1993…. We released IRIS Explorer on non SG platforms and then took over the marketing and sales activity for IRIS Explorer as well.

HAGUE: And later on took over the development activity.

HAIGH: I should say for the benefit of the transcriber that's IRIS, which was the name of the Silicon Graphics workstation range I believe.

FORD: So that perhaps deals with visualization and graphics.

HAIGH: The question I'd have, following up on that, is roughly what proportion of the market would these products get and who were your main competitors?

FORD: Our main competitor was undoubtedly AVS. Latterly it's also been open source software from –

HAGUE: Well several places. There's IBM's Data Explorer which was broadly equivalent, in the same category as AVS and IRIS Explorer –

FORD: And that's now open source, and has been for three or four years.

HAGUE: Probably more than that actually. There are publicly funded activities like SciRun at the University of Utah.

FORD: I was thinking of a company in Albuquerque that we've had competition from, we'll add that to the tape later.

HAIGH: Right. I know IMSL came out with some graphical capabilities for their library, was this a strong contender?

FORD: Can I say I was never aware of them? [LAUGHTER]

HAGUE: I was aware in passing. I think it wasn't a sustained threat, shall we say.

FORD: And with regard to visualization in the current market place, AVS is the dominant player. IRIS Explorer was always fighting to gain a toehold there, which it was able to do. We have a significant installed base of people still using IRIS Explorer, but currently we're not carrying forward for the technical development of it, simply because the visualization market is changing. In part because of open source issues, in part because of aspects of the technology, the impact of games and the visualization in games, many of the factors like that.

HAIGH: Yes. And I have a follow up question on the statistics too. So I know your timeline says that it says that in 1975 you started distributing GLIM, and in 1976 you started distributing something called GENSTAT.

FORD: Yes, let me, if I may. Fisher had obviously invented modern statistics at Rothamsted in the early 20s. John Nelder in the early 70s who was then head of bio mathematics at Rothamsted, had set about developing this package GENSTAT for use within Rothamsted and more generally within what would now be the BBSRC, the Biotechnology and Biological Research Council, but in agriculture generally. This was a world leader, very idiosyncratically designed from a software point of view, supported by a manual that needed a Ph.D. in both English and statistics to be able to read, but, nevertheless, a world leader.

John wasn't able to influence the development of policy within Rothamsted in the way he would have liked. He was also president of the Royal Statistical Society, and I believe it was whilst he was president, or near that time, he developed a new statistical technique called Generalized Linear Modeling, which is basically what he got his FRS for. He developed a package of GLIM to be distributed by the Royal Statistical Society, apparently in part competition with the package GENSTAT which he'd developed within Rothamsted. Because he liked the look of us, and because he was a very eminent statistician, we agreed to distribute both GLIM and GENSTAT from NAG. We have continued to distribute GLIM to the present day, although now it's now a rather tired and old package. The generalized linear modeling facilities within GENSTAT, in its current release, are much more full and diverse than those in GLIM itself, but the international reputation of GLIM is such that people still wish to buy it.

GENSTAT itself has had a checkered history. It's continued to be used within agriculture, in the 90s it nearly died because there wasn't a PC version of it, and the whole of the statistical community, particularly in agriculture, had gone to PCs. But then a PC version was created and latterly we set up a joint venture company with Rothamsted Experimental Station called VSN, VisioN International, to market and sell GENSTAT, and to take forward the development of STABLE, an advanced statistical visualization system, a concept that we and partners investigated in a recent European Union-funded research project . John Nelder has just celebrated his 80th birthday, in great style, at Imperial College, where at eighty he's a working academic.

HAIGH: So these products then would be substantially more specialized than something like SPSS?

FORD: No, they address different statistical communities. SPSS was directed very much at social scientists. We were lucky enough to be involved with the people at the University of Chicago, when SPSS was started in the early 70s. Indeed, we helped them move off the University campus into private accommodations in the early 70s. GENSTAT has always been based very much in traditional statistics, which grew up in British agriculture. While SPSS was being commercialized progressively, and buying new product to broaden it and fatten it, GENSTAT has remained essentially the product of one research group. There's been strong relations with people in CSIRO in Australia, but even the early version of GENSTAT had an involvement there too. GENSTAT itself has been recently strengthened in areas like time series analysis with the coming inclusion of GARCH [General Autoregressive Conditional Heteroscedasticity] facilities, and also new techniques like kriging, which are not available in SPSS. SPSS has become much more of a data handling system, and is basically competing in the world market with SAS, which comes out of the Carolina triangle. GENSTAT is now trying to climb into that market, and at the current time SAS would have about sixty percent of the market, SPSS would have something like thirty percent of the market, and GENSTAT would have one percent of the market. So there is a requirement for general statistics but then there are those particularized interests which the different packages have.

HAIGH: Right. And would these statistical packages have worked like the libraries, being called from a FORTAN program, or were they more interactive?

FORD: No, they were very much stand alone packages. Basically you went into the system and you did your statistics, and so we called that a closed system. Whereas, with the libraries the individual routines were called by a FORTAN programmer and there was an open system. Now GENSTAT not only had a line editor, from its early inception, but it also had a command language to. So in many ways it has the characteristics of MATLAB, but where MATLAB is directed toward mathematics, and numerical mathematics in particular, GENSTAT was directed towards statistics. So there's a real opportunity there, which I guess none of us recognized. The interesting thing of course, was that GENSTAT needed implementing on many different machine ranges in the same way as the library, which is one of the basic reasons that we did it. John always argued that with these masses of data that the statisticians needed to look at, it was best to use a system rather than to use a library. What we've discovered, of course, is that people are not uni-functional in that way. What we should have done is made sure that the library was as broad as possible in statistics as it was in numerical mathematics, because the statistical market is significantly bigger than the numerical mathematics market world wide.

HAIGH: Yes. And I believe the IMSL market the IMSL library putting statistical functions on even footing with the numerical ones.

FORD: That certainly would be their publicity. I think one would have to say that in terms of the statistical functionality it was always measurably less than that you'd have got in something like SPSS, or in GENSTAT. But yes it was in the early days stronger than ours.

HAGUE: It was, and we actively took steps to correct that. I think we have, and I'd say in terms of coverage and modernity of material we've been ahead for some time.

HAIGH: So that position not to prioritize statistics in the main library has been reversed later on?

HAGUE: Over the years I think it has. I remember planning meetings ten years ago where we debated this issue in, in particular to the relation to our competition with ISML in those days. That was competition being fought out on at least two fronts; one of those was breadth of porting, (who was on the latest Cray first, etc.), but functionality was certainly another key area. We did take that conscious decision to beef up the statistical content, and with the library we've continued to do that, and, I think as Brian's indicated, we're very pleased. That's proven to be a sound decision because in a number of ways it's led to the use of library routines in places that we'd never originally anticipated they'd be used. It's opened up new business possibilities.

FORD: And we had some very strong discussions in the NAG Technical Policy Committee with John Nelder, as we've made those policy decisions and enforced them.

HAGUE: Yes. The argument from a technical point of view between the library and packages is that a package, because it is self contained, can have a lot of housekeeping within its boundaries, for example for shunting data around. Whereas, with a subroutine library, it's all more explicit and, generally speaking, whatever data you may have, you have to handle it explicitly and pass on to the next subroutine. So it was a fairly strong argument that was put for the use of packages but I think we were right to pursue our own way.

FORD: One of the things that was associated with that there was a period in our life where we looked at problem solving environments and we looked at environments generally and we had a number of European Union grants, FOCUS perhaps was one of the more striking ones, that actually considered very actively the different characteristics of open systems like ours and closed systems like the ones we've just been discussing, in the case of statistics, and studying where the weight of our activities should be.

[End of Tape 4, Side A] [Start of Tape 4, Side B]

HAIGH: And so we've covered statistics, graphics and visualization…

FORD: I think one of the other things we need to look at are software tools, because we had them initially for use within the project, as we always called NAG, itself, but we also recognized early on the need for our customers and users to have access to tools as well. And I've talked already about the Pretty Printer we got from Lloyd Fosdick and the master library file system that we developed ourselves, and some of the early tools for portability purposes. One of the great activities that came out of Argonne National Laboratory in the PACK series was TOOLPACK which was an activity in the development of software tools working with Lee Osterweil, with Wayne Cowell of course, Lee Osterweil, I guess Jim Boyle and Steve Feldman, was still involved to a degree.

HAGUE: They were the main players, along with Webb Miller.

FORD: In Toolpack, we developed a set of integrated tools, which had a strong emphasis on software metrification. The project was funded by, I think, the U.S. Department of Energy and the NSF, for several million dollars. On the basis of our proven track record of successful software collaborations, we were invited by those responsible in the US for TOOLPACK to become joint coordinators of the project and later, we agreed to make the fruits of the project available as a public domain productwhich t became the central pillar of a software tool service that, I think it's true to say Steve, we've maintained ever since.

HAGUE: Yes, we've been motivated to do that because of external interests, but also because of our developing internal needs. As machines change, as programming standards change, your software library is some extent like a castle built on shifting sand. If you don't take care of that shift, you'll sink, and so over all the years we've been involved in the activity, each year it never fails to surprise me that we keep coming up with some new needs in terms of software transforming or checking. . For example, we have to check something different whether it's thread safety or whether it's using tools to transform XML documentation, one way or another you do need this proficiency in tools and it's therefore very important that you have the core technology that you can adapt as and when a need arises. So, for example, with one of our collaborations is with a Canadian software company, Maplesoft, and this is to build product that sits between the Maple environment and the NAG library. The technology we're using to automate construction of that so-called "connector product" has required some new technology but has drawn very heavily on the existing body of Toolpack-derived technology. So you can keep replenishing that technology, and we never thought we would reap such dividends from those humble investments many years ago. If you'd said to me twenty years ago that you (NAG) would still be using derivatives of this technology that much later, I would have scoffed.

HAIGH: Yeah, so the dates we have here are the TOOLPACK project started in 1978, and then it says in 1985 release of TOOLPACK Service.

FORD: That's right. The product being developed within TOOLPACK, we made available world wide as a TOOLPACK Service.

HAIGH: Now these developments you've alluded to were they fed back into updated versions of TOOLPACK?

HAGUE: No, not as such, no. I think TOOLPACK itself was the end of line of that public domain-based development, and for some years afterwards those tools were used essentially in the form that they reached in the mid 1980s. We ourselves continued to adapt and develop TOOLPACK technology for internal purposes, and then recognized that what we were doing was of commercial exploitable and that led to the NAGWare range of Fortran programming support tools.

FORD: With the NAGWare family we took, soon afterwards, another major step because these were tools basically often manipulating source text. What happened soon after that is we then moved into compiler development. We had a gifted colleague, Malcolm Cohen, I've already mentioned the strong interest in FORTRAN in the community generally and represented in the company too, and Malcolm came along. Stu Feldman had written the world's first FORTRAN 77 compiler, when manufacturers said that wouldn't be possible. Well when the FORTRAN 90 standards work was finished, IBM and other manufacturers said that it was an impossible language to implement, to get a compiler for. So we were delighted when Malcolm Cohen came in one day and said "I've been thinking about it, I believe I can write a FORTRAN 90 compiler in eighteen months." And he did. He won a number of prizes for it and international recognition, there's no doubt about it that saved FORTRAN and also pushed it forward. But the compilation technology that came with that, I think you'll agree Steve, we've found many other uses for it as well.

HAGUE: Yeah. When you start this automation drive, you find yourself getting more and more complexity in the tools that we were writing to take one body of software and put it into different state. We soon discovered that we were delving into issues of a complexity that are similar to those in the internals of a compilation system, so this initiative to develop a full compilation system was not in that sense out of the blue, it was instead a case of pushing what we were doing already to the point where it would be significantly usable by other people. Having our own compiler has proven to a very useful internal tool, and it's particularly good on checking, when you're developing library software, and especially developing library software on many different platforms. We use our own compiler very intensively on our own code. The compiler can have different back ends, its standard back end is to generate C that you then present to a C compiler, but we can use it to generate other variant forms, and that's why it has turned out to be so useful, and I think it's going to remain that way. There is a very close liaison between the library developers and the tools people, and that's been a critical element in the successful evolution and use of this technology. As we try to respond even quicker to market changes, having the technology to transform, for example, the interface to the library, becomes even more important. You cannot spend months and months investigating how to do it and then building a team to do it, then developing the technology to do it, and then finding, maybe two years downstream, that you didn't do it quite right. You just don't have the luxury of that kind of time scale. You're talking here of weeks, or months at most, in which you have to establish the feasibility of an activity, and then if you press the go button, you have to have the confidence that you can do it within some small number of weeks afterwards. That's the nature of the business that we're now in.

FORD: One of the activities was with the tools is our involvement in MathML and OpenMath, which is the representation of mathematics on computing systems for visual presentation, and caught up with the ontologies of mathematics, they're some of our current activities. Their use of XML as Steve indicated, is actually increasingly extensive and vital for carrying our components into many different sorts of computing environments through a wrapper technology. The other thing that happened, in the early 90s, was a recognition that symbolic computing was of interest to us as well.

HAIGH: I have a couple of follow up questions on the tools. So was NAGWare a kind of brand name that was applied to these tool and compiler technologies or did it have a deeper meaning?

FORD: It's a brand name. It's a family. What we had for a period of years were a number of business development units, where we consciously looked after different product families with a business development unit looking after each family. NAGWare, we thought, and I guess we still do think, was a good name to broadly describe what we were seeking to do there.

HAIGH: And the time line mentioned in 1991, the release of the NAGWare f77 tools, was that the thing that built on the TOOLPACK technology and modernized it?

HAGUE: Yeah. I think a lot of code was actually rewritten, but the principles… we could not have done that rewriting, had it not been for the TOOLPACK experience, that's how we learned how to do it.

FORD: You notice the work in parallel during the same timeframe on NAGWare and the f90 compiler, which came out at that time. These two were very much complementary to each another, I think it's probable that neither would have been there but for the other.

HAIGH: Yes. And you're suggesting that the compiler saved FORTRAN. About what share of the market for FORTRAN compilers did it achieve?

FORD: The fact of the matter is that we generally only have had s few thousand sites using the compiler. It's used in many of the big labs, it serves a portability aid across different machine families, and it's the only compiler that has correctly implemented some of the more esoteric newer features of the FORTRAN language standards. .

HAIGH: Are there any major competitors left for non-bundled FORTAN compilers?

HAGUE: Yes there are. You raised the point about saving FORTRAN. That same point was made by a presenter at our meeting last week: at a time when the major vendors, particularly major hardware vendors like IBM, were hesitating and we suspect secretly hoping that the latest form of FORTRAN would at last die (so they could concentrate on fewer languages), the advent of the NAG compiler demonstrated that the language was indeed implementable. In that sense had it not emerged, it's quite likely that the major vendors would have quietly buried their FORTRAN plans, and say, okay now we're all C or C++. It's not clear from where else that example of implementability would have come, so it looks as if that really was quite a crucial watershed in the evolution of the language.

FORD: That's one of the accolades that Malcolm Cohen carries, I think. And the fact, as Steve said, that's not by us, but by an independent and extremely experienced senior executive of a major world computing company, just emphasized the point.

HAIGH: And Brian I think you were leading onto another product area.

FORD: Yes, I was going to talk about Axiom and symbolic computing. Steve, I have to admit at this point, this precise moment, I have no great memories to how we got into it.

HAGUE: Oh I do, The word "Axiom" may be etched on my heart! ☺… inside joke. Symbolic computing has always had a rather odd relationship with numeric computing and for many years. Rather like between statisticians and numerical analysts, there was virtually no dialogue, and these were separate tribes who occasionally had wars, and otherwise ignored each other.

FORD: But one or two distinguished individuals who tried to keep contact with both groups, so Vel Kahan, for example, was always active between the two, and another person I feel would be Jim Pool.

HAGUE: But, again, computing has a very curious history that is largely governed by the rate of advance of technology. There were pioneering symbolic computing systems, for example IBM's CAMAL package, which was so resource intensive that it used to take over the whole of a mainframe at Yorktown Heights for a day. It wasn't a very popular package, because everybody else would basically be taken off the machine on the day allocated to CAMAL, at the end of which, some profound symbolic result would be printed such as X plus X is 2X, or something. ☺ Or at least, that's what the critics and skeptics would claim! But there were also believers, and gradually as machines became more powerful, disks became bigger etc, , performing symbolic manipulation became more and more feasible. Through various collaborative relationships, particularly with James Davenport of University of Bath, we and others began to look at the connection between symbolic and numeric computing, and links between public domain packages like REDUCE and libraries such as NAG were investigated.

FORD: Through IFIP Working Group 2.5 I'd also been present at discussions involving the REDUCE system which I think came from AT&T, Murray Hill, New Jersey.

HAGUE: Originally, yes.

FORD: And also with the early Maple activity, in Waterloo, Morven Gentleman had been involved early in that, and one of the other co-founders of the system we had contact with in the earlier years was –

HAGUE: There was Gaston Gonnet…

FORD: I'm thinking of the Candian…

HAGUE: Keith Geddes

FORD: Keith Geddes, yes. So this fertile group we were aware of, and the wars between these package developers, and the way that within the centers in which these packages were developed and run they were hated by the rest of the people there because of their enormous consumption of resources. But then, I guess, in the early 90s… it's probably true to say that… well Mathematica was been created then as well, wouldn't it?

HAGUE: I couldn't remember exactly when it was first released.

HAIGH: I think it was available in the late 80s.

FORD: I think, then without being revisionist of history if we said that, as far as we were concerned, it was the early 90s that we became aware of Mathematica. Mathematica never joined the symbolic algebra wars, it always tried to remain slightly aloof of them.

HAGUE: So we did in fact have trial products, limited forays into the market place in the late 1980s, with links between REDUCE and the NAG library but we didn't persist with them. We retained a strong interest, and through the various contacts we had early contact with people at IBM, Yorktown Heights, who were developing what you might call the mother of all computer algebra systems, which became known as Axiom (it was called Scratchpad in those days). The team at IBM was led by a very farsighted person called Dick Jenks…

FORD: Delightful man.

HAGUE: …who had witnessed what had gone on with previous attempts to build computer packages, and had really thought through the design so that he really wanted an extensible design that actually matched the mathematics of algebra. So you had the kind of freedom of expression in terms of package that you could have if you were trying to express the idea mathematically. He had some very, very able colleagues, Stephen Watt and Barry Trager among others, and they had built a working system. It still was very very consumptive of resources, and so this prototype product had scarcely got outside Yorktown into the real world. But it was very impressive, incorporating concepts such as hyperlinked mathematical documentation that was years ahead of its time.

FORD: And we were approached with the possibility of collaborating with the people at Yorktown in commercializing the system because, IBM had decided that they could never meet their own commercial criteria for exploitation. So for a number of very happy years we actually sought to develop Axiom as a separate product from Mathematica and Maple.

HAIGH: And it shows here 1991, release of Axiom, Release 1. And then actually 1992 Axiom project begins NAG/University of Bath, funded by TCS.

FORD: TCS stands for Teaching Company Scheme, which was a Government funded activity in this country. And we worked to create a market for Axiom and to develop it in terms of collaborators and implementations as with the other products. It was a vainglorious attempt, intellectually interesting, but it never established a strong user base. Maple and Mathematica had too strong a foothold. From a business point of view, we commercially bowed out of it in the late 90s, but always maintained very warm and harmonious relations with the people at Yorktown, I was very sad when Dick Jenks died, perhaps a year ago.

HAGUE: I think there's some parallels with GENSTAT. GENSTAT is alive and well, and Axiom is actually still alive, but tacitly almost as an open domain activity for the remaining enthusiasts. It's almost an accident of timing that we began to gear up our involvement in this area, in the early 90s, when the PC revolution was gathering, and had such profound effects on computing, generally. When Brian and I went to Yorktown to visit IBM directors and managers, they and we were still thinking (even they, the people at the heart of the PC activity) of workstation implementations and talking terms of workstation pricing which is measured in thousands of pounds, not in tens or thousands of dollars. Somewhat as with GENSTAT, it took Axiom too long to "slim down" (that is, from mainframe to work station to PC, to the point where it comfortably fitted on an individual PC. We finally got a PC version out after a lot of effort by the summer of 1996, but in that crucial five to six year period the market, such as it was, for symbolic packages, particularly in PCs and particularly in education, had been largely "colonized" by a combination of Mathematica and Maple. There was no way, within the level of resources that we could sensibly commit, that we could muscle in. In that sense, we were too late in *the* crucial market for symbolic packages, which is academia and education. So if I look back on that period there were many achievements, it was a very good technical collaboration, technically an excellent product, but the timing, we can now see from history, was not that optimal. I think we were right to discontinue commercial activity with it, because we had other priorities.

FORD: If we'd have had much greater resources, or had been able to go to the market for a major investment to us enable to do that as a spin off, it might have been a different case. But even then, for the reasons Steve's described, it probably wouldn't have worked.

HAIGH: And was that possibility precluded by market conditions or by NAG's own structure?

FORD: By both I think. For the reasons Steve explained, it was going to be an uphill struggle anyway, but we ourselves just don't have the ability to call on the resources for that sort of investment in development and marketing. Can we stop at that point?

HAIGH: Alright. End of Session 2.

[End of Tape 4, Side A] [Start of Tape 4, Side B]

HAIGH: Ford interviewed by Haigh, Session 3. This is a continuation in the afternoon of the 30th of June 2004. Steven Hague has left, but Brian Ford remains.

So we've discussed, I think, most of the main other projects, products. I've seen the names of a number of other things listed on here, for example, the SLICOT Library, whatever that is, the Numerical PVM Library, the TextWare hyperdocumentation product, and the Finite Element Library. It's really up to you if you think any of those need to be commented on or if you would like to move on.

FORD: At various times we've developed subset products of one sort or another, and we've also taken on board offerings from collaborators to make more generally available. So there's a Finite Element Library that we supported and a Time Series Analysis Library. We've mentioned already the Workstation Library and the PC Library, we distributed the Sparse Matrix Library from Harwell, and the NPL Curve and Surface Fitting Library, and all of these have found an interest for people looking for specialized software of that kind. Also the SLICOT Library which is software in controlled engineering, which came from a collaboration with a group in the Netherlands.

In the applications we've made to the European Union, which starts with the Multi Annual Program, went on to ESPRIT 2 and ESPRIT 3, and then led to fourth and fifth frameworks. The fifth and six frameworks are currently being worked on. Through these programs we got grants either for developing special software that we required for either the Harwell library or the SMP library, or we looked at various sorts of computing environments. (the Focus project was a classical example of that), or we looked at combining elements of our product set which together could make something greater than the two halves. An instance there was the STABLE project, with the bringing together of Iris Explorer and GENSTAT. Steve mentioned earlier activities we'd had involving symbolic solvers being moved into computing environments.

Because we were successful principal investigators in the sense that many of the applications we made we got funded (our hit rate was around sixty or seventy per cent, and the average hit rate is around twenty per cent) we found ourselves quite popular in the community running projects of this kind. The danger is that it takes senior management time in the coordinating function, and you have to be careful to make sure that the products you're developing fit naturally within the product family of the company. And this has caused us to become more critical of projects we would apply for and be involved with. At the moment we have applied for a thing called Skypark which is a large project for the European grid involving some sixteen major research groups throughout Europe. I think the application value was five or six million Euros. And what we have found is that as the frameworks have gone forward, their requirement for primary software, such as algorithms, or library suites, or even middleware, have got tighter and broad multi- or inter-disciplinary activities with social focuses are much more prevalent. This means that we're finding European funding more difficult to achieve, whilst in the States we know our colleagues there are having more and more goulash made available for their consumption. Finally in the area of OpenMath and XML and languages and ontology studies of that kind, there is interest in Europe and it does look as if we going to be able to get some future funding.

HAIGH: But in general it's becoming less easy to use grants to develop new software packages?

FORD: Right.

HAIGH: So moving back to the development of NAG itself. We've alluded a couple of times previously to its international expansion starting, as I understand it, in 1978 with the formation of an American subsidiary.

FORD: Yes. NAG Inc., which started in the bedroom of Brian and Carol Smith in Downers Grove, Illinois. We chose it as the center because of its proximity to Argonne National Laboratory, which has been our strongest collaborator in North America by far, and progressively NAG Inc, moved from their home to its own accommodation to becoming a successful subsidiary employing some tens of colleagues (I think it's twelve and a half at the moment) and a turnover of two and half million dollars a year, which is modest in comparison with what one might hope for from the American economy, but, nevertheless, is regular and reliable. For our first distributorship we appointed Ceanet as an Australasian agent in 1985, and further distributorships in 1988 in the Netherlands, France, Spain, and so on. Currently we have some sixteen distributorships throughout the world. We have the parent NAG with subsidiaries in Chicago and Tokyo. We also have a separate company in Germany which doesn't have an office at the current time, we service the German market from the UK. We found the necessary language skills here in the UK and with the wide availability of the Web, etcetera, we can run the business as effectively from here as we can within Germany. We then have the distributors throughout the world, in Europe, specifically in the Benelux countries, in France, in Spain, in Italy, in Greece, in Malaysia, Hong Kong, China and Indiawhere you're relying on language skills being in the same time environment, understanding of the local culture to assist in the marketing activity.

Like many other organizations, we are finding distributors less effective as sales through the Web and Web stores become much stronger. We had our first Web site in the early 90s. We've renewed the site fundamentally twice since then and are in a redesign phase at the current time. In terms of documentation, we started with printing manuals in 1970, we then went through stages having online doc in the late 70s, microfiche, a fully enhanced set of documentation through a Web server, and an ability now to create information for the Web and print various forms of documentation all from the same base.

HAIGH: Did the products themselves require translation and localization?

FORD: I think we believe the answer is no. Perhaps if we had a Russian version, a German version, and a Mandarin version, we might sell more, but cost of producing them is prohibitive. English or American, American probably, is the international language of science and engineering.

HAIGH: And how successful have the German, French and Japanese offices been?

FORD: The German office… I don't believe we ever managed to completely absorb into the culture of southern Germany. The office we had in France was actually a sales office, and when our colleague there decided to retire we decided to have a distributor instead. The Japanese office has been successful every year except two years. It opened in 1994 selling IRIS Explorer. In 1996 it moved to selling library products as well and it became the Nihon NAG KK, and in that form has continued to do good business. In the year 2001/02, it had a difficult year in keeping with many organizations in the Japanese economy. But I'm glad to say it's bounced back to profitability, and culturally and time wise, if we're going to have any impact in Japan at all, we have to have a subsidiary there. We're extremely fortunate in having a fluent American speaker running the office there. He's native born Japanese, and it means communication is very straightforward. Culturally they have a different approach to service in Japan, and we've learned a lot from that and they also have great concerns about the protection and security of software and that's helped us worldwide as well.

HAIGH: Were there any Japanese competitors in the market?

FORD: IMSL had a thirty person distributorship there when we arrived. IMSL then went through the classic American process of sacking the distributor and buying the office, except for the Japanese leaders. For three or four years that appeared to work, but since then they've been in rapid decline. We've adopted a somewhat different approach of seeking to engage the Japanese organization in the NAG activity worldwide. We have a worldwide board, wwNAG, which meets four times a year, and although because of time separations it's difficult to hold a meeting involving all three offices (when it's 8pm in Japan, it's twelve noon here and 6 am in Chicago), but people do make sacrifices for the company, and it has meant we've been progressively able to move the company forward as a united body.

HAIGH: And has the geographical range of contributors been similarly diverse?

FORD: Yes. We've never actually managed as many Japanese contributors as we would have liked, but we have a small number. As we've recognized the constant refreshing of the contributor pool is important, and not easy because the people developing the new algorithms are very often younger people who are not as well known as their older peers, but we are seeking to carry that process through.

[End of Tape 4, Side B] [Start of Tape 5, Side A]

HAIGH: Ford interviewed by Haigh, Tape 5, this is a continuation of Session 3, in the afternoon of the 30th of June 2004.

With the international theme still fresh in our minds, I wonder if you mind talking about your involvement with SIGNUM, and with any other professional societies or groups other than the IFIP group that we've already discussed.

FORD: Yes, I was lucky enough because of my regular involvement with North American societies, ACM, SIAM, SIGNUM, and also being involved in many of the research conferences arranged by the U.S. Department of Energy, particularly by the Tri-labs, but also by Argonne, in being comparatively well known within the SIGNUM community. To my pleasure, I was elected as the first European member of the SIGNUM board, in fact the first non- American member of the SIGNUM board.

HAIGH: Do you know when that would have been?

FORD: It would have been in the 80s. I served for three years, which is the standard time, trying to broaden the appeal of SIGNUM, particularly adding a European component since there are many members of ACM in the UK, and in what you described this morning as continental Europe. And that was a very valuable contact, because it meant one had contact with universities of the middle rank, state universities, some of the smaller private universities, and also people working more generally in business. Not just the big manufacturers and pre-eminent financial institutions, which were the people I'd been inclined to meet up to that time. This was helpful in getting a better appreciation of the sort of software that those communities were looking for, and also influencing them in the adoption of international standards for language and for portability interests. Also for the development of defacto standards such as the Level 2, Level 3 BLAS and LAPACK and SCALAPACK which I know you're going to be talking with Sven about shortly.

I was also a member of the Council of the Institute of Mathematics and Its Applications, which is the comparable body to SIAM in the UK, and I served on its Council I think in all for six years, I think three years I was an elected member and three years I worked as a co-opted member on the Council and some of its subcommittees. This is a society obviously pursuing the interests of mathematics in the country as a whole, a meeting place for mathematicians of the special interest groups. With them and through them, we've arranged a number of conferences, the earliest being in the middle 70s, and the last one being on finance two or three years ago.

I've also been a member of the board of UKERNA. UKERNA is the not-for-profit company set up to oversee the academic network in the United Kingdom, JANET2, JANET3, JANET4, and had the pleasure in that capacity of working with Roger Needham as my fellow user representative on that board. I served on the UKERNA board for six years, and was delighted that when my period was up my position on the board was taken over by Richard Field, our current chairman. I was responsible for being on the subcommittee that appointed the last two chairmen of UKERNA, CEOs/chairmen.

I also serve and have served on a number of Government committees in the UK. The most interesting one currently is for scientific software for metrology, metrology's the study of measure. This is becoming fundamentally important throughout the world economy, and there were major activities in Europe and in North America around NIST in this area. We oversee as an advisory board the spending of some five or six million pounds, every three years, in the metrology area.

HAIGH: So which of the professional associations or groups would you say have been the most effective in dealing with the needs of the mathematical software community?

FORD: That's quite a difficult question to answer. I think that latterly, for communications and relationships and meetings within the community, SIAM has been the most successful. In earlier days ACM, through the SIGNUM working party, was particularly effective. ACM's continuing great contribution to our community is TOMS, the *Transactions On Mathematical Software*, and I'm currently in the process of chairing a review board to find the next editor in chief for the TOMS series. There's nothing in Europe which compares, in my view, with either of those bodies. I find this disappointing. Europe, in that sense, has still to find its own conscientiousness. We're basically a number of nation states when it comes to bodies of this kind. Being part of the English speaking world, the Anglo-Saxon world in the French model, is an enormous advantage, and I see myself very much as being mid-Atlantic in outlook, just glad to have a profound weekly exposure to the developments that are going on in the States.

One of the fun things in the twilight of my career with NAG has been setting up a number of startups and spin-offs. So I proposed the establishment of VSN, with Rothamsted Experimental Station, to exploit GENSTAT, and I hope a little later, our STABLE product. I've been involved with another startup called Telegnomic which was involved in real time data collection and data analysis. So, if you will, a company with a million and a half pigs being fattened in sheds throughout Canada and parts of the northern United States, is able to control that installed investment through reading fifty characteristics of the environments in which the individual groups of pigs are living and then use a syndrome recognition strategy which I developed with one of my colleagues, Jeff Morgan, to recognize when there is something in the environment which is putting the investment at risk. This is also being used as a technique in the oil industry, in waste management, and in utility management. Sadly, that particular company was forced into administrative receivership recently, through the non-performance of one of its shareholders, a massive Canadian multi-national, but I'm hopeful that the IP that was created can be effectively developed and exploited through another startup Helveta.

HAIGH: And a failed startup called OxDM was also mentioned earlier.

FORD: Yes, Oxford Data Mining This was a startup initiated with the University of Oxford's ISIS innovation organization, funded by the Oxford Onion which is the Oxford network for investment, and is the most successful IT investment group in Europe at the current time. Oxford itself, from the university, has seen the creation of many startups. Some thirty-seven of them have gone on to become publicly traded companies, several of them have made very substantial returns to the university through its shares. Sadly with Oxford Data Mining, whilst we had a good model, because of the nature of the investment that was being requested and the time in the market post the bursting of the Web bubble, and concerns about the Iraq war and the oil price, we decided that the model could be put into mothballs for a period and hopefully wait for a better day. Data mining itself has become a little overexposed, but business analytics seem to be all the rage, which are basically the same statistical ideas under another name.

HAIGH: Yes. Well that brings us around to the issues of what's distinctive about NAG as a kind of enterprise. One thing that I think you've referred to, in one way or another several times, is the distinct demands of running something which is not a company in the conventional for-profit sense, so that you can't issue bonds or make an IPO. Would you like to talk some more about the challenges that that's posed, and ways that you've learned to overcome them?

FORD: This undoubtedly has been NAG's biggest organizational problem, and remains so. On the one hand the organizational set up as a not-for-profit company encourages and engenders cooperation, collaboration, voluntary activity, the gifts of IP. On the other hand, whilst under British law a company limited by guarantee has the same status as a company limited by shares, in reality the financial institutions, in particular, don't look upon it as any way near as kindly. We did not have a major investment to start the company up, so basically we have had to create the contents of our own balance sheet. We have been able to create a situation where we have a balance sheet of several hundred thousand pounds, but clearly this only gives us the salaries looking forward for two or three months. This does not provide the resources for the creation of new products, or for major spending activities to launch of new product, and to purchase new equipment and take on further staff. One's abilities are very limited. We have been able to develop a line of credit with a local bank, HSBC, I believe the world's second largest bank, and we're very grateful for that. There have been times when it would have been very helpful to have been able to blow a couple of million on launching a new product, we've always had to find other ways of doing it, and indeed in the straightened times of three or four years ago, we had the bank effectively directing us to cut down our product staff. So the lack of an ability to draw investment has been continuously a challenge, and we have looked at the possibility of different company models to overcome this issue. For example, we thought of making the whole company a standard for-profit company, but we could only do that if we got the absolute agreement of all the existing members.

HAIGH: Of whom there are several hundred.

FORD: Correct. Some of whom are now in great age, and many of whom are philosophically opposed to the idea. Another idea was to have core NAG which dealt with the library products that would be not-for-profit and off it a number of satellite companies doing good business on a commercial basis. This model has been looked at. The issues are ones of linkage, because of the tracing back to the parent not-for-profit company. There were limitations on the subsidiaries under these circumstances as well. So our third model was basically to keep the company as it is but to think actively in terms of genuine spin-offs, some of them joint ventures, as a means of creating resource to fund the central organization. We've been looking at that model for the last two or three years and it's met with mixed success. The market in IT has been progressively downwards, the IT world market has shrunk over that period, and this has meant it's been a difficult time for getting startups successfully launched, and indeed funded. We have had some success with VSN and a management buy-in there last August, and we have hopes of the follow-on to Telegnomic achieving success as well, but it is a hard road. And I guess my conclusion, as I prepare for retirement in a month's time, is that this is a particular nut that I haven't cracked but I'll use all the experience and knowledge I have to help my followers, Rob and Steve, be successful in that regard.

HAIGH: Now as I understood your description of the management structure earlier on, my impression was that with the conversion to a limited company it still retained a rather large equivalent of a board of directors?

FORD: Yes but they're not exactly directors. So basically we have a Council of Management which now consists of two ex-officio members (myself and Steve), a representative from NAG Inc., who currently is Brian Smith, and then six directors either appointed by the AGM, or co-opted by the council itself. But, except for Steve and I, the whole board is non-executive and the decisions are made by myself, assisted by Steve, and my senior colleagues.

HAIGH: So the actual responsibility of management rests within a quite small group?

FORD: Correct. Having started NAG some thirty-four years ago I've been trusted, with Steve, through the Memorandum and Articles of Association in running the company and responding to a subcommittee of the Council, the Finance and General Purpose Committee, which meets once a quarter. Of course on major issues I consult my Chairman. Within the office I have Steve as my Deputy, three divisional managers in the current circumstances, and a business development unit which operates worldwide from this office. This is also the corporate headquarters. I mentioned, wwNAG, which is the meeting of the executive directors throughout the Group, so my fellow executive director in Japan, our president in NAG Inc., and Steve and I, with our advisors.

HAIGH: Now you've talked before about the idea of the NAG machine and the specific kind of cultural and methodological methods for actually producing the software among the contributors. I wonder if you'd like to say something about the organizational culture among the actual NAG employees?

FORD: In the office we have people from twenty countries. Our approach has always been to appoint the best people we can, whatever their cultural, national, academic background, and so we're a very cosmopolitan community. One cut would be the country of origin: the largest group of our employees are inevitably from England, but we have Scots, Welsh, Irish, and people from seventeen other countries. For example, I think we have two people from Iran, and one person from Iraq. To cut through the cake another way: the largest group of colleagues in the office are agnostic, we have a number of atheists, a small number of active Christians, three practicing Muslims who have prayers five times a day in their individual rooms, and a Jew and a Zen Buddhist. And it's quite usual in an office like ours in Europe to be aware of factors of that kind and show mutual consideration and concern, for example, for our Muslim colleagues during Ramadan (for the Christian colleagues the observance of Easter and Christmas is not made that difficult because they are national holidays). In terms of sex mix, we have about twenty-five women and thirty-five men.

HAIGH: So you have about sixty employees here?

FORD: We have at the moment sixty employees here. Demographically, we now are a mature community, having operated for thirty-four years. So that we have young people entering the company (for example, at the moment we have a young sixteen year old with us on work experience from one of the local schools), but we also have people retiring having completed a full career working for the company. We are a multiracial, multicultural community, which throughout the whole of the last thirty-four years has never had a major problem with any issues that have arisen politically, or in sport, or in any other matter on the world stage. We're also a caring community, our legal advisors often approach us for a datum on particular difficult employment issues they have to see how we would respond to them since they regard us as being very sensitive and culturally aware. The most challenging experience we ever had is an attempt by one of our colleagues to commit suicide, but fortunately we were able to save him and I'm delighted to say due to the caring and support he received within the office he's still working with us now, some five years later, without embarrassment at all, or loss of confidence.

HAIGH: Did the lack of share options and financial limitations pose a problem with employee recruitment and motivation during the boom years?

FORD: I don't believe so. My younger colleagues, the post-Thatcherite children, are much more pecuniarily minded than the establishers of the company, but they've put up with the old wrinklees very well. They have, from time to time, expressed the desire to have share options schemes and the like. To date we've argued against those, and they're still working with us. But in accepting the not-for-profit culture, I think they also recognized when they came that they were also taking on-board sound salaries, good holidays, an excellent pension scheme, but no direct benefits in the business successes of the company.

HAIGH: And is seemed that your personal style is quite socially oriented. I was wondering if there is a connection between this and the fact that the organization is very reliant on contributors and is non-profit?

FORD: I think it's all of a piece. I had a model of relationship and the academic community, which I learned from Jim Wilkinson, and this was fostered and strengthened by my relationships with groups like those at Argonne, those at the National Physical Laboratory. I'm glad to say even now as a philosophy it seems to work successfully.

On the question of the staff by the way, I should have said there is a staff panel, that meets four or five times a year, which discusses every aspect of the life of the company and through that and other mechanisms people are able to influence our structuring and our policy.

HAIGH: Continuing Session 3. Brian Ford has now left temporarily and I'm joined by Sven Hammarling. Sven, I wonder if you could begin by describing in general your involvement with NAG.

HAMMARLING: Okay. I started, I think, as a member of NAG in about 1974 when I was at what was then Middlesex Polytechnic. I contributed software in those days and principally routines like singular value decomposition, which were actually routines that weren't in the Handbook originally. After a time at Middlesex Polytechnic I wanted to do more with research and get perhaps a bit more involved in thinking about software and numerical issues, and I was offered a three year contract at the National Physical Laboratory as a principal research fellow. Sadly, at that time Jim had retired, although he still came in frequently. I looked after his small group at NPL when he left, and we developed a linear algebra library there and continued contributing to the NAG library. We used to have linear algebra meetings at NPL, where Brian and colleagues would come, Jim Wilkinson, would often attend them. Then when that contract finished Brian kindly offered me a job here at NAG and that's where I've been for the last, I think, twenty-two years, and I've enjoyed it very much.

HAIGH: Since then you've been a full time member of the NAG staff?

HAMMARLING: Yes.

HAIGH: Having begun as an outside contributor?

HAMMARLING: Yes.

HAIGH: And what have your responsibilities been here?

HAMMARLING: Oh, varied. When I started off, I think, I was called algorithms coordinator, or something. I became a manger of the numerical library's division for quite a time, but I wanted to get back to technical work and so became a principal consultant. For a short time I was even company secretary, as the previous company secretary left and they needed some help doing that. I had also a year's sabbatical at the University of Tennessee with Jack Dongarra, which was really good experience. Currently I'm still principal consultant, and also have a visiting position at Shrivenham which is a campus of Cranfield University. Now that's a potted history, and as divisional manager of numerical library's division, of course, I have been responsible for the development of the various NAG libraries and any projects that we were involved with. For example, we had a number of European projects which we have helped us with those developments. Principally my interest all the way through has been with technical side of software development, and the numerical aspects of software development.

HAIGH: Yes. So how close would you say ties remained between people doing cutting edge research in numerical analysis itself and these kinds of mainstream widely used libraries such as NAG? Clearly, in the early days Wilkinson's code was an embodiment of new and cutting edge kinds of algorithms –

HAMMARLING: Yes, and he was very influential.

HAIGH: So has that same process continued into the 80s and 90s with important new algorithms rapidly making their way into general use?

HAMMARLING: I'm saying not as it has done in the past, and sadly in academic circles it's hard really to get recognition for software development. I think that that's meant that a lot of people haven't put the effort that used to be put into turning their algorithms into useful software. Of course, there's a lot of research codes around, but they're often not polished codes that one could safely use in a production sort of environment. There are exceptions and in recent years I think LAPACK is perhaps the most outstanding example, where a lot of state of the art algorithms have been included in LAPACK. There are other software packages, so don't misunderstand.

HAIGH: No, no. Would you like to talk more about LAPACK then: how you came to be involved with it, NAG's contribution to it, and perhaps ways in which it's fed back into NAG's own products?

HAMMARLING: It's perhaps appropriate to start a little bit further back with the BLAS. Of course NAG's always tried to port its libraries to whatever machine users want the library on, and in the late 70s the Cray machine came along, and people of course started using it, and so people here decided well we better port the library to this machine. And when it was first ported, the performance was absolutely terrible on that machine, and people like Jeremy Du Croz said, "Well we had better look at why we're not getting performance out of it," and thought of some of the issues involved.

HAIGH: Was that because of the vector architecture?

HAMMARLING: Yes, yes, just so. And so we weren't in any way making use of the vector registers on the Cray machine. And so Jeremy thought, "Well if we had one or two computational kernels, particularly in linear algebra, and we could tune those, build software on those that would really help." Of course a number of other people had the same sort of idea, notably, of course, Jack Dongarra. And when those ideas were coming to fruition, I knew Jeremy and Jack by then, so a number of us got together and that led to the Level 2 BLAS activity. Very soon after that it was realized, once again, those really aren't at the right level of granularity for what were then becoming the higher performance machines, machines with hierarchies of memory and small numbers of processors. So that led then to the development of the Level 3 BLAS. So the Level 1 BLAS, of course, were driven by considerations of efficiency. Perhaps it was an even stronger drive with the Level 2 and Level 3 BLAS, but hopefully we stuck to the same principles of development as for the Level 1 BLAS. And having really got those as standards, people began to accept that one could develop software based on those computational kernels. That led people to think about development of a linear algebra software library that would make as much use as possible of those kernels.

HAIGH: Did NAG itself use any of the BLAS technology in its libraries or other products?

HAMMARLING: Oh very much, yes, yes.

HAIGH: At what point did that begin entering the product line?

HAMMARLING: Really as soon as the BLAS were standardized, and Jeremy and myself were involved in their development. I can't remember the exact date, but we put them in quite pretty early into the library, and encouraged vendors to implement efficient versions so that we could link to those when those were available.

HAIGH: The Level 2 and Level 3 BLAS?

HAMMARLING: Yes, we already used the Level 1 BLAS as well.

HAIGH: And that had been in essence the early 70s for that, when did that come in?

HAMMARLING: Well they weren't standardized actually till the late 70s, I think the TOMS paper was 1979. So this was some time after that, you started finding them more regularly in software.

HAIGH: The existing library code was reworked in subsequent versions to make use of those?

HAMMARLING: Yes, yes. And I must say Jeremy Du Croz was one person who did a tremendous amount of work in doing that. And this led Jack to putting together a proposal to NSF for LAPACK. Jeremy and myself were named as principal investigators, I think was the term used, although of course not being American we couldn't be directly funded, but we were at least named in the original proposal. As I'm sure you're well aware, from Jack there were many LAPACK meetings, working parties, and for me it was a project that I'm delighted to have been involved in. I probably got most satisfaction from LAPACK of any project that I've been involved with. And many many people contributed to it, there were eleven authors, I think, of the user guide.

HAIGH: Were your contributions underwritten by NAG?

HAMMARLING: Yes, yes. I mean NAG encouraged the collaboration in the project, so I think they had a really good view of that and very forward looking view of it all. So we were much involved in that and of course one of the motivations was to be able to incorporate the software in the library. And hopefully we could influence the design of the software, it would help us to understand the algorithms that are being used in the software. Over a period we incorporated LAPACK into our library and it's now a fundamental part of the NAG library, and we've reworked a number of other areas so that they use LAPACK instead of old linear algebra routines,.

HAIGH: So pieces of the LAPACK code have been extracted and rewritten and incorporated into the library?

HAMMARLING: I wouldn't say we've done a lot of rewriting, what we've done is a lot of additional testing of the software. If we find bugs of course we correct our versions but we try and be sure that is reflected in the LAPACK software that's available generally as well.

HAIGH: Actually, one interesting topic there would be: is there a pressure from users for backward compatibility between different versions of the library? So do you have to keep function calls and names the same even if you're changing the technology that's underneath them?

HAMMARLING: We wouldn't keep the same name and change the interface to the routines, so if that happens it gets introduced into the library as a new routine. But, of course, yes people do want that. As you said, backward compatibility.

HAIGH: So is that because the new routine might have subtly different mathematical properties?

HAMMARLING: Yes, yes. And, for example, one of the important features of LAPACK is that it gives you information about the quality of the computed solution. You can get condition number estimates from the software, you can get error bounds from the software, and much of the older software didn't have those capabilities, so that's new features in the software. The underlying algorithm may not have been substantially different, but it's added features to it, and so that meant that LAPACK couldn't use identical interfaces to LINPACK, for example, which I'm sure a lot of people would have liked if it had been possible.

HAIGH: And I know that NAG has been involved with a large number of other collaborative research projects. Have any of those played a similarly important role in improving NAG's product line or bringing new technologies into the firm?

HAMMARLING: I mean that's certainly the case. As I've mentioned, we've had European projects and we had one, for example, that was significant in the development of our parallel library, a project called PINEAPL. There've been other projects, such as TOOLPACK, which have led to the development of our own software tools and those are vital to us in the development of the library, we use our own tools in-house, and a lot of the motivation for their development has been our own library development.

HAIGH: Yes, Brian spoke about the TOOLPACK project. So how about those challenges posed by parallel systems, have the main stream library products been able to deal successfully with those?

HAMMARLING: I'd say with shared memory machines we've been pretty successful. LAPACK, itself, if there are suitable BLAS available, will run efficiently on such machines, so you automatically get that benefit by using LAPACK and the BLAS. And, in addition, we've done further tuning of a number of LAPACK routines and other routines in other areas to take advantage of shared memory machines. When you come on to massively parallel machines, you have this parallel library that I mentioned, which was lot of the motivation for the PINEAPL project, but that's a much smaller subset of the library, and that's much more challenging to think about converting more of the routines to run efficiently in those systems.

HAIGH: And is that a separate product?

HAMMARLING: The parallel library is a separate product, yes. The library we have for shared memory machines, we market as a separate product but it has exactly the same functionality as the main library but with these added enhancements of tuned versions of a number of the routines.

HAIGH: And something else Brian mentioned was the relationship with Maple and some kind of technical connector between the two products. Have you been involved with that?

HAMMARLING: I haven't been directly involved with that but have, of course, been involved in discussions internally with the people who are doing the work. And of course that's in some ways more a tool type work in connecting the NAG library and the Maple systems together, I certainly haven't had much direct involvement in that. And in the past we've also collaborated with the MathWorks, we have something called a Foundation Toolbox, which is a subset of our library but has MATLAB interfaces for it.

HAIGH: And where would you see then the current challenges that the library has to address, hopefully with the aid of improved technologies and better research?

HAMMARLING: I think from the point of view of making it attractive to users we have to, and we are, with things like Maple, involved in thinking how the users want to use this software. Because less and less do people want to program, they want to be involved in their science and be able to use software in ways that are relevant to their particular problems. So I think we have to encourage people in developing packages to make use of good quality software inside those packages, not just write their own numerical software. And we have to think about what sort of interfaces do people want, so we can continue development in our, perhaps, our traditional way but have to do it in such a way that we can readily, for example, make a Java interface, make an Excel interface, or make a MATLAB interface.

HAIGH: More of a component model.

HAMMARLING: More of a component model, yes.

HAIGH: So in that sense then you'd see the challenges as being more new ways of packaging the mathematical functionality than actually producing new area of mathematical functionality?

HAMMARLING: Yes, yes. I mean I think of course we have to keep up to date, we have to keep including state of the art algorithms, but yes it's that sort of interface technology that we need to think about, and indeed we are thinking about very much at the moment.

[End of Tape 5, Side A] [Start of Tape 5, Side B]

HAIGH: Do you have anything else to say about your personal contribution to the BLAS or to LAPACK?

HAMMARLING: I doubt if I do. For me one of the most important features was that they were very much collaborative projects. People really, I think, worked very well together and hopefully we all made very significant contributions to it. For the BLAS there were much smaller numbers involved, so we would have discussions and we would go off, and principally I wrote the vanilla software, so somebody else wrote the test software, and we would divide the work up like that. With LAPACK, even though it was a much larger project, which many more people involved, I still think it worked very well the way we divided the work up.

One person I'd like to pay tribute to is Jeremy Du Croz since he retired sadly through ill health a little while ago. He's been vital to the quality of much software that's around, of course particularly in the NAG library, quality both of the software and the documentation. And I think one can see that very much also in the LAPACK project, and I'm sure Jack and Jim Demmel would agree, that his contribution, was vital to the quality of both the software and the documentation, he was always so careful. I have in my office downstairs just a wonderful set of notes of his, and Jack and I keep talking about how can we somehow get these notes published, because they really give so much background to the LAPACK software and its development. Maybe we'll find a way at some point.

HAIGH: Well if you can't get them published, one thing you could do is to donate them to an archive, and then it they would certainly be available to researchers.

HAMMARLING: Yes, yes. I'll be seeing Jack shortly so I'll perhaps discuss that with him, that's a very good idea.

HAIGH: There's probably a lot of other material related to the project that could be preserved.

HAMMARLING: I'm sure that's right, yes.

HAIGH: Another thing that you might know about, are there any people who you think have, as contributors, made very substantial contributions to the NAG library who might not be as well known as some of the other people we've been talking about?

HAMMARLING: I'm sure that's true, it probably would be unfair of me because I'll no doubt pick out a particular individual. I can think, for example, of Ian Gladwell who was at the University of Manchester for quite a time, we've had a long tradition of collaboration with Manchester. He's now at Southern Methodist University in Texas, and he was very influential in our ODE software, contributed much of the software in that library. And there have been a number of people like him who've made those sort of contributions to particular areas of the library, and I'm sure you'll have heard of Gill, Murray, and Saunders, who were a group at Stanford University. Gill's now at San Diego. Gill and Murray started off at the National Physical Laboratory, contributed most of the optimization software to the library, and have continued that contribution to the library, so they've been extremely important to the development of the NAG library. For example, in the quadrature area, people were involved in the QUADPACK project, you may have come across QUADPACK, and they also contributed to the NAG library. As you'll gather I haven't got my thoughts together properly there, but that's just a few examples anyway, people who have contributed.

HAIGH: Well unless you have any other thoughts.

HAMMARLING: No, no, it's a good place to stop.

HAIGH: Thank you for taking part.

Sven Hammarling has now left, Brian Ford has returned and we are moving to conclude the interview.

So one topic that you've alluded to a couple of times, and we should probably spend some time on, is open source software. Now it's apparent that NAG itself has some of the elements of the open source software model, particularly the fact that especially in the early days most of its source code was coming from unpaid voluntary contributors who just wanted to see their work get out into the world and be used. On the other hand the software itself was never open source in the most literal sense, and except in the very early days, users have had to pay for the software.

FORD: I understand broadly, I think, the question you're asking me about. We obviously had a contributed library, which was the source code from which we created the NAG Library. The source text for that library was made available to academic sites, and indeed all our early sites, so they could inspect the code if they wished to, see what algorithm was being used, and to check it in cases they thought there were mistakes in the code. But they weren't given it in a machine based form so they couldn't use that as a basic source for further development. And the reason for doing that was to avoid the claims that had been made against the SHARE library and the *Communication of the ACM* library, for example that this was SHARE code and it was wrong. What one learned was that it wasn't the SHARE code at all, it was a code based on the SHARE code that had been modified and it was the modified version that wasn't working correctly. One of my most amusing experiences, in the 70s and early 80s, was to look at a major molecular quantum chemistry code that was used throughout the community, and discover there was a code in it called Jacobi, and when I actually looked at the code I found that someone had substituted the QL algorithm, which was the best algorithm solving the eigenvalue problem, but had not changed the name of the routine so that the people who knew the ancient literature of the eigenvalue problem felt confident, but they were actually using a much more up-to-date algorithm.

For security reasons, for the integrity of the library, we didn't make the source code available. Now Stallman came along with his new philosophy, where ostensibly he has an open source policy. In reality of course it is only open source if you are an undergraduate or post graduate in the university doing pure academic research, which you only intended ever to publish within the academic community within which you were working. If it was to ever be used by anyone for any production type purposes or by research groups that meant that the code went out into "industry" or "government", you of course had to pay for it. Stallman, of course, made his money from that philosophy by selling services, and this is one of the reasons why there was such concern, for example, within the U.S. Department of Energy regarding the GNU philosophy, as it was then called, as it was spreading. The open source approach, as I understand it now, with a package like

LINUX, for example, is much more based on the public domain philosophy which underlay EISPACK , FUNPACK, and Linpack, which meant that people simply had the code in front of them and they could do what they wanted with it without limitation.

HAIGH: That would be the BSD version of open source. I think much of the LINUX related software would be like that, but with the proviso that if you took this and made improvements to it then the improvements had to be released back into the public domain…

FORD: Yes, this is absolutely correct.

HAIGH: Which would be the difference, and clearly historically people didn't think about it in that radical kind of way. But the modern open source partisan might, if you take code from unpaid contributors and improve it, see you as being a little bit wicked for not freely releasing the improved code back again.

FORD: We have a relationship with all our contributors, which underpins the basis on which they've made the code available to us. In the main they continue to work with us as contributors, so they receive notice of the improvements that have been made and indeed copies of it, processed through our response center. The BSD strategy, of course, was initiated by Jim Pool while he was working for the U.S. Department of Energy, and I'm sure you'll want to mention that, or discuss aspects of that with Jim when you see him later during your visit.

I take some of these fine distinctions. I think the LINUX activity has sprung from a coming together of very substantial interest groups amongst computer manufacturers, and ISVs, and computer users, concerned about the total dominance of Microsoft and the Windows system. It clearly left people making choices between some distinct approaches. Open source seems to have worked quite well, with major investments by the people using the open source material to create operating systems for their own use. The great concern in the industry, and amongst the numerical software community, is that open source doesn't encourage investment in any systematic way in the development of products and new services. It weakens, and in some instances can destroy, specific markets where there is need for development and research. But people don't make the software simply because of the loss of so much of the market, which normally would provide income to enable the development of newer versions and new services to be made.

And because of this we have found that there are people in major Government programs in Europe and, particularly North America, who are concerned about the growing dependence on open source mechanisms, and the inability that leaves organizations with, having developed software products and services through their grants, to be able to maintain their software in use. And indeed if you look at recent offerings made from the Tri-labs and from other major Government facilities, for service in the States, you'll find there are many projects now seeking to fund people to run maintenance and support services on software which has been developed in this way. So on the face of it, it is a help within the academic and Government community. The concern is that unless there is a strong organization with deep resources which is maintaining the open source code and using modern software engineering techniques to maintain it, the product will run into the sand, lose quality.

HAIGH: So I wonder if you have any thoughts on why the area of scientific software packages seems to have developed in almost opposite way to operating systems. In the mid to early 70s, from what've you've said, the NAG library was almost one hundred percent contributed code and the various PACKs were freely available. Today you've said that the NAG library has relied less and less on contributed code, MATLAB, Mathematica, and other commercial programs have a very large market share among ordinary users. Whereas operating systems have moved from a world of completely proprietary closed systems in the 1970s to one where Linux is now becoming an extremely important enterprise platform.

FORD: I take the question. I don't think there's any easy answer. As far as we're concerned as NAG with the contribution base it's still substantial. It is a question of recognizing new talent, developing new relationships, people being free to work on the offered basis. Of course it's clear that with the Web people have a very strong personally based and driven mechanism they can use to make their information and their software available should they choose to go that way.

HAIGH: So they would have less need to use the Library as a distribution mechanism?

FORD: Correct. On the other hand there are still, fortunately, many numerical analysts and computational statisticians who seek to have their library software, available through that mechanism. Some sort of stamp of approval or mark of quality, and so we have people regularly approaching us offering us code. Sometimes on the old basis, and sometimes offering to make the code available through the mechanism of the library if they get a small royalty for each sale. The weakness of accepting such a model for NAG, I think is clear. At no point have we paid for the capital that has been put in the library, although intellectual property value has been generated. If we start paying everybody who offers us or has given us code to put in the library, with the modest earnings that come from these sorts of services we'd have a non-supportable activity. So that is difficult.

To address the difference between areas of scientific software and the operating system, the one thing they seem to have in common is that both academia and the government labs of which I'm talking, are looking for open source solutions at the current time, that's the preferred mechanism. When it comes to our own software, organizations which are funding Linux activities for their own computing requirements, finance houses, banks, insurance companies, many of whom have been experimenting with Linux, those people actually want quality algorithmic solvers, which they can rely on and they recognize they don't have the in-house competence (nor do they seek to try and hire such people) for the generation of such software. Because the monies involved are in their terms comparatively modest, they're willing to buy them. And I think that is an important differentiator: the work on the operating systems is a general requirement throughout the organization, whereas, the quality algorithms that they need for the fundamental elements of their research are components which come modestly priced, on the one hand, and they're not capable internally of generating and developing, on the other. And if they are, as I mentioned earlier in the interview, the institution is not going to let them develop themselves because of concerns about lack of experience in this area, and the impact it can have within economies of scale, and general company and business development.

HAIGH: So, other than the obvious PACKs, have there been any open source competitors to the library or other NAG products to emerge over the last decade or two?

FORD: Yes, currently there is an open source library being developed in, I think, MIT. The quality of some of that code is really quite good, and this is just a further carrying through of the open source philosophy. Now again what we're finding, fortunately for us, is many of our customer are suspicious of it because of its unproven and untried quality, but the pedigree and parentage of some of the material is of a very high order, and some of the code could be used I think with real confidence. We of course have been in this very strange situation, where we've worked actively in generating standards for things like the Level 2 and Level 3 BLAS, which are now used throughout the industry. But also, for example, Jeremy Du Croz and Sven Hammarling were directly involved in the development of LAPACK, and that code of course is now essentially identical to the code that we have in the NAG Library. So we've been publicly involved in the activity, which has developed competitor software to our own Library product. We've done that to show our good will within the community, and our earnest, and also as a means of carrying forward improved practices. What we've found, of course, is that because of the tight way that the code has been written to the specification of different configurations and the difference balance within the hardware and firmware configurations, we've been able to generate optimal libraries which run much, much quicker than the PACKs that have been put into the public domain, through LAPACK and through SCALAPACK. I think that's a tightrope we have to walk. You know we want to be part of the technical community, we want to encourage good practice, and that's the way the community has chosen to carry that through. On the other hand we have the clear concern I've shown about the open source movement and its potential for destroying the company that we have created and the products and services we have developed.

HAIGH: So do you think there are any strategies that you would be able to adopt other than those which you already have in place to deal with this threat?

FORD: Within the IFIP Working Group we've initiated discussion of a strategy for this within the community internationally and we've explained our concerns. This issue of the lack of investment of additional resources, because of feeling impacts from the open source philosophy on returns, is one that has found some sympathetic understanding. Have we found a solution yet? No. We've always had the issue of the largest group of users of numerical software being people who develop their own. We've never overcome that. We also now have the people who use the Numerical Recipe books, and the often aged algorithms that are incorporated in some of those books, although I must say they have got better. We just have to be coming out with algorithms which are state of the art, meet the criteria that we've always published, and through use of our libraries people find they've got a built-in updating service, so they're getting the improvement in the performance of algorithms that they're using year by year that competes with, and indeed, is now ahead of Moore's Law as far as the hardware is concerned.

HAIGH: So looking back over your career. What do you think would be your biggest regret?

FORD: That we didn't find a way of machine based preparation of documentation to support the library software earlier. That we didn't find a better methodology for preparing and testing libraries. That we perhaps spent too much effort in creating specific language versions of the library, particularly ADA and Pascal, which never had effective shelf lives, and that we didn't find ways of presenting our library components in a way that could effectively compete with MATLAB. We haven't yet found attractive techniques for teaching people about numerical analysis and computational statistics in a lightweight framework. That we didn't have the wit to ask the Computer Board, when they cut us loose, to give us a capital base to enable us to overcome the slings and arrows of outrageous fortune. That we didn't recognize at the time, when we were enjoying exponential growth and enormous success, to make sure we took and stored resource from that time for a bad day and to enable us to develop new products and services further down the road.

That we didn't put greater emphasis earlier into the United States to make the company there, more successful in the climate and the environment of the late 80s. We were offered the chance by Digital at one point to make the library exclusive to them, purely for use on VAX machines, and it's arguably if we'd done that we would have been able to build up substantial reserves and recreate the library for general use at a later time, hence fulfilling the contract with them while it ran, but, nevertheless, taking advantage of the enormous market that was offered us at that time. So a major regret was our failure to exploit the market when it was growing effectively, and exploitation of that specific relationship.

I also regret the fact that by sheer poor timing, because we were as well advised as we could have been, that the company sought to go for growth and the strengthening and broadening of its product base at the time the whole IT market went into sharp decline. I guess the only thing we're happy about from that is that we survived, where so many other companies didn't. NAG had a time, in the early 90s, when its business life was a bit quiet, associated with the period when my son died at University, and I've sometimes wondered if I ought not to have arranged a transition either temporarily during that period, or more strongly, to keep us pushing forward more effectively.

HAIGH: And on a more positive reverse of that, what thing or small number of things would you say that you're proudest of as tangible accomplishments?

FORD: I think the creation of NAG was worthwhile. I think the creation of the IFIP Working Group was worthwhile. I'm so grateful for all the friendships and relationships that NAG's brought me. The enormously different experiences that have flowed from it. I'm grateful for the opportunity to work with people like Steve, Linda, Shirley and Joan, in the early group, and with colleagues in the office and in the Council and in the Technical Policy Committee that have remained faithful friends through all the years, and how we've been able to do so much together. I'm glad to have been involved in a very young technology and had some small influence in helping it develop as far as scientific computing is concerned. I feel an enormous debt of gratitude to Jim Wilkinson for having faith in me when I was young, and other people who have kept faith with me through the years.

What we've sought to do has been very worthwhile and in a number of different ways throughout the time, thirty odd years, we have changed the face of scientific computing in a number of different ways, and I take some pleasure from that achievement. At least thirty known Nobel Prize winners have based their research work in whole or in part on NAG Library software. Perhaps the most exciting moment of my professional life was when Kahan said in the public meeting, “But Brian we’re all working on portable software now”, and this meant we’d turned him around in a period of less than five years. And it was nice to be given a decoration by the Queen. And in recent weeks it’s been equally nice to have received the support and interest of colleagues I’ve worked with, the collaborators I’ve worked with, my three chairmen, entrepreneurs leading other software companies who have taken the trouble to come and celebrate what we have achieved. This is not something one can expect, or do anything but be humbled by, and very grateful for.

HAIGH: That would seem to be the place to stop then.

FORD: Thank you. Thank you for coming and giving us the opportunity to talk with you. We were encouraged to do what we’ve done, initially by Jim Wilkinson, and then by many people in Europe and North America. Inevitably, we’ve always been keen to put a European viewpoint, which means on the one hand we have an understanding of history, but on the other hand means we have an international tradition of squabble. Thankfully we’ve left the squabble largely. We’ve been glad of the camaraderie and involvement, and commitment of our colleagues in North America, without who much of what we’ve done couldn’t have been achieved. We’ve been delighted that they’ve given us the opportunity to be involved in presidential committees on high performance computing, on selection of equipment for major installations, and program design within the great departments of the U.S. Government. And it’s been very good of SIAM to arrange for Tom to come visit us and give us a delightful two days of thinking about what we’ve tried to achieve and reminding us what we haven’t achieved as well. So we’ve got aims and objectives for the future. Thank you very much.

HAIGH: And thank you for taking part, and putting all this on record for future generations.