An Interview with

DAVID J. WHEELER

OH 132

Conducted by William Aspray

on

14 May 1987

Princeton, NJ

Charles Babbage Institute
The Center for the History of Information Processing
University of Minnesota, Minneapolis

David J. Wheeler Interview
14 May 1987

Abstract

Wheeler, who was a research student at the University Mathematical Laboratory at Cambridge from 1948-51, begins with a discussion of the EDSAC project during his tenure. He compares the research orientation and the programming methods at Cambridge with those at the Institute for Advanced Study. He points out that, while the Cambridge group was motivated to process many smaller projects from the larger university community, the Institute was involved with a smaller number of larger projects. Wheeler mentions some of the projects that were run on the EDSAC, the user-oriented programming methods that developed at the laboratory, and the influence of the EDSAC model on the ILLIAC, the ORDVAC, and the IBM 701. He also discusses the weekly meetings held in conjunction with the National Physical Laboratory, the University of Birmingham, and the Telecommunications Research Establishment. These were attended by visitors from other British institutions as well as from the continent and the United States. Wheeler notes visits by Douglas Hartree (of Cavendish Laboratory), Nelson Blackman (of ONR), Peter Naur, Aad van Wijngarden, Arthur van der Poel, Friedrich L. Bauer, and Louis Couffignal. In the final part of the interview Wheeler discusses his visit to Illinois where he worked on the ILLIAC and taught from September 1951 to September 1953.

DAVID J. WHEELER INTERVIEW

DATE: 14 May 1987 INTERVIEWER: William Aspray

LOCATION: Princeton, NJ

ASPRAY: This is an interview in Princeton, New Jersey with David Wheeler. We're going to discuss von Neumann and Dr. Wheeler's views on von Neumann. Let us begin by talking about your period at Cambridge University. We know that Cambridge and the Institute were two of the early centers interested in developing programming techniques and methods. The Institute reports and the Wilkes, Wheeler and Gill study were made widely available in the community. Could you tell me about your interactions at Cambridge with the Institute group? What did you hear about them? What did you hear from them?

WHEELER: I joined the lab about 1948. I joined as a research student. I was learning about the EDSAC. I can't quite remember which reports were available. I believe there were some reports available from Princeton, which were widely circulated. That was for a parallel machine -- a rather different one from the EDSAC. I cannot remember very much on the programming methods from the Princeton group. I didn't see anything about that, I think, until I went to Illinois some time later.

ASPRAY: I see.

WHEELER: Nevertheless, Wilkes, of course, had been to Princeton. But I don't think he was aware of some of the long-term timing of the larger computer projects, which were the basis of von Neumann's descriptive papers. The attitude at Cambridge, I think, was more that we had a machine coming along and we wanted to use it. Looking back, I think the emphasis (inspired by Wilkes) was that he wanted it to be used by many people; and I think for smaller problems, possibly, rather than a few large ones. The objective from the early days was to make it easy to use for people without specialized training. I think that was accomplished.

ASPRAY: Were there any other groups or individuals whose work was looked at by your group at Cambridge at that time, people like Rutishauser or Corrado Bohm or any of those people?

WHEELER: I didn't know those; but we did, in fact, have a weekly seminar which was attended from nearly all the installations interested in computing throughout the United Kingdom -- except, believe it or not, Manchester, which was too far away. It took rather a large number of hours to get to and from, so the interactions with Manchester were rather more occasional. There certainly were interactions with the NPL, with Booth's group, TRE, and so on. Most people interacted through those particular groups in the early days. Information went back and forth, and one knew of progress which was being made.

ASPRAY: What years did these weekly meetings occur?

WHEELER: I can't remember the exact years, but there were ones certainly while I was there as a research student, which would be 1948 until 1951. I think they went on afterwards. They were on Thursdays.

ASPRAY: They were going on the entire time you were a research student, or don't you recall?

WHEELER: I think they were going on the entire time, but my memory may not be accurate. They certainly started very early. I think it was weekly during term, maybe for half a year; possibly every week. But in a sense, that's the same thing.

ASPRAY: What group from Cambridge would attend? In addition to Wilkes, how many others might attend?

WHEELER: I'm using my memory, which is not very accurate, and it's probably averaging in a funny way over three years. But I think the numbers tended to vary between twenty and fifty. The numbers from Cambridge were, perhaps, somewhere between five and twenty. The Mathematical Laboratory in those days was not a large outfit. In fact, the number on the staff was three or four technicians and engineers who were building the EDSAC and maintaining it. Basically, because of the ease of use, once it started working (potentially slightly before), there were many groups of scientific people who were using the computer as much as those in the Mathematical Laboratory.

ASPRAY: What kinds of topics that we would now consider to be software would come under discussion at these meetings?

WHEELER: Well, there were the variety of systems. James H. Wilkenson gave some talks at some of these meetings. In the early days, the hardware was the primary concern because nobody had used these machines, but later on it tended to move over to software. I cannot remember exactly what was said about software. I did, for example, give a lecture, I can remember, on the circle coordinating orders, which was some time before a machine worked --which, therefore, would mean that these meetings were certainly going on in 1948.

ASPRAY: Yes.

WHEELER: So that was the software system, that later turned into the initial orders and the basis of the assembler. There were talks, for example, by J.C.P. Miller, on numerical methods. Hartree, I think, gave one or two. So there was quite a wide variety on both use and construction of computers.

ASPRAY: Do you recall much, if any, discussion of American developments that were reported on at these meetings? Were there ever American visitors to these meetings?

WHEELER: Certainly, Americans did give talks. I can't quite remember which ones. I haven't got the list of names; they've been lost, but, for example, we had strong contacts with ONR. A representative from ONR used to attend these, so that was one direct contact.

ASPRAY: Was this Nelson Blackman at the time?

WHEELER: There was Nelson Blackman and one or two others, I think. I can't quite recollect their names. They picked up information there and, also, circulated, among other things, the *ONR Newsletter* by Matt Menovich, which dealt with computer topics. Or was it under that name? It was, essentially, a circular about computer developments.

ASPRAY: Yes. A man by the name of Goldstine, I think, was the author of that newsletter.

WHEELER: I can't quite recollect the names. I can just recollect the thing. Well, these must exist in the records in a more available form.

ASPRAY: Yes.

WHEELER: Of course, the other point is the community was very much smaller. It is true we didn't know about Zuse. He only came to light sometime later. We knew about most of the American work, and there was Canadian work going on in Toronto. Some of the others we didn't know about. But we were very fortunate. Many people seemed to visit Cambridge. I didn't necessarily see them all.

ASPRAY: Did you hear about some of the early continental developments -- ERMETH, things like that?

WHEELER: I personally didn't, though, of course, Peter Naur and van Weingarten and Van der Poel had all visited, I think even attended courses in the Laboratory. So yes, we had some knowledge, but I can't actually remember anyone from ERMETH coming. I do remember though, rather later, a visit by Samuelson and Bauer. Of course, we knew about the work in France. I think Couffignal came over. We've had the occasional visitors since, but I think those are from that time.

ASPRAY: Yes.

WHEELER: And Hartree, of course, had quite a wide knowledge. He had visited, for example, the ENIAC and even run a problem on it. He was not in the Mathematical Laboratory, but he was a strong associate of the Mathematical

Laboratory.

ASPRAY: Yes. So he would be a regular visitor?

WHEELER: Oh, yes. I think, possibly, it's the formal structure. He did computational physics and some numerical analysis. He was in the Cavendish Laboratory, which was about fifty yards away.

ASPRAY: I see.

WHEELER: But formally he was not part of the Laboratory, and many other computer users in Cambridge were not part of the Laboratory.

ASPRAY: If you were to compare the styles of research or the progress that was made at the Institute as compared with what was done at Cambridge, what could you say about the differences and similarities between them?

WHEELER: Well, I think those of the Institute seemed to concentrate on somewhat larger problems than at Cambridge. Incidentally, it is only in the past few days I've really formalized this in my own mind. At Princeton, von Neumann described in a very careful analysis, sorting methods, and essentially how to use these computers on very large problems. Now, I think the attitude at Cambridge was rather different. It said, "We have a university. There are lots of scientific problems. Let the people use them and we will assist them." And most of these problems, as you would expect, were small. But nevertheless, a few of these turned out to be quite large. One of the largest sets of calculations that I know of, were done on the electronic wave functions by Frank Boys, who had a most intricate programming style, which I didn't report on at the meeting. This involved running all their programs twice, and students doing part of it. It involved six-dimensional integration, and the actual computation was essentially done by doing a computer algebraic differentiation, in order to get a series of terms, in order to do this integration. This was a computation which took tens to hundreds of hours. Therefore, he did repeat everything. He could be sure of some machine errors during that time. These were probably the largest calculations done, although crystallography and X-ray analysis of some of the crystals, myoglobin and things like that, took quite a long time as well.

ASPRAY: Well, do I remember correctly that there was some Nobel prize winning work done?

WHEELER: Yes, there were three. John Kendrow determined the structure of myoglobin by an X-ray analysis and a Fourier type synthesis. There was Peratz, who was concerned with DNA. There was Martin Ryle, who built a radio-telescope that would not have worked without a computer to do the Fourier transforms, because it was an interferometer. To convert this back into ordinary observations, you had to do a Fourier transform. In the early days, the information was conveyed by punched paper tape on a bicycle from the radiotelescope to the machine. That might seem a very low bandwidth operation, but it was relatively reliable; rather better than setting up apparatus to do it over telephone lines at that time.

ASPRAY: Yes. Were the specific programming technologies or methodologies different between Cambridge and the Institute?

WHEELER: They were totally different.

ASPRAY: Could you explain that to me?

WHEELER: I do not know very much about the actual (you might call it) operating system at Princeton, although I did visit it once. It struck me by being remarkably primitive. The programs, I think, were input in binary. Even the input of programs required manual intervention; it was not automatic. At Cambridge we started out very early by having an "assembler". It essentially converted programs as they were written, went from decimal to binary, used mnemonics, had a reference scheme, allowed compartmentalization, allowed for automatic positioning of library routines, and had various other facilities built in as well. This was all from, roughly, the day we started. This made it very much easier to put programs together. As a slight comment, if you like, on the design of the machine, Wilkes had arranged for a press button, which put a small program in store, which then read in the tape. It was not a question of reading some tape in, and then manually stopping it, and then manually starting the machine, which was as it was, I think, on the Princeton machine. So, it was arranged to be simple to use and automatic. Attention was given to details, which were just not considered important at Princeton. I think this is because they ran a few very large programs. For example (to take a trivial example), consider loading a paper tape. Our tapes started with some blank, so you could load anywhere in the blanks. At Princeton you had to load at a specific position on the tape. That's just an example. Our attention, basically was on different things. We were particularly interested in making it easy for the users to get results, and we concentrated on that quite a lot. The programming styles were modularized very early on. That was the way we taught. We certainly knew, for example, about von Neumann's flow diagrams. These were never used, except possibly as an explanation of what had been done. We discovered that if you basically took the problem, decomposed it into subroutines, and put these under the control of the master routine, this more or less forced you to think modularly and enabled you to do all these programs without going to the flow diagrams, which in my own personal opinion contributed more to bad programming than almost anything else. This is not to say that they don't have their place, but if they're used as a substitute for thought in the preparation of your program, they don't work.

ASPRAY: I see. What about subroutines, libraries of functions, functions that were built into the machines? Were there differences that you can recall between the two programming groups in that connection?

WHEELER: Yes, I believe we standardized, if you like, a very simple subroutine structure. This was maintained through the life of the machine. I think the von Neumann machine had a rather more complicated one. Now I'm not sure whether they changed over later; certainly the ILLIAC and the ORDVAC used conventions similar to that which we used at Cambridge. Now, for that matter, so did the 701. But the Princeton machine stated it could be done, and as it could be done, this was sufficient and they got on with the hard task of doing the big problem analysis. We rapidly generated the subroutines for all the standard functions. I discovered, incidentally, on the phone from an old colleague, as I came, that the sub routines used economized power series to reduce space and time. We had rather more versions of some routines than you would have now, because we used to have some which were small and some which were fast. People had to select on this basis. As we had a few hundred words of store this selection was needed.

ASPRAY: Yes.

WHEELER: I think special functions may or may not have been added later on, but the simple functions were there. The input-output routines were relatively elaborate. There's another distinction there, possibly between Princeton and Cambridge, or between very many other groups and Cambridge, which has lasted quite a long time. It may have been the influence of Miller; I am not quite sure, but we thought that we had to present results properly. Output routines, for example, allowed you to put in a fill-in space. You print numbers to ten figures. You could choose to put an extra space, or spaces anywhere you liked. So it was possible to make genuinely readable results. Then comes along, say FORTRAN, and that facility is taken away, and not yet been restored. We had routines for block layout and other things, but the teleprinter had a width of 70 characters and printed on a roll of paper, so we were restricted. But there were quite a range of input and output routines; mainly to provide good printing facilities, even though we had few figures to print at seven characters a second.

ASPRAY: Were hardware constraints sufficiently different on the EDSAC and the Institute computer that programming considerations were somewhat different on the two?

WHEELER: I think they were. I mean, I went to the ORDVAC and ILLIAC, which had a thousand words. At the most, the EDSAC had probably 512 words. And most of its, or certainly, some of its life it had far fewer than that. I think it also had a variable available number every day, which doesn't usually appear in documents, but it was typical of the things that happened.

ASPRAY: Why would that occur?

WHEELER: Some of the mercury delay lines developed troubles, and, basically, the engineers, instead of repairing immediately, would rearrange them. So, today the store is 448 words; tomorrow it's 512 words. It varied a little. But some programs didn't require the full store and they could run. This was a temporary expedient. So, this happened. The second constraint was certainly the EDSAC, in its early days, was very unreliable.

ASPRAY: So was the Institute machine.

WHEELER: The Institute machine was. The EDVAC never really achieved working status, though that's possibly a slightly biased opinion. But all these early machines were not only unreliable, also they had days when they were much more reliable, and days when basically there were some marginal faults which had not been removed and the reliability on those days was very much smaller than the average.

ASPRAY: Yes.

WHEELER: Now, we both know this doesn't mean they couldn't do any work. It did, in fact, lead to problems of administration. If the machine is known not to do all problems but it's able to do mine, I'm not very interested in having it repaired.

ASPRAY: Yes.

WHEELER: This effect was around. Very often it was the case that a machine would not do all programs, but it would do most of them.

ASPRAY: What was the administrative response to this?

WHEELER: The administration, in this case, was probably the engineer who took this part of the decision. I think, in general, they would say that they would repair it; but in certain cases, they didn't. In certain cases, I can't blame them. If I may include a personal recollection here, which is perhaps not very good, there was a program I wrote to test the primality of numbers on the EDSAC. This took half an hour. It tested, I think, at the time the highest prime number. It was an interesting program. It computed half an hour. The result was either one or a wrong number. For some strange reason, this was taken to be a test of the machine. If it would do that calculation, it was pretty good because the slightest error would cause a one to become some other number. It's not like a little bit of rounding error. But it rather frustrated the engineers, because the users tended to believe that if they could run this and show the machine wasn't running correctly, it was up to the engineers to make it work. On the other hand, as you would undoubtedly know, very many times when that would say the machine is not working correctly and all the engineers' test programs would work perfectly, which gave them no clue as to where the error was. That was one of the problems that has not altogether been solved, but that occurrence is far less frequent than it used to be.

ASPRAY: From a practical consideration, what is the difference between having a machine with approximately 500 words, as compared with one with a thousand words? What sorts of things are you now able to do with the doubled store?

WHEELER: I've always maintained that the machine selects the problems that they can do. I think this is true. Once you had gone up to a thousand, matrices became much more realistic. They were large enough to be interesting in places where they had not been used before. This was certainly a main difference that I noted. The proportion of matrix work on the thousand word machine was much larger than when you had a few hundred words, which you also shared with your program (so the data storage was even less than that). In these cases the matrices were rather small, unless you read them in with punched cards, and out with punched cards, and in with punched cards -- which happened on some of the early computers -- or alternatively had a backing store, which the EDSAC did not have. So I did not have in the early days a better way of putting it. That was the story. Now you could go out on punched paper tape and back on punched paper tape. But that was very, very slow. So basically, that didn't happen. The machine instead selected differential equations, Fourier transforms, and other things, which were not quite so space-intensive.

ASPRAY: I see. Can you give me some idea of the relative impact, as far as you knew it, of the Wilkes, Wheeler and Gill book, as compared with, say, the Institute reports on programming from 1947 to 1949?

WHEELER: I think that's very difficult. The von Neumann reports were a kind of a seed. It germinated in many places, but didn't make much growth. By the time the book came out the whole stage had changed. There were very many more people interested; therefore, it impacted on a larger number of people. Now, I don't really know how you compare. It's certainly true that part of the programming system of the EDSAC was copied in many places -- a surprising variety of places. It did influence things; but on the other hand, so did the von Neumann influence people to get started. So I think they both had their influences, but at difference points of the growth cycle.

ASPRAY: One last question before turning to your Illinois experiences, though it might cover the same period. In some sense, what we now call software got moving rapidly around 1956, 1957, 1958, when we were getting more established operating systems, although there were certainly earlier developments. The first high level programming languages start to appear and so on. In this period of the *early* 1950s, what would you judge to be the most significant contributions in this software area?

WHEELER: That's very tricky to say. In a sense, they all disappeared. That was the time when most machines were coded in binary or in assembler. Compilers had not come and users did not communicate across different types of machines, except by using ideas -- which very often is still the way it's done. Now if you asked me what was the most significant part, it's very difficult for me to say. Most of the evolution could probably be predicted in advance, except that I think the job of getting programs right had been underplayed. This was played, partly, I think, by poor machine design. I think by the end of that year, it was beginning to improve slightly. Earlier, people had nothing to go on. From then, designers claimed they were making computers easier to use. There were endless arguments about whether a one address or a three address code is the most efficient. Similar arguments concerned word length, floating point, and so on.

TAPE 1/SIDE 2

ASPRAY: To go back to where we were: one address, three address; but they didn't affect the user.

WHEELER: They didn't affect the user very much. The infelicities of the order code did. You know, you can think back to some of them there were two zeros of different sign in some machines, and so counts didn't necessarily terminate correctly, if you tested for the sign and the computed zero had the other sign. Many computers had a large number of accidental instructions, because their instructions weren't fully decoded, and some users rather liked doing bizarre things. All this distracted from the job of getting things right. At that time there weren't many ideas on how to get errors out of programs. It was originally thought that you wrote it down and it was right. In practice, it turned out to be an experimental art of gently removing the errors. This was affected not only by the machine design, but also by the aptitudes of people, and also by the operators. In some of those early days, something called "batch mode" turned up, which meant that users only had access to a machine maybe once every 24 hours, or once every 48 hours, which was definitely bad. We were lucky; we seemed to have learned from the beginning that you had to have a large number of test periods a day, probably because we didn't design the operating system in advance. This was effective. But nevertheless, a lot of information did start circulating. Checking on computers was not put in in the early days. Later parity circuits were nearly always used as memory checks. Why wasn't it put in? Because this would have made more equipment; and more equipment was less reliable (which is probably right). The EDSAC, for example, had marginal checking added later, which was copied from WHIRLWIND. This improved reliability. Before then components such as tubes had gently aged. Naturally, most of them died during the working period and not during the maintenance period. The idea of marginal checking was that they would decay and they would be detected during the maintenance period as the worse voltages would cause errors. There were many systems getting errors out of programs. In the early stages there was an autocode made by Alec Glennie. In my opinion, it was a distraction from the job of programming, at that time, because it didn't actually tackle the hard parts of programming. The actual translation of formulae into algebra was really a very small part of the total programming task. Besides that, you had input of numbers, cycling, printing results, possibly checking results, and going through the calculations, and all these other matters. And so, these things were, I think, a side track at the time. They made it easier to write programs and change programs, but gave less flexibility, used valuable store and reduced the number of test runs in a given time. The overheads on small programs were considerable.

ASPRAY: All right. Let's turn to your time at Illinois. Could you very briefly recount the conditions under which you came to Illinois and tell me about your duration there, and what your responsibilities were while you were there?

WHEELER: I think Hartree had visited Illinois, and so had Maurice Wilkes. I was offered a job there, I think, as a visiting assistant professor for a nine-month term. I extended that for one extra year before going back to Cambridge. I had also been awarded a fellowship at Trinity College. Essentially I went back to that.

ASPRAY: What years were you in Illinois?

WHEELER: It was from September 1951 to September 1953. When I got there, the ORDVAC was just about working. I can't actually remember if it was working on the day that I got there. But "working" was a gentle term in those days; the proportion of time it is working gently increases. So, to give it a day is a little difficult. When I got there it had a very slow tape reader. It was a complete change from the EDSAC. Instead of taking a whole room, it was essentially one very large cabinet in a very much larger room. It was delightfully compact. The input and output were very similar to the EDSAC. There was the job of systemizing its use. Rather naturally, I chose a system similar to the one which we had at Cambridge. I don't think I had any direct responsibilities, but I was part of the group there, which basically constructed a system to get people using the machine. This was done relatively quickly. Unlike the EDSAC, which was started with a simple pushbutton, you had to, in fact, inject an order pair by touching wires. The order pair then read in a bootstrap, which read the assembly program, which then read the main program. Thus, the system was more complicated. Later on, they got a Ferranti reader, which read at 300 characters a second, which meant the input time was far less. Of course, there were certain changes from the system at Cambridge. It was arranged that the assembler could also read in numbers, which wasn't possible in the space that the EDSAC had, due to the limited size of the stepping switches. But here one could choose to do it. One chose to allow for numbers to be read by the assembler, so the need for a large set of input and binary input subroutines was diminished. Apart from that there were various similarities and a number of differences. The ILLIAC was a parallel machine that went much faster. The constraints were rather different. Instead of having a store with a few hundred words, you had a store of a thousand and twenty-four words precisely. On the other hand, there were limitations on the use of the store. One limitation was read-around, which meant you couldn't use a particular register too frequently. You couldn't use the registers in the store adjacent to one which you intended to refer to very often, because the cross-talk could accumulate and caused errors in adjacent words. The read-around ratio was very low to start with, maybe 20, but it improved later to some value like 50 or 100. At the start it was an important programming consideration. At the end it was a minor constraint. This difficulty was overcome by using a few rules. These rules were far simpler than the optimum programming rules, say, used on the Ace Pilot model or the EDVAC. So it was a chore, but it was not a very severe chore.

ASPRAY: Yes. I see. Did you do any teaching while you were there?

WHEELER: Yes, I did something that I had not done before. At Cambridge, we taught the graduate students who were about to do research. We had an intensive course for them. But at Illinois I taught the undergraduate classes, which was part of their curriculum, and this was very interesting. I think this probably had an influence later on. The departments were interested in using computers, and students got trained for them. It was a kind of culture using the computer, which started from that. I think this is partly a consequence again of the open-shop way in which the computer was used.

ASPRAY: Do you recall much about what was taught in the courses, or how many students you had?

WHEELER: I think that at the courses that I taught, as opposed to other people, there must have been 30 to 60 attendees. Essentially, programming methods were taught. Programming methods for the ILLIAC or the ORDVAC.

ASPRAY: Yes, but there were a number of others teaching courses also?

WHEELER: Oh, yes. The other members of the Laboratory gave courses. There was a chairman whose name I forget, Abe Taub, John Nash, Jim Robertson, later on Jim Snyder and a supporting staff that grew through the years. Other people became involved, but those were there at the beginning.

ASPRAY: Yes. Do you recall anything about the nature of the students? You said it was primarily an undergraduate group.

WHEELER: Yes, I think it was in the Electrical Engineering Department. The computing lab there was part of the Electrical Engineering Department and was housed there. The structure didn't actually affect me, but I think that it was quite advantageous that the computer laboratory there was connected both with mathematics and electrical engineering. I can't remember the exact administrative details, but I have the feeling that, basically, in a sense it gave the rights back to mathematics and electrical engineering in an administrative politically expedient way.

ASPRAY: All right. Let me turn to a slightly different question about ILLIAC and ORDVAC. When the plans were being drawn up for the Institute computer, there was a great deal of discussion between von Neumann and various people at high levels in the government about what these machines should do -- the Institute computer and the clones (copies of the Institute computer). There was one plan that von Neumann had advocated that the Institute computer was to be used for determining whether certain kinds of scientific problems could be solved on machines of this sort, but that the machine wasn't supposed to be used to do these problems once it had been shown feasible -- that the more routine working out of this was to be done on the ILLIACs and the ORDVACs and so on. Did that actually happen in your experience of ILLIAC and ORDVAC?

WHEELER: Well, the ORDVAC went to Aberdeen Proving Ground. This was a government department. It catered for a lot of the competing load. It was probably one of the longest lived early computers, because I think it was still going 16 years later. Nobody knows actually when it died, which I find rather funny. The ILLIAC was, of course, owned by the University of Illinois, not any government department. The contract, I think, said that it was a quarter of million dollars put in by the government and the University of Illinois. They built two machines and shipped one off to the government, but the other, as far as I know, had no restrictions on its use.

ASPRAY: I see.

WHEELER: So they could use it for whatever problems they liked. Basically it was used for problems as submitted from other departments. One of the strange things, to me, was the enormous number of factor analyses done by essentially what you now call non-recessional social sciences.

ASPRAY: Social sciences?

WHEELER: Social sciences. Now, finding eigenvalues is kind of a very strong mathematical field. To find the greatest application was in the social sciences was a little strange. But of course, once you'd been programmed up it was a question of turning a handle, and then the results came back. So, there were jobs from various departments. There was some work under government control for the control systems laboratory, which was housed immediately above the ILLIAC, and some wires came down. This was used on secret government work and they did use the machine for an allotted period of time; I can't quite remember what it was.

ASPRAY: Did you see the hand of von Neumann at all in the work at Illinois? Were there problems sent from the Institute that you recall?

WHEELER: Abe Taub was a great friend of von Neumann. He often came. I didn't actually see him, which is quite surprising.

ASPRAY: That is, von Neumann often came.

WHEELER: Yes.

ASPRAY: Yes.

WHEELER: They communicated quite rigorously, so there were certain lines of communication. The engineers certainly communicated. Progress on the use of the cathode ray, William Store, and overcoming the read around problem was done by changing some circuits. This was communicated between basically all the copies of the Institute machine. So they did cooperate on the memory field quite effectively at a different level. I think that was probably not necessarily through Julian Bigelow, but through some of his engineers.

ASPRAY: I see.

WHEELER: I know I coded the Jacobi method for the eigen calculations on matrices. I know it originated in Princeton with Herman Goldstine. I don't think I saw the paper but the method was just explained to me, and I coded it up. Of course it is slightly difficult to code even when you actually have the algorithm. It was heavily used. Many systems of linear equations were solved. We used methods developed for that size machine. I think there were also sorting routines, and chain lists, and others used, which later were given rather more elaborate names. The control systems lab had sorting problems. They basically wanted to sort out positions of aircraft detected by radar, so that they could be displayed and so on. I certainly suggested some techniques for sorting, which I believe they used, but I'm not absolutely sure.

ASPRAY: I would be pleased if you would make a few general remarks about von Neumann and the Institute's early contributions to the field of computing numerical analysis.

WHEELER: Well, I think the Institute was interested in large problems and analyzed certain problems very carefully before most people had. Of course, their intention was directed to some quite large problems, but possibly more important than that, they stimulated development. The Princeton machine was built, and a large number of copies were built. The people who built copies stimulated developments widely. The ORDVAC was built; the ILLIAC was built; and then the SILLIAC was built in Sydney, that was one particular chain which I happen to know of. And BESK, which we heard about today, was built. There were very many copies of the Institute machine. It was one of the most copied machines which there was, even though no copies were identical.

ASPRAY: And the 701, in a way.

WHEELER: The 701 was also a similar type machine. At that time, they were most certainly the most powerful machines. Even though the original Selectron didn't work, the Williams Tube compensated for that in a fairly effective way. Even though it wasn't a perfect memory, it did get that speed edge, which was very useful in the early days.

ASPRAY: Do you think that von Neumann's established reputation as a scientist and his contacts within the government had some bearing on his success?

WHEELER: Yes, I'm sure von Neumann stimulated the progress by many years. This is particularly true in the States where he seems to have had an influence everywhere he went, and seems to have been beneficial for the computer industry. To estimate how fast the progress would have been without him is difficult, but I feel it would have been delayed by a fair number of years. As you said, he also basically made the subject respectable. Numerical analysis and the use of computers was a very funny subject in the early days. It was slightly beneath the dignity of mathematicians; possibly, engineers were more interested. They had to get results. In any case, they were used to doing calculations, whereas mathematicians weren't. Real mathematical problems didn't get put on computers for nearly two or three decades after they started. All right, there was some minor group theory in the early days; generating prime numbers or testing primality. This was a very elite branch of mathematics, if you like, but real mathematics just did not go on computers. Mathematicians tended to believe that they did not do calculations. They did algebra, and topology, and other things; whereas scientists used mathematics and did calculations and tended to use computers.

ASPRAY: To follow up on that remark, one of the things that has always struck me about von Neumann was that he's in a class by himself, or with only a small number of people in the following sense: the other very well-established scientists who had reputations already, people like Vannevar Bush and Sam Caldwell, were tied to an earlier calculating technology. The other very fine scientists and engineers involved in computing (who we now look back on as the pioneers) were, for the most part, young people without reputations and without those opportunities to get grants to build machines in those days. He's the one person who stands out (in this country, anyway) as being able to do that.

WHEELER: Yes, but he also directly, I think, stimulated thought on some more special items. He did work in economics; he stimulated work on Monte Carlo methods. He also stimulated work on cellular automata. He showed, for example, that you can make unreliable components work as a reliable computer; and also he worked on conditions for self-reproduction of components. I can't actually remember where I saw those papers, but they were certainly some of the more interesting ideas. Some of them haven't really reached full fruition yet, but nevertheless it was a great stimulus.

ASPRAY: Would you say that Hartree, in a sense, played the role of legitimizer in Britain the way that von Neumann did in the United States?

WHEELER: I think not to the same extent. He certainly had a role in legitimizing it, which was at the time significant. I think we're slightly fortunate that the mathematical laboratory was set up in the mathematics faculty, as opposed to the engineering faculty at most other places. Now this had a bearing, I think, in spite of the fact that most mathematicians didn't use the computer and that the early users were definitely from the other branches. In a sense, it made it a slightly more legitimate operation in the eyes of the mathematical faculty. Hartree certainly publicized, stimulated, and tended to make things respectable. In Cambridge, there were so many different scientific departments which wished to do calculations; that if the mathematicians didn't wish to do them, it really didn't matter. Although in a certain vagueish way from, say, academic prestige, there are some people, (particularly mathematicians) who tend to put mathematics on the top of the tree. But in real life, it's much less a disadvantage.

END OF INTERVIEW