LECTURE 5

HISTORY OF APPLICATIONS

As you have probably noticed, I am using the technical material to hang together a number of anecdotes, hence I shall begin this time with a story of how this, and the two preceding talks, came about. By the 1950's I had found that I was frightened when giving public talks to large audiences, this in spite of the fact that I had taught classes in college for many years. On thinking this over very seriously, I came to the conclusion that I could not afford to be crippled that way and still become a great scientist; the duty of a scientist is not only to find new things, but to communicate them successfully in at least three forms:

writing papers and books
prepared public talks
impromptu talks

Lacking any one of these would be a serious drag on my career. How to learn to give public talks without being so afraid was my problem. The answer was obviously by practice, and while other things might help, practice was a necessary thing to do.

Shortly after I had realized this it happened that I was asked to give an evening talk to a group of computer people who were IBM customers learning some aspect of the use of IBM machines. As a user I had been through such a course myself and knew that typically the training period was for a week during working hours. To supply entertainment in the evenings IBM usually arranged a social get-together the first evening, a theater party on some other evening, and a general talk about computers on still another evening - and it was obvious to me that I was being asked to do the later.

I immediately accepted the offer because here was a chance to practice giving talks as I had just told myself I must do. I soon decided that I should give a talk that was so good that I would be asked to give other talks and hence get more practice. At first I thought I would give a talk on a topic dear to my heart, but I soon realized that if I wanted to be invited back I had best give a talk that the audience wanted to hear, which is often a very, very different thing. What would they want to hear, especially as I did not know exactly the course they were taking and hence the abilities of people? I hit on the general interest topic, The History of Computing to the Year 2000. - this at around 1960. Even I was interested in the topic, and wondered what I would say! Furthermore, and this is important, in preparing the talk I would be preparing myself for the future.

In saying, "What do they want to hear?" I am not speaking as a politician but as a scientist who should tell the truth as they see it. A scientist should not give talks merely to entertain, since the object of the talk is usually scientific information transmission from the speaker to the audience. That does not imply that the talk must be dull. There is a fine, but definite, line between scientific communication and entertainment, and the scientist should always stay on the right side of that line.

My first talk concentrated on the hardware, and I dealt with the limitations of it including, as I mentioned in Lecture 3, the three relevant laws of Nature; the size of molecules, the speed of light, and the problem of heat dissipation. I included lovely colored VuGraphs with overlays of the quantum mechanical limitations, including the uncertainty principle effects. The talk was successful since the IBM person who had asked me to give the talk said afterwards how much the audience had liked it. I casually said that I had enjoyed it too, and would be glad to come into NYC almost any evening they cared, provided they warned me well in advance, and I would give it again - and they accepted. It was the first of a series of talks that went on for many years, about two or three times a year, in which I got a lot of practice and learned not to be too scared. You should feel some excitement when you give a talk since even the best actors and actresses usually have some stage fright. Your excitement tends to be communicated to the audience, and if you seem to be perfectly relaxed then the audience also relaxes and may fall asleep!

The talk also kept me up to date, made me keep an eye out for trends in computing, and generally paid off to me in intellectual ways as well as getting me to be a more polished speaker. It wasn't all just luck - I made a lot of it by trying to understand, below the surface level, what was going on. I began, at any lecture I attended anywhere, to pay attention not only to what was said, but to the style in which it was said, and whether it was an effective or a noneffective talk. Those talks that were merely funny I tended to ignore, though I studied the style of joke telling closely. An after dinner speech requires, generally, three good jokes; one at the beginning, one in the middle, and a closing one so that they will at least remember that one joke; all jokes of course told well. I had to find my own style of joke telling, and I practiced it by telling secretaries various jokes.

After giving the talk a few times I realized that, of course, it was not just the hardware, but also the software that would limit the evolution of computing as we approached the year 2000 - Lecture 4 that I just gave you. Finally, after a long time, I began to realize that it was the economics, the applications, that probably would dominate the evolution of computers. Much, but by no means all, of what would happen had to be economically sound. Hence this Lecture.

Computing began with simple arithmetic, went through a great many astronomical applications, and came to number crunching. But it should be noted that Raymond Lull, (1235?-1315), sometimes written Lully, a Spanish theologian and philosopher, built a logic machine! It was this that Swift satirized in his Gulliver's Travels when on the island of Laputa, and I have the impression that it corresponds to Majorca where Lull flourished.

In the early years of modern computing, say around 1940's and 50's, "number crunching" dominated the scene since people who wanted hard, firm numbers were the only ones with enough money to afford the price (in those days) of computing. As computing costs came down the kinds of things we could do economically on computers broadened to include many other things than number crunching. We had realized all along that these other activities were possible, it was just that they were uneconomical at that time.

Another aspect of my experiences in computing was also typical. At Los Alamos we computed the solutions of partial differential equations (atomic bomb behavior) on primitive equipment. At Bell Telephone Laboratories at first I solved partial differential equations on relay computers; indeed I even solved a partial differential-integral equation! Later, with much better machines available, I progressed to ordinary differential equations in the form of trajectories for missiles. Then still later I published several papers on how to do simple integration. Then I progressed to a paper on function evaluation, and finally one paper on how numbers combine! Yes, we did some of the hardest problems on the most primitive equipment - it was necessary to do this in order to prove that machines could do things that could not be done otherwise. Then, and only then, could we turn to the economical solutions of problems that could be done only laboriously by hand! And to do this we needed to develop the basic theories of numerical analysis and practical computing suitable for machines rather than for hand calculations.

This is typical of many situations. It is first necessary to prove beyond any doubt that the new thing, device, method, or whatever it is, can cope with heroic tasks before it can get into the system to do the more routine, and in the long run, more useful tasks. Any innovation is always against such a barrier, so don't get discouraged when you find that your new idea is stoutly, and perhaps foolishly, resisted. By realizing the magnitude of the actual task you can then decide if it is worth your efforts to continue, or if you should go do something else that you can accomplish and not fritter away your efforts needlessly against the forces of inertia and stupidity.

In the early evolution of computers I soon turned to the problem of doing many small problems on a big machine. I realized that, in a very real sense, I was in the mass production of a variable product - that I should organize things so I could cope with most of the problems that would arise in the next year, while at the same time not knowing what, in detail, they would be. It was then that I realized that the computers have opened the door much more generally to the mass production of a variable product, regardless of what it is; numbers, words, word processing, making furniture, weaving, or what have you. They enable us to deal with variety without excessive standardization, and hence we can evolve more rapidly to a desired future! You see it at the moment applied to computers themselves! Computers, with some guidance from humans, design their own chips, and computers are assembled, more or less, automatically from standard parts; you say what things you want in your computer and the particular computer is then made. Some computer manufacturers are now using almost total machine assembly of the parts with almost no human intervention.

It was the attitude that I was in the mass production of a variable product, with all its advantages and disadvantages, that caused me to approach the IBM 650 as I told you I did in the last Lecture. By spending about 1 man year in total effort over a period of 6 months, I found that at the end of the year I had more work done than if I had approached each problem one at a time! The creation of the software tool paid off within one year! In such a rapidly changing field as computer software if the payoff is not in the near future then it is doubtful that it will ever pay off.

I have ignored my experiences outside of science and engineering - for example I did one very large business problem for AT&T using a UNIVAC-I in NYC, and one of these days I will get to a lesson I learned then.

Let me discuss the applications of computers in a more quantitative way. Naturally, since I was in the Research Division of Bell Telephone Laboratories, initially the problems were mainly scientific, but being in Bell Telephone Laboratories we soon got to engineering problems. First, Figure 5-1, following only the growth of the purely scientific problems, you get a curve that rises exponentially, (note the vertical log scale), but you soon see the upper part of the S curve, the flattening off to more moderate growth rates. After all, given the kind of problem I was solving for them at that time, and the total number of scientists employed in Bell Telephone Laboratories, there had to be a limit to what they could propose and consume. As you know they began much more slowly to propose far larger problems so that scientific computing is still a large component of the use of computers, but not the major one in most installations.

The engineering computing soon came along, and it rose along much the same shape, but was larger and was added on top of the earlier scientific curve. Then, at least at Bell Telephone Laboratories, I found an even larger military work load, and finally as we shifted to symbol manipulations in the form of word processing, compiling time for the higher level languages, and other things, there was a similar increase. Thus while each kind of work load seemed to slowly approach saturation in its turn, the net effect of all of them was to maintain a rather constant growth rate.

What will come along to sustain this straight line logarithmic growth curve and prevent the inevitable flattening out of the S curve of applications? The next big area is, I believe, pattern recognition. I doubt our ability to cope with the most general problem of pattern recognition, because for one thing it implies too much, but in areas like speech recognition, radar pattern recognition, picture analysis and redrawing, work load scheduling in factories and offices, analysis of data for statisticians, creation of virtual images, and such, we can consume a very large amount of computer power. Virtual reality computing will become a large consumer of computing power, and its obvious economic value assures us that this will happen, both in the practical needs and in amusement areas. Beyond these is, I believe, Artificial Intelligence, which will finally get to the point where the delivery of what they have to offer will justify the price in computing effort, and will hence be another source of problem solving.

We early began interactive computing. My introduction was via scientist named Jack Kane. He had, for that time, the wild idea of attaching a small Scientific Data Systems (SDS) 910 computer to the Brookhaven cyclotron where we used a lot of time. My V.P. asked me if Jack could do it, and when I examined the question (and Jack) closely I said I thought he could. I was then asked, "Would the manufacturing company making the machine stay in business?", since the V.P. had no desire to get some unsupported machine. That cost me much more effort in other directions, and I finally made an appointment with the President of SDS to have a face to face talk in his office out in Los Angles. I came away believing, but more on that at a later date. So we did it, and I believed then, as I do now, that that cheap, small SDS 910 machine at least doubled the effective productivity of the huge, expensive cyclotron! It was certainly one of the first computers that during a cyclotron run gathered, reduced, and displayed the gathered data on the face of a small oscilloscope (that Jack put together and made operate in a few days). This enabled us to abort many runs that were not quite right; say the specimen was not exactly in the middle of the beam, that there was an effect near the edge of the spectrum and hence we had better redesign the experiment, that something funny was going on, and we would need more detail here or there - all reasons to stop and modify rather than run to the end and then find the trouble.

That one experience led us at Bell Telephone Laboratories to start putting small computers into laboratories, at first merely to gather, reduce, and display the data, but soon to drive the experiment. It is often easier to let the machine program the shape of the electrical driving voltages to the experiment, via a standard digital to analog converter, than it is to build special circuits to do it. This enormously increased the range of possible experiments, and introduced the practicality of having <u>interactive experiments</u>. Again, we got the machine in under one pretext, but its presence in the long run changed both the problem and what the computer was actually used for. When you successfully use a computer you usually do an equivalent job, not the same old one. Again you see that the presence of the computer, in the long run, changed the nature of many of the experiments we did.

Boeing (in Seattle) later had a somewhat similar idea, namely that they would keep the current status of a proposed plane design on a tape and everyone would use that tape, hence in the design of any particular plane all the parts of the vast company would be attuned to each other's work. It did not work out as the bosses thought it would, and as they probably thought it did! I know, because I was doing a high level, two week snooping job for the Boeing top brass under the guise of doing a routine inspection of the computer center for a lower level group!

The reason it did not work as planned is simple. If the current status of the design is on the tape, (currently discs), and if you use the data during a study of, say, wing area, shape, and profile, then when you make a change in your parameters and you find an improvement it might have been due to a change someone else inserted into the common design and not to the change you made - which might have actually made things worse! Hence what happened in practice was that each group, when making an optimization study, made a copy of the current tape, and used it without any updates from any other area. Only when they finally decided on their new design did they insert the changes - and of course they had to verify that their new design meshed with the new designs of the others. You simply cannot use a constantly changing data base for an optimization study.

This brings me to the topic of data bases. Computers were to be the savior in this area, and they are still occasionally invoked as if they would be. Certainly the airlines with their reservation systems is a good example of what can be done with computers - just think what a mess it would be when done by hand with all its many human errors, let alone the size of the troubles. The airlines now keep many data bases, including the weather. The weather and current airport delays are used to design the flight profile for each flight just before takeoff, and possibly change it during flight in view of later information.

Company managers always seem to have the idea that if only they knew the current state of the company in every detail then they could manage things better. So nothing will do but they must have a data base of all the company's activities, always up to the moment. This has its difficulties as indicated above. But another thing; suppose you and I are both V.P.s of a company and for a Monday morning meeting we want exactly the same figures. You get yours from a program run on Friday afternoon, while I, being wiser and knowing that over the weekend much information comes in from the outlying branches, wait until Sunday night and prepare mine. Clearly there could be significant differences in our two reports, even though we both used the same program to prepare them! That is simply intolerable in practice. Furthermore, most important reports and decisions should not be time sensitive to up to the minute data!

How about a scientific data base? For example, whose measurement gets in? There is prestige in getting yours in, of course, so that there will be hot, expensive, irritating conflicts of interest in that area. How will such conflicts be resolved? Only at high costs! Again, when you are making optimization studies you have the above problem; was it a change made in some physical constant that you did not know happened that made the new model better than the old model? How will you keep the state of changes available to all the users? It is not sufficient to do it so that the users must read all your publications every time they use the machine, and since they will not keep up errors will be made. Blaming the users will not undo the errors!

I began mainly talking about general purpose computers, but I gradually took up discussing the use of a general purpose computer as a special purpose device to control things, such as the cyclotron and laboratory equipment. One of the main steps happened when someone in the business of making integrated circuits for people noted that if instead of making a special chip for each of several customers, he could make a four bit general purpose computer and then program it for each special job. He replaced a complex manufacturing job with a programming job, though of course the chip still had to be made, but now it would be a large run of the same four bit chips. Again this is the trend I noted earlier, going from hardware to software to gain the mass production of a variable product - always using the same general purpose computer. The four bit chip was soon expanded to 8 bit chips, then 16, etc. so that now some chips have 64 bit computers on them!

You tend not to realize the number of computers you interact with in the course of a day. Stop and go lights, elevators, washing machines, telephones which now have a lot of computers in them as opposed to my youth when there was always a cheerful operator at the end of every line waiting to be helpful and get the phone number your wanted, answering machines, automobiles controlled by computers under the hood are all examples of their expanding range of application - you have only to watch and note the universality of computers in your life. Of course they will further increase as time goes on - the same simple general purpose computer can do so many special purpose jobs that it is seldom that a special purpose chip is wanted.

You see many more special purpose chips around than there need be. One of the main reasons is that there is a great ego satisfaction in having your own special chip and not one of the common herd. (I am repeating part of Lecture 2.) Before you make this mistake and use a special purpose chip in any equipment ask yourself a number of questions. Let me repeat the earlier arguments. Do you want to be alone with your special chip? How big a stock pile of them will you need in inventory? Do you really want to have a single, or a few, suppliers rather than being able to buy them on the open market? Will not the total cost be significantly higher in the long run?

If you have a general purpose chip then all the users will tend to contribute, not only in finding flaws but having the manufacturer very willing to correct them; otherwise you will have to produce your own manuals, diagnostics, etc., and at the same time what others learn about their chips will seldom help you with your special one. Furthermore, with a general purpose chip then upgrades of the chip, which you can expect will sort of be taken care of mainly by others, will be available to you free of effort on your part. There will inevitably be a need for you to upgrade yours because you will soon want to do more than the original plan called for. In meeting this new need a general purpose chip with some excess capacity for the inevitable future expansion is much easier to handle.

I need not give you a list of the applications of computers in your business. You should know better than I do your rapidly increasing use of computers, not only in the field but throughout your whole organization, from top to bottom, from far behind the actual manufacturing up to the actual production front. You should also be well aware of the steadily increasing rate of changes, upgrades, and the flexibility that a general purpose symbol manipulating device gives to the whole organization to meet the constantly changing demands of the operating environment. The range of possible applications has only begun, and many new applications need to be done - perhaps by you. I have no objections to 10% improvements of established things, but from you I also look for the great new things that make so much difference to your organization that history remembers them for at least a few years.

As you go on in your careers you should examine the applications that succeed and those that fail; try to learn how to distinguish between them; try to understand the situations that produce successes and those that almost guarantee failure. Realize, as a general rule, that it is not the same job that you should do with a machine, but rather an equivalent one, and do it so that then future, flexible, expansion can be easily added (if you do succeed). And always also remember to give serious thought to the field maintenance as it will actually be done in the field - which is generally not as you wish it would be done!

The use of computers in society has not reached its end, and there is room for many new, important applications. They are easier to find than most people think!

In the two previous Lectures I ended with some remarks on the possible limitations of their topics, hardware and software. Hence I need to discuss some possible limitations of applications. This I will do in the next few Lectures under the general title of Artificial Intelligence.

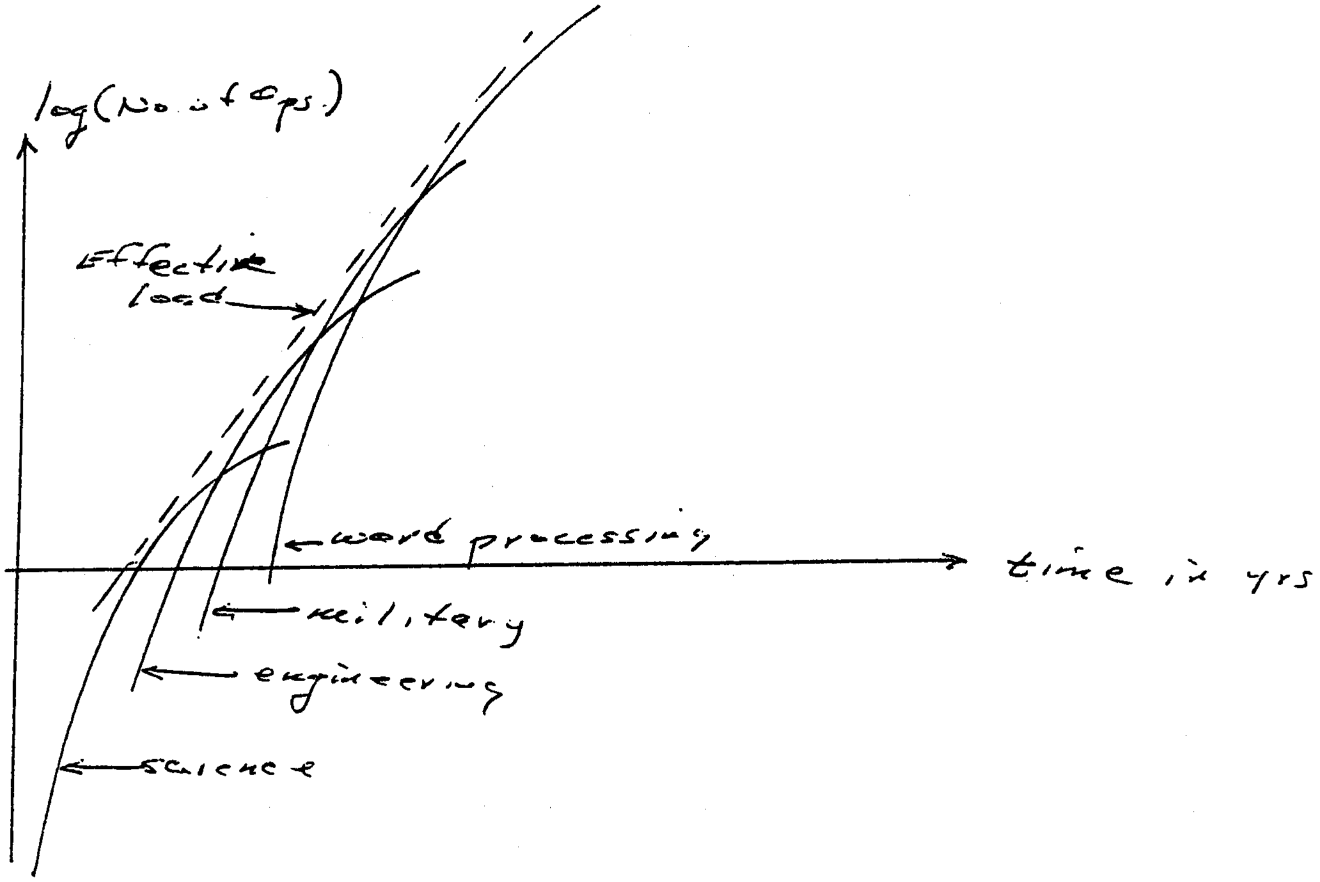
log(No. of ops.)
Effective load
word processing
time in yrs
military
engineering
science

Figure 5-1