Box 7
folder 12

2 of 3

102739347

ACM Conference on the History of Scientific and Numeric Computation

Conference Proceedings

Papers Presented at the Conference

Princeton, New Jersey
May 13–15, 1987

Sponsored by the

Allegheny Region of the
Association for Computing Machinery

in cooperation with ACM SIGNUM
and the Society for Industrial
and Applied Mathematics

ACM Conference on the History of Scientific and Numeric Computation

Conference Proceedings

Papers Presented at the Conference

Princeton, New Jersey
May 13–15, 1987

Sponsored by the

Allegheny Region of the
Association for Computing Machinery

in cooperation with ACM SIGNUM
and the Society for Industrial
and Applied Mathematics

The Association for Computing Machinery
11 West 42nd Street
New York, New York 10036

ISBN 0-89791-229-2

A limited number of copies may be ordered prepaid from:

ACM Order Department	*Price*
P.O. Box 64145	Members $11.00
Baltimore, MD 21264	All others $15.00

ACM Order Number: 701870

Conference Committee

General Chairman and Program Chairman

Gene H. Golub
Stanford University

Conference Organization and Local Arrangements

Frank L. Friedman
Temple University

Conference Support

Susan J. Foster
Temple University

Proceedings

George E. Crane

Registration

Pegotty Cooper
Association for Computing Machinery

Video Taping

Bell of Pennsylvania

ACM Headquarters Coordination

Judi Taylor
Association for Computing Machinery

Acknowledgements

This is the second in a series of ACM sponsored conferences on the History of Computing. The first conference, on the History of Personal Workstations, was held in Palo Alto, California, in January, 1986, and was chaired by Alan Perlis of Yale University and John R. White of Xerox PARC. The Personal Workstations Conference served as both an inspiration and a guide for the organization of this conference; Perlis' perspectives on the history of technology development and White's willingness to share his experiences in running the first conference provided highly valuable guidance in the planning for this conference.

The credit for the program all belongs to Gene Golub. Gene's network of friends and colleagues in the field is incredibly extensive and he knew exactly whom to invite to produce an exciting, high quality program. His very personal concern for his colleagues, his understanding of the relevance and importance of their work and his recognition of the importance of bringing together the pioneers in the field of scientific and numeric computation were the major factors in putting together the outstanding list of speakers for the conference program.

Gene and I are most grateful for George Crane's assistance with the program and for his work in putting together this proceedings. He coordinated the entire production effort for the proceedings which was a time consuming, major communications task. We thank George for his most significant contributions to this conference.

I would also like to thank Adele Goldberg who, as President of in 1985, had the vision and foresight to launch the ACM History Conference series. Now as Past President of ACM and as the History Series Advisor to the ACM Conferences Board, Adele has played a leading role in designing the plan for the series and in providing the ideas and enthusiasm to carry out this plan. Her unflagging support has been invaluable.

Others who have played a major role in the planning and organization for the conference include Ed Block, the Executive Director of SIAM, Susan Foster of Temple University, and Donna Baglio, Judi Taylor, Pegotty Cooper and Lynne D'Addesio of ACM. Ed provided encouragement and assistance at the early stages of our work and ensured us of direct access to the resources at SIAM Headquarters in Philadelphia. Susan, Donna, and Judi provided the financial, publicity and local arrangements support required to put together the many and varied pieces required to run a conference. Lynne worked on the final stages of the proceedings preparation, and Pegotty handled conference registration. Pegotty, Judi and Donna also provided considerable assistance with the overall management of the conference details and planning milestones.

Finally, on behalf of everyone involved with the History of Scientific and Numeric Computation Conference, I would like to thank Bell of Pennsylvania for contributing the video-taping support for the conference, and the U. S. Army Research Office (Mathematical Sciences Division), the U. S. Army Ballistics Research Laboratory, the National Science Foundation (Division of Computer and Computation Research, Division of Social and Economic Science), for their financial support of the conference.

Frank L. Friedman
Temple University

ACM Conference on the History of Scientific and Numeric Computation

MAY 13–15, 1987

WEDNESDAY, MAY 13, 1987

Opening Session 8:30–10:15

Break

Session 2 10:45–12:30

LUNCH 12:30–2:00

Session 3 2:00–3:45

Break

Session 7 2:00–3:45

Break

Session 8 4:15—6:00

FRIDAY, MAY 15, 1987

Session 9 8:30–10:15

Break

Session 10 10:45–12:30

REMEMBRANCE OF THINGS PAST

Herman H. Goldstine
American
Philosophical Society

I am very sensible of the great honor that you have bestowed upon me by asking me to give the opening paper in what I am certain will prove to be a most useful and productive symposium on the topic of scientific and numeric computation. In order to comply with Gene Golub's request I have been forced once again to think over what these terms mean now and what they have meant throughout time. At least in my case this is usually a worthwhile task requiring me to reappraise the subject and ask myself if this is what was intended by the fathers of the field.

I think that with very many branches of mathematics we can well ask the perfectly proper question: What is the purpose of this subject? Why did its creators choose to go in this rather than in some other direction. After all even though mathematics is very largely a magnificent creation of the human intellect it is not merely a collection of complicated but arbitrary topics lumped together in an inchoate whole. We know that there are remarkable threads and themes that run through many of the topics and that many others are there to provide us with the tools needed to make yet other studies. The unities present are remarkably abundant and the sense of arbitrariness that people sometimes mention seems to me often a reflection on their lack of understanding of the topics in question.

At this point it is perhaps relevant to quote some of von Neumann's views on mathematics and mathematicians. He said: "Most people, mathematicians

and others, will agree that mathematics is not an empirical science, or at least that it is practiced in a manner which differs in several decisive respects from the techniques of the empirical sciences. And, yet, its development is very closely linked with the natural sciences. One of its main branches, geometry, actually started as a natural, empirical science. Some of the best inspirations of modern mathematics (I believe, the best ones) clearly originated in the natural sciences..."

The subject of mathematics is however very different from say theoretical physics and it is perhaps worth pausing for a bit to understand just how this is so. As we know mathematics falls very naturally into a large number of more or less distinct fields and almost no one today has any reasonable grasp of the whole. On the contrary physics seems to be a very different sort of topic; a crucial difficulty is met in the experimental area and whatever anomaly this presents must be cleared up before the practitioners of the field can go forwards. It is not possible for them to do what we very often do: drop the problem as being intractable and proceed to an entirely different challenge. As we can appreciate certain critical experiments in the real world cannot be ignored if their results contradict existing theories. All the best scientists in the field are forced to face up to the challenge and to make whatever modifications are necessary to reestablish equilibrium in their science. Thus experiments such as Michaelson's led to the introduction of special relativity and the conflict between that subject and classical celestial mechanics to general relativity.

Let us look back at the beginnings of our subject which we have to seek in the works of Hipparchus and Ptolemy who worked in the period from about 150 BC to 150 AD. Obviously they were not the first men to make significant use of mathematics, but they did make such use. The great geometers many of whose names have been lost to us through the remarkable efforts of Euclid to pull together all the empirical, semi-empirical, and pure mathematical efforts in geometry certainly developed one of the most noteworthy structures in the ancient world. We need not concern ourselves here with how much was empirical and how much purely mathematical. All that we need to know is that Hipparchus and later Ptolemy used the Euclidean apparatus to explain the motions of the heavenly bodies with excellent accuracy. I feel that it was they

and perhaps the latter who were mainly responsible for the initiation of our subject. Ptolemy was faced at the beginning with the problem of explaining the motions of the visible planets, the sun, and our moon with sufficient accuracy so that an observer armed with the astronomical instruments of that day could locate the body in question. The paper construct that Ptolemy created in his Almagest is in some ways like an elaborate mechanical device or rather a series of these devices, one for each of the visible planets, the sun, and the moon. They are made out of circles with smaller circles mounted on their perimeters. Each of these was, so to speak, hand made so that the particular body moved in accordance with observational data which in many cases went far back in time and enabled Ptolemy and his colleagues to determine many parameters with considerable exactitude.

Ptolemy did not of course develop the basic mathematics which he used to explain or rather to predict the locations and times of various celestial events. He obviously decided that he would accept the mathematics which was available at that time, Euclidean geometry, and went on to develop a means for using it in a practical way to give results in numerical form. The apparatus that he and Hipparchus put together is what we call trigonometry. Its utility I need hardly mention has been so great that it survived as a standard topic in the curriculums of schools for almost two millennia. Let me hasten to point out that there are very few things in our magnificent western culture which have such survival times. Therefore let us not sneer at this subject. Ptolemy was faced with two real choices when he realized his need for a computational tool. He saw that a table of the sines -- actually the subtended chords -- of a series of equally spaced angles was just what would do what was needed. This is quite clear but what I think is very remarkable is that he did not measure these chords or sines by physical means but instead developed the lovely relations of trigonometry and coupled these with the knowledge of the number of degrees in the angles of certain regular polygons such as 30°, 45°. By these means he was able to build up virtually all the needed entries in a table of sines with a half degree spacing. He needed however one more thing: the sine of 1/2°. To do this he developed a very neat scheme for interpolation based on an elegant inequality of Archimedes which says that if $A > B$ then $A/B > \sin A/\sin B$. He applied this to obtain

the result

$$(2/3)\sin(3/4) < \sin(1/2) < (4/3)\sin(3/4).$$

This gave him sin 1/2 with a relative error of 2×10^{-6}.

While this gave Ptolemy his table, it gave us a whole way of viewing mathematics. It meant that the scientist who wants to explain or discuss the world around him or some aspect of it need not go off into an empirical or experimental study but that he can at least first try to see whether there is not some mathematical tool available to use instead. This has reduced the need for experimentation to the determination of physical fundamentals such as physical constants whose values are very properly the subject for experimentation. If Ptolemy had not seen how to use mathematics to fill in his table but had constructed various sized angles and actually measured chords, heaven knows what applied mathematics in general and computation in particular would have become.

In any case so great was this success through its remarkable predictive powers that it became and has continued to be a desideratum of virtually all sciences to try to emulate applied mathematics by becoming more mathematical in nature. So for example we see that some of the very great advances in theoretical physics have been made possible at least in part by the very highly mathematical form that the subject has assumed.

Let us now momentarily return to Ptolemy. Of course we can now retrospectively say that the notion of making physical measurements is silly and that no one in his right mind would have resorted to that technique when so much Euclidean geometry was available. This argument is not necessarily as convincing as it now appears at first blush. I can imagine that a lesser man than Ptolemy might at least have had a dreadful time with the interpolation scheme that he developed for his purposes. Perhaps it was that the Grecian world already knew and appreciated the power of applied mathematics because of the great exploits of Archimedes, the Hercules of mathematics.

There is a basic point worth noting before we leave the topic of Ptolemy and that is that his work and that of his great predecessor Hipparchus was on astronomy. Our particular subject has been for a long time very much a hand maiden of the mathematical astronomer. In this same connection the next figure that I

Alan Perlis
Xerox, PARC
3333 Coyote Hill Road
Palo Alto, CA 94304
(415) 494-4458

6/25/87

Adele,

Thank you. I enjoyed reading everything that was there.

I hope the abstracts will surface in a final publication, since it would be a shame not to have them.

Again, thanks very much for letting me look at your personal copy.

AJ

should like to mention to you at least in a passing way is Napier. As you probably know this Scot was an astronomer who felt keenly the need for a better way to undertake the onerous tasks which faced him in his studies.

It is perhaps not without some interest to note what a distinguished Arab astronomer al-Kashi (1400) who lived during the time of Tamerlane in Samarkand did in his observatory. He was concerned with seeking a more elegant way to find the sine of 1/2° than Ptolemy had produced. To this end he noticed that there is a very neat cubical relation between the sine of 3A and the sine of A:

$$\sin 3A = 3\sin A - 4\sin^3 A$$

so that if he had the sine of 3° he could then find the sine of 1°. This led al-Kashi to develop an interactive scheme for solving the cubic and very likely led to the subject that was known as the Theory of Equations. This was a field that was often taught at elementary level in a number of universities. One of the most noteworthy topics in that field at least for me when I was a student was the so called Newton - Raphson method for iteratively solving functional equations.

But let us return to Napier and say just a word or two about his logarithmic function and his tables. He defined his function in the following words: The logarithm of a given sine is that number which has increased arithmetically with the same velocity throughout as that which radius begins to decrease geometrically, and in the same time as radius has increased to the given sine. He further says that the logarithm of r is 0. We thus see that

$$\text{Nap Log } x = r\log_e r/x.$$

To quote Shakespeare "It needs no ghost come from the grave to tell" us that this Naperian logarithm is deficient in that the Naperian log of a product is not equal to the sum of the logs of the individual factors but there is an extra term, the log of 1 which enters.

The other matter that bothered Napier about his logarithm was that it was not easy to calculate powers of 10. He therefore proposed to Henry Briggs, an English friend, that he work out the logarithms which we today call Briggsian. Briggs not only did this but he also worked out a very elegant method of interpolation that is really still impressive today.

Let us now leave this very ancient history and move forwards into merely ancient history, and let us discuss my doings. Back in the days before the second World War Gilbert A. Bliss at

Chicago was interested in exterior ballistics and announced a course in the topic. He also was planning to write a book on the subject which he in fact did. But the teaching of his graduate courses had fallen to me in those days because his health was uncertain and I was very fortunate. In the course of teaching the students at Chicago I had to take them through a certain amount of numerical analysis so that they could learn how to solve the differential equations of motion for a projectile - fuze combination. This was a skill that I had acquired more or less painfully from an astronomer at Chicago named Walter Bartky. We had tables of logarithms and little else besides a method first generally named after Adams and Moulton.

This method and ones similar to it which played a major role at the Ballistic Research Laboratory at the Aberdeen are all characterizable by saying that they involve calculating and recording on paper many differences since linear operations are cheap to perform by hand and paper for storage of partial results is very inexpensive. These methods make use of as few no - linear operations such as multiplications and divisions as possible since these involve the use of log tables and entail a lot of table look ups and interpolations.

When therefore I arrived at Aberdeen and was assigned to the department that had to produce all the Army's and the Air Force's firing and bombing tables I found myself back home again with the same techniques as I had been teaching young people at Chicago. Fortunately from my point of view I was put in charge of a substation of the laboratory at the University of Pennsylvania's Moore School of Electrical Engineering which allowed me to be in touch with several men who were very keen on the practical engineering level with the problem of automating a very dull subject capable of being done better by machine than by human. In fact the Moore School staff included a number of faculty and at least one graduate student who were very much involved in precisely this topic and had been for some years in connection with an analog computer called a differential that had been built at the school in the mid 1930's and a copy made for Aberdeen. This was in fact one of the reasons why Aberdeen and the Moore School were contractually related during the war.

The differential analyzer was an electro - mechanical device invented by Vannevar Bush in the early 1930's to

integrate differential equations which customarily arose in the field of electrical engineering in those days. The equations for the motion of a projectile were readily adaptable to these machines and afforded a fast but not accurate way to solve them. Their accuracy was not high; in fact about 5 in 10,000 was about the best one could get. It took about 10 to 20 minutes to integrate the average trajectory. To understand what this involved let me remark that such a trajectory involved about 750 multiplications and would take a human at least 7 man - hours. Our main aim in life was to bring this 10 to 20 minute time down by an order of magnitude and to provide at the same time a nonhuman way to perform all the interpolations and other numerical steps that were needed to produce a firing table.

Fortunately for me J. Grist Brainerd, then a young professor at the Moore School proposed to me a solution to the problem first raised by a colleague of Brainerd's named John W. Mauchly to build an electronic digital computer to replace the differential analzyer and bring two enormous advantages to us: the speed of electronics and the accuracy of the digital principle. The Army accepted this proposal and the Moore School under Brainerd's aegis with a young and superb engineer named J. Presper Eckert actually built the device, the ENIAC. It is not my place here to spend more time on the details of this essential advance in our field. Suffice it to say that it immediately changed the face of the computational world.

Since the ENIAC had an incredibly small memory and its successor machines built in a number of places had very small memories for intermediate results the entire economy of computing changed overnight. Instead of being in a world of expensive multiplication and cheap storage we were thrown into one in which the former was very cheap and the latter very expensive. (In fact we are only now getting into an economy where storage or memory is becoming exceedingly cheap.) This meant that virtually all the algorithms which man had devised for carrying out calculations needed reexamination but also many areas of numerical analysis such as the numerical solution of partial differential equations were suddenly potentially open to us. This was the world in which we found ourselves at the end of the second World War.

It was into this world that Johnny von Neumann projected himself with the gusto and élan that characterized all his activities including eating. Either

he went at something with full speed ahead and damn the torpedoes or not at all. Nothing was ever so complete as the indifference with which Johnny could listen to a topic or paper that he felt he did not need to hear.

At this time in Johnny's history he was "gung ho" for the wonderful world that the electronic computer was opening up. We decided that we should set up at the Institute for Advanced Study a full scale effort to have a major hand in creating this brave new world. To do this we instituted what we called the electronic computer project and decided that our thrust needed to be multi - pronged.

We accordingly had a group devoting itself to what might now be called computer architecture and science. Here our main aim was to discover the right way to organize or structure a computer so that it would be flexible and easily responsive to its users. This effort resulted in a series of papers on planning and coding of problems which had I immodestly claim a fundamental role in shaping the architecture of the modern computer. We also pushed in a small way into topics such as meshing and sorting of data and into the question of the least number of operations needed to perform a given function.

Another group which we had was one devoted to numerical methods and we shall say more on this as we proceed. A third group was created to do the engineering and fabrication of a computer embodying our architectural ideas. As you might suppose the results of this were transitory; the changes taking place in the engineering field were so great that the machine was perhaps obsolescent within a year or so of its completion.

Finally we envisaged a group that would use the results of the others to solve some important problem or problems that the whole outside community including even the lay public could grasp to show the significance of the electronic computer to the world around us. Johnny chose the field of meteorology and set up a first rate group of men around Jules Charney who formulated the equations for the motion of climatic phenomena as partial differential equations. They of course had to make many simplifying assumptions both to formulate the problem and to get it into a size that our computer could calculate the motion of the weather at speeds in excess of the real speed so that forecasting into the future became possible.

It is not our business here to discuss the details of this project beyond remarking that the results of that effort

were taken up by the Weather Bureaus of all the leading nations of the world. In fact you may know that here in Princeton there is a laboratory established by our Weather Bureau which devotes its activities to trying to extend knowledge so that accurate long range forecasts will become possible.

Let us now take up some of the things that engaged our attention during the period from 1946 to 1957 and which relate to our field. Obviously one of the first and most likely topics to be discussed was the solution of large systems of linear equations since they arise almost everywhere in numerical work. V. Bargmann and D. Montgomery collaborated with von Neumann on a paper on this subject. Then H. Hotelling who was a well known statistician of that era wrote an interesting paper in 1943 which he studied a number of numerical procedures including the Gaussian method for inverting matrices. He pointed out in a very heuristic and as it turned out, inaccurate way that the Gaussian method for inverting statistical correlation matrices would require about k + 0.6n digits during the computation to obtain k digit accuracy. Thus to invert a matrix of order 100 would in his terms require 70 digit be used if one wanted 10 digit accuracy.

Johnny and I never quite believed that Gauss would have used a procedure so lacking in elegance when one thought for a few moments about his great love for computation. Indeed his collected works contain a considerable amount of material both on astronomy and on geodetic work which shows his love for and great skill at calculation. As some partial evidence of this we know he certainly used the so called Cooley Tukey method to handle Fourier transforms. Taking his skill as a given we looked closely at the procedure and wrote a paper on the subject which we used as a vehicle to introduce an elaborate introduction on errors in numerical calculation. We tried in that paper to alert the practitioners in the field to a phenomenon which had not been particularly relevant in the past and which was to be a constant source of anxiety in the future: numerical instability. In the course of the analysis we also brought to the fore the notion, now obvious, of well and ill conditioned matrices. Since then of course people such as Wilkinson have greatly simplified the very complicated analysis we went through to arrive at our final results.

In a second paper we raised a question which we thought might become more important than it in fact ever became.

We said let us not worry so much about what might happen in a very small number of pathological cases; instead let us see what occurs on the average so that if we need to do this same task very many times what we can expect. To achieve this probabilistic result I had to develop proofs for several theorems in probability theory which I did with considerable difficulty only to receive a letter from a statistician named Mulholland after the paper appeared in which he showed me how to do one part with the slightest work: a mere flip of his wrist sufficed to demonstrate some obvious thing. My only consolation was that Johnny had not seen how to do it simply either. In the event I suppose that our second paper scared practitioners of the subject away from the field of probabilistic estimates instead of bringing them in, or perhaps it simply was not a very important idea. Human egotism being what it is I naturally hope it was the former but honesty makes me think it was the latter.

The other thing that one might reasonably want to know about a symmetric matrix are its eigen values or as Veblen used teasingly to say its proper Werte. At that time we had in Princeton for a term Frank Murray, a mathematician from Columbia who had collaborated with von Neumann at one period on operator theory. The three of us set ourselves the goal of considering all reasonable ways that one might find the eigen values and discover which seemed the best in the sense of numerical stability. We made an extensive search and came up with one which pleased us very much. Since I seem to have had some priority or other on this scheme it was agreed that I would present it at a 1951 meeting to be held at UCLA where the National Bureau of Standards had a western numerical institute. In the event I presented the paper which was very well received and then Ostrowski got up and asked me if I knew that this method had first been worked out by Jacobi in 1846. Of course the answer was no. Jacobi was interested in finding a better way to analyze some data of Leverrier in the Connaissance des temps and did it by finding the eigen values of a symmetric matrix of order seven. His results significantly improved Leverrier's. I shall not discuss the improvements that Householder and then Givens made to our knowledge of how to find eigen values.

Instead I must turn now to the field of partial differential equations and say some words on this topic. You will of course be hearing from several people who are much more learned than I in the

numerical solution of such equations and who collaborated with Johnny on this topic during his lifetime and continued to make major thrusts after his death. One of his early interests was hydrodynamics which he understood profoundly. I must tell you that some, indeed perhaps most, applied mathematicians know a great deal about the mathematical tools that they can use to solve problems but have little deep knowledge of the physics, chemistry, biology or what have you that underlies their subject. Not so Johnny. His grasp of the physics, the theory, the apparatus, and the experiments were all food for his interest. It is this which made his interest in the computer so profound. He was very concerned about the electrical characteristics of each type of vacuum tube, of what resistors, capacitors, and inductances were made and why. One had the impression that when he entered a field he had to encompass it all, however elaborate it might be.

In any case he was one of the very few people outside of the three authors who knew the 1928 paper on the solution of partial difference equations. Here Courant, Friedrichs, and Lewy considered how to solve partial difference equations and in the course of their analysis based on the characteristic curves of hyperbolic difference systems showed that certain inequalities had to be satisfied. They now go by the name of Courant conditions as you well know. In any case Johnny was a consultant to a variety of places including Los Alamos where his expertness in hydrodynamics, among many other things, was of great value. He was an apostle there for numerical calculation and gathered around himself a group of very keen physicists, including Nick Metropolis, who became his followers. His object all sublime was in so far as possible to replace experimentation in fields where the equations for a problem could be unambiguously formulated by numerical calculation. He even did this using Howard Aiken's electromechanical machine at Harvard to show the feasibility of such procedures.

His enthusiasm and vitality were so great in this connection that I agreed to let Los Alamos put on the ENIAC for its test calculation a huge problem for those times. The task was horrendous: people such as Metropolis and his then colleague S. Frankel worked like mad to get results. Whether this particular calculation was of any real use to Los Alamos I never asked but it certainly started that laboratory and all other Atomic Energy Commission laboratories taking a vital interest in numerical work.

A look at von Neumann's collected works will show the most casual reader how much effort he and his collaborators such as Goldstine, Metropolis, Richtmyer, Taub, Ulam, and others put into hydrodynamical calculations. This meant in effect studies of hyperbolic and parabolic partial differential equations. One of the most interesting things for von Neumann in the study of hyperbolic equations was the truly anomalous and remarkable emergence of shocks -- discontinuities -- in otherwise thoroughly smooth situations brought about by very slight and continuous motions. A number of papers of his relate to precisely this point. One that I recall we wrote was on an analysis of what happens if a very powerful explosion takes place at a point in a homogeneous medium. The result is a spherical blast wave which emanates out from the point. The shock was handled by making use of an iterative procedure originally due to Peierls for solving the Rankine - Hugoniot equations. Another intriguing method for coping with shocks was developed by Johnny and Robert Richtmyer who conceived of the idea of introducing arbitrarily into an otherwise inviscid fluid some viscosity. This is the same thing as introducing into the equations being considered some artificial dissipative terms which serve to give the shocks a thickness roughly comparable to the mesh size of the numerical net. This changes the shocks into near discontinuities which propagate at essentially the right speeds and across which the temperatures and pressures change by nearly the right amounts. This meant that one could totally ignore the Rankine - Hugoniot equations and proceed in a very simple numerical fashion.

Von Neumann's interest in hydrodynamical and related sorts of calculations arising at Los Alamos and other places where nuclear particles were under study also resulted in the development of a lovely and perhaps totally unexpected gem of a field: Monte Carlo.

This was a nice example of von Neumann's combining interests in a number of subjects. He saw here how Newton's brilliance had enabled man to express in continuous form equations relating discrete particles so that instead of horrible systems of unmanageable equations one could write down a few elegant conservation relations and solve the equations that they embody. In fact the numerical revolution caused the analyst to replace the continuous equations by systems of discrete ones. Johnny and Ulam got the idea of returning to finite systems and playing repeated

games according to the rules of probability theory.

Instead of saying more on this, perhaps I can just mention some work that we did on a conjecture of Kummer's. This was part of an idea that we had of using the computer as a new and improved form of scratch pad to develop examples and counter - examples. Artin had mentioned to us this conjecture of Kummer's which was based on a very few -- in fact on 45 -- cases. Artin felt that it was too difficult to undertake a proof of the conjecture without more evidence of its truth. We accordingly ran a test for about 10,000 values and found that there was little evidence from our results to justify Artin or anyone else from undertaking a major effort to try to establish the result. Subsequently others have run further tests but I no longer remember with what consequences.

I often think that in addition to all the individually remarkable things that Johnny did in our field he also did something which may almost be more important. This is a matter that I have skirted in what I have said to this point and which I find very difficult to discuss without someone thinking that I am making a pejorative remark. I believe that von Neumann's great status in the world of the physical and social sciences was sufficient so that when he told people to compute digitally and not to make analog computations by means of various sorts of physical experiments they believed him. I think that this in large measure accounted for the early acceptance of the digital computer. I do not imply by my remark that it was necessary for the ultimate use of the computer by the scientific world at large; I simply mean that he caused it all to happen at a rate which was much accelerated over what it would have been had he not influenced the field so decisively. I should like to give two examples of this: young Tom Watson, Jr. was just back from being a pilot in the CBI area and having heard of Johnny and his interest in electronic computing came to the Institute for Advanced Study to see for himself what the new world was all about. I feel very certain that this had an extremely important impact on IBM and hence on the world at large. The other example arose from the fact that Johnny after becoming a commissioner of the AEC exerted great influence on the laboratories of the Commission to use computers and to authorize both IBM and Sperry - Rand to undertake a sort of competition which resulted in two monster machines for their era -- the Stretch and the Larc computers. Out of these many great advances in our modern world arose.

Instead of continuing further I think that this is perhaps a good point for me to stop and to turn over the floor to my colleague and collaborator while he discusses what we today call the earth sciences.

"The three main areas of geophysics are, of course, air, water and earth. Let me begin with the air, i.e. with the phenomena in the atmosphere. I am referring to dynamical, or theoretical meteorology. This subject has for a number of years been accessible to extensive calculations. It is, therefore, worthwhile to estimate what NORC could do in this area.

We know today, mainly due to the work of J. Charney, that we can predict by calculation the weather over an area like that of the United States for a duration like 24 hours in a manner, which, from the hydrodynamicist's point of view may be quite primitive because one need for this purpose only consider one level in the atmosphere, i.e. the mean position of the atmosphere.

We know that this gives results which are, by and large, as good as what an experienced "subjective" forecaster can achieve, and this is very respectable. This kind of calculation, from start to finish, would take about a half minute with NORC.

We know, furthermore, that this calculation can be refined a good deal. One cannot refine the mathematical treatment ad infinitum because once the mathematical precision has been reached a certain level further improvements lose their significance, since the physical assumptions which enter into it are no longer adequate.

In our present, simple descriptions of the atmosphere this level, as we know, is reached when one deals with approximately three or four levels in the atmosphere. This is a calculation which NORC would probably do (for 24 hours ahead) in something of the order of 5 to 60 minutes.

We know that calculations of meteorological forecasts for longer periods, like 30 to 60 days, which one would particularly want to perform, are probably possible but that one will then have to consider areas that are much larger than the United States. In a duration like 30 days -- in fact in much shorter durations, like 10 - 15 days -- influences from remote parts of the globe interact. We also know that interaction between the Northern and Southern Hemispheres is not very strong. Therefore, one can probably limit the calculation in the main to one

entire hemisphere, but not to a smaller area.

Such calculations have so far only been performed in tentative and simplified ways and all those who have worked on these problems have done so in the sense of a preliminary orientation only.

One of the main reasons for going easy about this problem is that with the best available modern computing machines it is still a very large problem, and when one deals with a new problem one must solve it a few dozen times the "wrong way" before one gradually finds out by trial and error, and by coming to grief many times, what a reasonably "good way" is. Consequently, one will simply not do it unless one can obtain individual solutions quite rapidly.

A calculation of this order on NORC would, I think, require something of the order of 24 hours' computing time. This can be off by a factor of perhaps two, one way or other, but in any event this order of magnitude is acceptable for research purposes.

In this area, therefore, an instrument like NORC becomes essential at about this latter level. Indeed, whether one does a simple 24 - hour forecast in half an hour or in two minutes is not decisive. But in a 30 day hemispheric calculation it is very important whether one needs 24 hours or a month. If it takes a month one will probably not do it. If it takes 24 hours, one may be willing to spend several months doing it 20 times, which is just what is needed.

I will now pass to the second area, i.e. to calculate relative to the ocean. I will only mention one thing, which I think now has become possible. There has always been a need for this and with a machine like NORC it can now be done. I am referring to complete calculations of the ideal motions in the entire oceanic system.

This will have to be done in two parts, namely, first for the large body of the high seas; and secondly for the marginal phenomena which are the primarily interesting ones, i.e. the events near the continents. In the first calculation, one will have to treat the phenomena close to the continents summarily (at low spatial resolution), and then use these primary results for the high seas as "external boundary conditions" for the detailed calculations on limited areas near the continents.

I will not try to put numbers in hours and days on calculations of this type, because one has to go into considerable details of evaluation before one can quote such figures meaningfully. However,

it seems that with a machine like NORC, it will be for the first time that such a calculation becomes a matter of days only, and therefore, with all the trials and errors that will be inevitable due to our general ignorance, it becomes practical at this point only.

Next, I am coming to the third area, to things which relate to the earth. The following problem is typical. It has been realized for some time (by Bullard and Elsasser) that the hydrodynamics of the liquid core of the earth are of very great importance, in particular in explaining terrestrial magnetism. It is also found that the liquid core of the earth is in a very complicated state of motion, where mechanical and electromagnetic forces both play about equally important roles, and that this motion belongs to a very difficult class known as turbulent.

Calculations dealing with this motion are difficult and complicated, and can probably not be reduced by any idealizations to less than three dimensions. The pioneer efforts in this field have made it possible to see what calculations will be necessary here. It seems clear that this class of problems too becomes accessible to exhaustive and direct calculation now for the first time.

Reminiscence of Howard Aiken

I. E. Cohen
Harvard University

I should like to present a few aspects of what seem to me to be the significant parts of Aiken's career, giving also some indication of the simultaneous rise of an interest in large-scale digital computation -- a phenomenon that arose independently in quite different parts of the world. I have reference here to Aiken, Atanasoff at Ohio State, Stibitz at the Bell labs, Zuse in Germany, plus the Eckert project at Columbia and Comrie's installation in London. It is not only interesting to see how demands for solution to problems were arising in such different parts of the world, but it is an awareness of what each of these had accomplished which enables us (I believe) to make precise the actual accomplishment of Aiken, which also then sheds light on the next high level of achievement represented by ENIAC followed by the stored program.

The Los Alamos Experience, 1943

N. Metropolis
Los Alamos National Laboratory

Compelling applications have provided the stimulation for the unprecedented developments of modern computing. There were a few examples in the 1930's, but the primary sources occurred during World War II, starting with "hand computing" using desk calculators, followed by electromechanical devices, and then the revolutionary transition to electronic computers. An account of this experience at Los Alamos is given.

EARLY NUMERICAL ANALYSIS IN THE UNITED KINGDOM

L. FOX
Emeritus Professor, Oxford University

1. Introduction

Rumour has it that the term "Numerical Analysis" was coined sometime in the late nineteen forties by the numerical statistician J.H. Curtiss at the National Bureau of Standards in Washington, D.C., the NBS being effectively the American National Physical Laboratory, the English version of which I shall mention a bit later. That sort of date makes numerical analysis a rather new subject, and in fact I lived through quite a lot of the early history in the UK. But in some respects the subject has a quite long history, and these I shall mention briefly.

Throughout history individuals have wanted numerical solutions for simple problems like the volume of a rectangular solid with given sides to very complicated matters like the determination of the position in space at a particular time of a vehicle launched from a specified point on earth. Such individuals are not really numerical analysts, and I think of them as engineers or scientists. But then there are others, perhaps of a more mathematical bent, who decide that they can help the scientists in general rather than in particularly numerical contexts. These are the people I do think of as numerical analysts, and indeed in the early days, before numerical analysis became a topic in a mathematics degree, at least in the U.K. assistance to the scientists was a very important motivation for their work. One of the earliest such operations, which continued for many years, was the construction and publication of mathematical tables.

2. Table making

When all arithmetic was done by pencil and paper, multiplication and division, at least, were tedious and time-consuming operations. To ease this some early mathematical tables were produced which gave the results of multiplying any number say up to four figures by any other such number. Allied tables of reciprocals helped with a corresponding division operation to a certain level of accuracy which self-respecting tables would discuss in a suitable introduction. The invention of logarithms more or less eliminated the need for multiplication and division, and many tables of logarithms were produced by numerical analysts, differing mainly in the selected arguments and the number of figures given in the tables.

Other functions of integers were found to be useful, and one of the very first books of such tables was first published in 1814 by Barlow, and recast in two more modern editions in 1930 and 1941 by the famous English table-maker L.J. Comrie. The last edition gave n^2, n^3, $n^{\frac{1}{2}}$, $(10n)^{\frac{1}{2}}$, $n^{\frac{1}{3}}$ and n^{-1} for $n = 1(1)12500$, with extra attention for reasonably small n for n^4, $n!$ and $n^{-\frac{1}{2}}$, integer powers up to n^{10} for $n = 1(1)100$ and up to n^{20} for $n = 1(1)10$, binomial coefficients for $n = 1(1)12$, and a list of useful constants. The non-exact numbers have 7,8 or 9 significant figures, with some facilities for interpolation and a relevant description thereof in the introduction.

Barlow's 1814 preface makes interesting reading, the following being part of it, with address The Royal Military Academy, Woolwich (July 1, 1814).

"In presenting the following Mathematical Tables to the attention of the public, the far greater part of which are the result of laborious calculation, little need be said to prove that I have not had in view the accomplishment of any pecuniary object, as the time employed in the computation, the expense of publication, and the limited number of purchases which from the nature of the subject is to be apprehended, preclude any idea of adequate remuneration. And as little is to be expected of mathematical reputation, nothing more being requisite for the execution of such an undertaking than a moderate skill in computation and a persevering industry and attention; which are not precisely the qualifications a mathematician is most anxious to be thought to possess.

"In fact the only motive which prompted me to engage in this unprofitable task was the utility I conceived might result from my labour; and if I have succeeded in facilitating any of the more abstruse arithmetical calculations, and thereby rendered mathematical investigations more pleasant and easy, I have obtained the principle object I had in view."

As applied mathematicians developed their skills their computational problems became increasingly complex, and more advanced mathematical tables were needed and indeed produced. The first group included the trigonometric functions and their inverses, the corresponding treatment of hyperbolic functions, together with the increasing and decreasing exponentials and the logarithmic functions. Common logarithms of these elementary functions were also frequently tabulated for obvious purposes.

The next group included the so-called higher functions of mathematical physics, commonly occurring for example in certain methods of solving partial differential equations. For this and other purposes they included the functions of Bessel, Legendre, etc, gamma and allied functions, Weber parabolic cylinder functions, exponential and logarithmic integrals, elliptic functions, elliptic integrals and many others.

More and more numerical analysis was now needed because the calculation of relevant tabular values was no longer trivial. Morever, close preliminary attention was needed to the question of what auxiliary function or functions should be tabulated, particularly for the simplification of interpolation in difficult regions. As a simple example consider the tabulation of the exponential integral

$$-Ei(-x) = \int_x^\infty t^{-1}e^{-t}dt. \qquad (2.1)$$

For small x we have the series expansion

$$-Ei(-x) = -\gamma - \ln x + \sum_1^{\infty}(-1)^{n-1}(x^n/n.n!), \tag{2.2}$$

but the singularity at x=0 makes it desirable to tabulate the function $-Ei(-x) + \ln x$, which is not singular and interpolates nicely. For large x there is the asymptotic expansion

$$-Ei(-x) \sim \frac{e^{-x}}{x}\left(1 - \frac{1!}{x} + \frac{2!}{x^2} - \cdots\right) = \frac{e^{-x}}{x} S(x). \tag{2.3}$$

Here S can be tabulated nicely and conveniently with argument $z=x^{-1}$, and the required quantity is easily recovered.

Of course the ascending series may not be economic for good accuracy for medium-sized x, and the asymptotic series may not give the required accuracy for too small an x, and there may be a middle range in which other methods are desirable if not completely necessary. For example, for the function

$$f(x) = \int_0^{\infty}(u+x)^{-1}e^{-u^2}du \tag{2.4}$$

there are two series corresponding respectively to (2.2) and (2.3), but in a middle range of x it is more convenient to integrate by numerical methods the ordinary differential equation

$$f' + 2xf = \pi^{\frac{1}{2}} - x^{-1}. \tag{2.5}$$

Other frequent computations involved recurrence relations. For example the Bessel function $J_r(x)$, for fixed argument x and variable order r, satisfies the recurrence relation

$$J_{r+1}(x) = \frac{2r}{x}J_r(x) - J_{r-1}(x), \tag{2.6}$$

and to all intents and purposes this can be used to compute successive $J_r(x)$ for integer r, starting say with known values of $J_0(x)$ and $J_1(x)$. The other obvious task was the direct evaluation of definite integrals, and all these various operations had to be performed on desk calculating machines, sometimes with very high accuracy and always as economically as possible.

The final important topic in table-making was the systematic use of finite-difference formulae for checking computed values by inspecting differences, for sub-tabulating them as mechanically as possible to obtain other tabular values very easily, and then for providing accurate and reasonably economic methods for interpolation in the published tables. The sub-tabulation, which is systematic interpolation at a constant fraction of the original interval, usually one-fifth or one-tenth thereof, was performed quite mechanically by machines like the Hollerith punched card machine or the National Accounting Machine.

The interpolation by the user was based on finite-difference formulae typified by the Everett formula

$$f_p = (1-p)f_0 + pf_1 + E_2\delta^2 f_0 + F_2\delta^2 f_1 + E_4\delta^4 f_0 + F_4\delta^4 f_1 + \ldots, \tag{2.7}$$

where the δ^2 and δ^4 are central-difference symbols and the E and F functions are simple polynomials in p, the fraction of the (constant) distance between tabular points. Comrie found that the fourth difference could be "thrown back" into the second difference, with the explicit part of (2.7) replaced by

$$f_p = (1-p)f_0 + pf_1 + E_2\delta^2_m f_0 + F_2\delta^2_m f_1, \quad \delta^2_m f = \delta^2 f - 0.184\,\delta^4 f, \tag{2.8}$$

and that (2.8) is only very slightly less accurate than (2.7) and clearly much more convenient.

The construction of the more advanced mathematical tables and relevant publishing continued until the nineteen sixties. The main table-making activities were organised by the British Association Mathematics Tables Committee, starting about 1930, and then from 1948 onwards by the Royal Society.

Other early publications included work by Sheppard (1906) on the accuracy of finite-difference interpolation, Bickley (1939, 1941) on formulae for numerical integration and differentiation, Comrie (1931) on "throwback interpolation" and (1936) on mechanical operations with the National Accounting Machine, and Bickley and Miller (1936) and Airey (1937) on the summation of slowly-convergent series. Fletcher, Miller and Rosenhead (1946) published the comprehensive Index of Mathematical Tables. Miller (1949) wrote about table-making in general and on his solution of ordinary differential equations in particular. This he performed with the Taylor-series method, not too difficult when, as often occurred, the relevant differential equations were linear, but Miller thought nothing of using up to twelfth derivatives with a large interval of tabulation.

Miller was probably the dominant member of the relevant British Association and Royal Society committees, and much of his work appeared for the first time in the introductions to the various tables which were written singly or jointly by members of the committees and included quite important numerical analysis. Prominent in this respect is the introduction to B.A. Vol. 10 (1952), which includes the famous Miller algorithm in connexion with the recurrence relation (2.6). Miller quickly realised that the forward recurrence produced increasing inaccuracy as r increased beyond x. He solved this problem by backward recurrence with a replacement of (2.6) given by

$$\bar{J}_{r-1}(x) = \frac{2r}{x}\bar{J}_r(x) - \bar{J}_{r+1}(x), \quad \bar{J}_N(x)=0, \ \bar{J}_{N-1}(x)=1, \tag{2.9}$$

and then by scaling the computed $\bar{J}_r(x)$ to give for example

$$J_r(x) = k\bar{J}_r(x), \quad k = J_0(x)/\bar{J}_0(x). \tag{2.10}$$

For sufficiently large N this gives very good results, accuracy increasing as r decreases.

Perhaps the final useful publication was the booklet "Interpolation and Allied Tables", developed at H.M. Nautical Almanac Office. It first appeared in 1936 when Comrie was Superintendent. It was reissued at frequent intervals and with amendments and additions until the last appearance in 1956, when D.H. Sadler was superintendent. The original booklet contains finite-difference formulae of all kinds, and the 1942 edition also gave a method for solving ordinary differential equations which actually used central differences with estimation and subsequent correction, in the spirit of more modern predictor-corrector methods. A companion booklet "Subtabulation", published in 1958, gives a comprehensive version of the relevant methods developed over many years in H.M. Nautical Almanac Office.

3. Other early numerical analysis

Apart from table making and the much earlier contributions by Gauss, Newton, Runge and Kutta and Bashforth and Adams, a few other workers, particularly astronomers and theoretical scientists, suggested

numerical methods for both ordinary and partial differential equations and a few other topics. But by 1939, the start of the second world war, there was little in the way of numerical literature and numerical analysis was hardly a mathematical topic. Published in the UK, there were only a few books with a numerical content, such as Brunt (The combination of observations, 1923), Whittaker and Robinson (The calculus of observations, 1924), Steffenson (Interpolation, 1927), Scarborough (Numerical mathematical analysis, 1930), Milne-Thomson (The calculus of finite differences, 1933) and Levy and Baggott (Numerical studies in differential equations, 1934).

Scattered in the journal literature of this period were papers for example by Aitken (1926,1937) on Bernoulli's method for solving algebraic equations and his own δ^2 method for accelerating the convergence of such iterations, Hartree and Womersley (1937) on mathematical and mechanical (differential analyser) methods for the solution of parabolic partial differential equations, and Richardson and Gaunt (1926) on "the deferred approach to the limit" for accelerating the convergence of finer-net approximations to the numerical solution of ordinary differential equations.

The last mentioned method is still in common use, and Richardson, a major figure in this field, also wrote important papers on the solution of partial differential equations. Perhaps the most famous of these is Richardson (1910), and Richardson (1925) gives a short summary of this and other work. The 1910 paper discussed finite-difference methods for what he called "jury" problems given by

$$\nabla^2\Phi = 0,\ (\nabla^2 + k^2)\Phi = 0,\ \nabla^4\Phi = 0,\ (\nabla^4 - k^4)\Phi = 0 \tag{3.1}$$

with suitable boundary conditions. He postulated a "deferred approach to the limit" rule when central differences are used, not only for the function in all cases but also for the eigenvalues. He solved the finite-difference equations by direct methods if their number was small enough, and otherwise he used an iterative method, now known as Richardson's method.

The following example appears in the 1925 paper. The equations

$$Ax = b, \qquad A = \begin{bmatrix} -4 & 1 & 0 & 1 \\ 1 & -4 & 1 & 0 \\ 0 & 1 & -4 & 1 \\ 1 & 0 & 1 & -4 \end{bmatrix}, \qquad b = \begin{bmatrix} -3 \\ -7 \\ 0 \\ 0 \end{bmatrix} \tag{3.2}$$

obviously relate to a particular member of the first of (3.1), with boundary values on a unit square and with interval $h=\frac{1}{3}$ in both directions. He uses the iteration

$$x^{(r+1)} = x^{(r)} + a_r^{-1}\,(Ax^{(r)} - b), \tag{3.3}$$

showing in this example that if $x^{(1)} = (1,2,0{\cdot}3,0{\cdot}2)^T$, then with $a_1=4$, $a_2=2$, $a_3=6$, the computed $x^{(4)}$ is the exact solution of (3.2). This, of course, follows from the fact that the eigenvalues of A are -4, -4, -2 and -6, but Richardson was aware that the eigenvalues are not usually available. He observed that the largest and smallest can be obtained with rough accuracy, that a single $a_r > \frac{1}{2}|\lambda_{max}|$ in (3.3) will produce ultimate convergence, but that "it saves time to spread out the values of the a_r over the range

covered by the eigenvalues."

This was a remarkable piece of work, on which Golub and Varga and others did more research in the nineteen-fifties with the name "semi-iterative method". The paper has many other interesting sections which suggest other things about finite differences, what to do for example near boundaries which are not rectangular, and the importance of a non-dimensional treatment of the problem prior to computation. He also considered the parabolic problem

$$\frac{\partial \Phi}{\partial t} = \frac{\partial^2 \Phi}{\partial x^2} , \tag{3.4}$$

with appropriate boundary conditions, but his suggested

$$\Phi_{r,s+1} = \Phi_{r,s-1} + 2\left(\frac{\Delta t}{(\Delta x)^2}\right)\left(\Phi_{r+1,s} - 2\Phi_{r,s} + \Phi_{r-1,s}\right) \tag{3.5}$$

is now known to be unstable.

In passing it is interesting to note that it was this pre-war numerical analysis which was mainly examined in postgraduate courses in the subject, courses which did not start seriously until the early nineteen-fifties. They were usually organised by the Computing Laboratory rather than by the Mathematics Department, and one of the very first was the Cambridge Diploma in Numerical Analysis and Automatic Computing. At the start this had one theoretical paper and one practical paper on numerical analysis and one paper on the hardware and software of the new stored-program computers. The two numerical papers of the first examination in 1954 reveal that the material so far discussed is very well represented. This is perhaps not surprising since Miller was the dominant force in this part of the Diploma, but later diplomas had very little more variety. Even by 1959 the theoretical paper at Cambridge had three questions on interpolation, two on quadrature, one on the Taylor-series method for a particular (non-linear) ordinary differential equation, one on Richardson's method for elliptic equations, one on Aitken and other iteration topics, and one on three methods for the eigenvalues of symmetric matrices of small and large order with comparison of desk machines and automatic computation.

4. War-time groups

(i) Relaxation at Oxford

In 1939 I had just started my D.Phil research at Oxford with R.V. Southwell, who had told my tutor that he needed a mathematician to work on extensions of his "relaxation method". In the early thirties he had invented what was originally called the method of "systematic relaxation of constraints" for solving problems of loaded frameworks, and in the decade 1932 - 42 he had a regular group of research students at Oxford working on these problems and somewhat similar problems in the finite-difference solution of elliptic partial differential equations. From 1939 onwards arrangements in the second world war caused applied and even pure mathematicians to work on military problems with whatever techniques they had available, but the Oxford group was one of the first of these and, in particular, the name "relaxation", if not the original method, has carried over to quite modern techniques for relevant problems.

Two papers by Southwell in 1935 described the method for frameworks. Basically this used iteration

to solve the linear equations

$$Ax = b, \tag{4.1}$$

x being the vector of displacements and b of the forces at the joints of the framework. The matrix A was sparse and generally diagonally dominant. Southwell considered in an engineering sense not only the problem but also its method of solution. He postulated a system of "constraints" at the joints which could bear the forces without allowing any displacements. Then, usually selecting the joint with the currently largest force, he permitted a displacement at this point by "relaxing the constraint", wholly or partially at this stage so that at this joint the framework was now bearing all or at least some of its force. This also changed the forces at other joints, in an easily calculable manner, and by systematically "relaxing the constraints" (the word "systematically" originally meaning "in descending order of magnitude of forces still borne by the constraints") he expected on engineering principles that the process would converge. In other words the residual forces still borne by the constraints, components of the residual vector

$$r^{(n)} = Ax^{(n)} - b \tag{4.2}$$

at stage n of the iteration, would systematically be reduced to zero or to very small quantities as n increases. In fact Southwell contemplated the acceptance of any solution for which the residual forces were less than some "engineering fraction" of the original forces, since the latter are quite unlikely to be known very accurately.

Now if at joint s the residual force r_s is reduced temporarily to zero by a change in the displacement x_s at that joint, then this is one step of Gauss-Seidel iteration, and indeed for some problems this method had already been used by other workers. But Southwell concentrated on the residuals, which were actually recorded at every joint, and he and his research students used a variety of methods to reduce them sensibly to zero. The following simple examples illustrate some of these methods.

First we solve a one-dimensional problem with equations

$$f_{r+1} - 2f_r + f_{r-1} = b_r, \quad f_0=100, \quad f_5=-1000, \quad b_1=20, \quad b_2=80, \quad b_3=-40, \quad b_4=600, \tag{4.3}$$

the selected values of b_r and f_0 and f_5 being effectively arbitrary numbers. Suppose that we start with the guess $f_1 = f_2 = f_3 = f_4 = 0$, so that the first relevant picture is that of Figure 1, in which the current f values are to the left and the current residuals to the right of the "nodal lines". The first residuals are just the $-b_r$ at r=2 and 3, and $-b_1$ and $-b_4$ plus the respective contributions from the specified boundary values at the two nodes next to the boundaries.

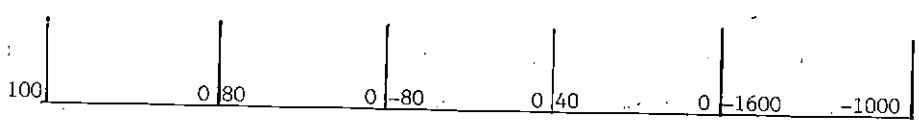


Figure 1

In the relaxation process we use the same diagram throughout, recording additions to the displacements

on the left of the nodal lines and the current residual on the right. In the first step we "liquidate" the residual of largest magnitude, but the form of the equations (4.3) shows that a group displacement of a multiple -320 of displacements 1, 2, 3 and 4 at the successive nodal points will eliminate the -1600 residual without altering any others. The current state is then shown in Figure 2.

		-320		-640		-960		-1280	0	
100		0	80	0	-80	0	40	0	-1600	-1000

Figure 2

Next we eliminate the -80 residual with a single joint relaxation, a displacement of -40 at that joint changing the residual by 80 at that joint and -40 at the adjacent joints on each side, leaving a residual of 40 at the first joint and zero at all other joints. Finally, the multiple 8 of the group displacement 4, 3, 2, 1, the reverse of the first group displacement, produces zero residuals everywhere, the picture of Figure 3, and values of -288, -656, -944 and -1272 at the successive points. A check calculation of the residuals from (4.2) confirms that all the residuals have zero values.

		32	0	24					
		-320	40	-40	0	16	0	8	0	
100		0	80	-640	-80	-960	40	-1280	-1600	-1000
		(-288)		(-656)		(-944)		(-1272)		

Figure 3

In a group displacement several constraints are relaxed simultaneously, and when the displacement changes are the same at the relevant set of joints it is called a block displacement. This, as well as the joint displacement, is very useful in the treatment of differential equations by finite-difference methods. Equation (4.3) might approximate to the solution of a simple ordinary differential problem like

$$\frac{d^2f}{dx^2} = g(x), \quad f(x_0) = \alpha , \quad f(x_n) = \beta , \tag{4.4}$$

with $g(x)$ and α and β specified and the chosen interval taken to be $h = \frac{1}{5}(x_n - x_0)$.

Similarly, for the simple elliptic partial differential equation

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = g(x,y), \tag{4.5}$$

with f having specified values on a closed boundary and with $g(x,y)$ also specified for all x,y within the region, the equation corresponding to (4.3) is

$$f_{r,s+1} + f_{r,s-1} + f_{r+1,s} + f_{r-1,s} - 4f_{r,s} = h^2 g_{r,s}, \tag{4.6}$$

in obvious notation and with constant interval h in both directions. We present a solution of this (there are, of course, many other possibilities) with $f=0$ on the boundary of a unit square, and $h=0{\cdot}2$ and $g=-2500$ so that with an initial guess of $f=0$ everywhere the first picture corresponding to that of Figure 1 is given in Figure 4, with the boundary lines and all zero displacements ommitted for convenience.

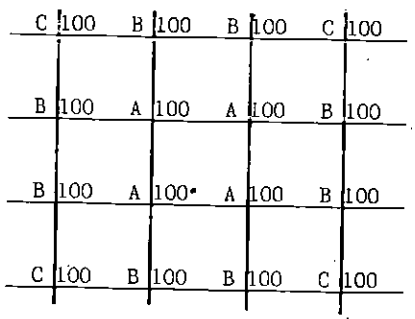


Figure 4

We now use the word "point" instead of "joint" since in Southwell's language the framework had become a tensioned net, and "net point" became the accepted terminology. Other useful words were "overrelax", deliberately to change the sign of the relevant residual(s) when adjacent points have residuals of the same sign and have a "wash-back" effect, and "underrelax" in regions in which the signs of residuals alternate. These words are still used in modern methods but with rather different applications. We also note with respect to (4.6) (and indeed also with respect to (4.3)) that the algebraic sum of residuals is unchanged unless a displacement is made at one or more points next to the boundary, so that residuals should be "swept" from the centre of the region towards the boundaries rather than in the reverse direction. A useful first step is to use a complete block operation which reduces the algebraic sum of residuals virtually to zero.

There is much symmetry in Figure 4, and indeed there are only three independent values, respectively at points marked A, B and C. Table 1 gives a list of operations and the resulting residuals, a displacement at A meaning the same displacement at all A points in Figure 4, and similarly for B and C.

Table 1

Operation	Displacement changes			Current residuals		
	A	B	C	A	B	C
(i)	100	100	100	100	0	-100
(ii)	70			-40	70	-100
(iii)		20		0	10	-60
(iv)			-16	0	-6	4
(v)	-3	-3		0	0	-2

Operation (i) reduces the sum of residuals to zero, and the remaining operations would be understood quite easily by any competent and experienced operator. Notice that simple numbers are used throughout, with no useless early attempts to make any residuals exactly zero. The fact that at the end of Table 1 there are only zero or negative residuals tells us immediately that all the values are too large, that of C perhaps especially. But a complete extra block of -1 would leave residuals of 0, 1 and 0 for A, B and C, so that every value would then be slightly too small. Table 1 gives A=167, B=117, C=84; the exact values being $166\frac{2}{3}$, $116\frac{2}{3}$, $83\frac{1}{3}$. This table, of course, would nowhere be recorded, and all the operations would be performed on a single sheet of paper, with perhaps only one-eighth of Figure 4, by an experienced operator who takes the symmetries in his or her stride. Figure 5 shows all that is needed.

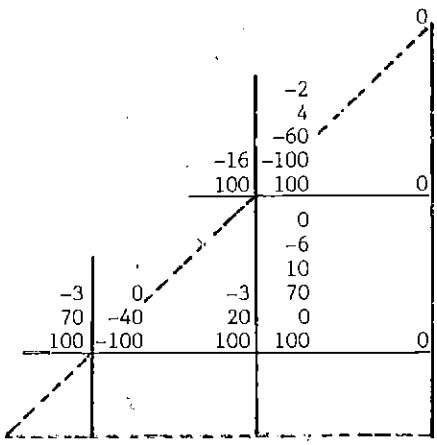


Figure 5

We learnt a lot about the "condition" of various problems, measured by the size of the displacements needed to liquidate sets of residuals. The condition of course worsens as the interval is reduced, but more to the point is the fact that the biharmonic equations are much more ill-conditioned than the Laplace equations. As the condition worsens the need for significant overrelaxation increases. A good starting approximation, of course, helped considerably with the convergence, and in an engineering background some workers could envisage pictorially and really quite accurately the nature of the correct solution. We simplified the use of a finer mesh, first by interpolating quite accurately or, where possible, using the differential equation to get a good start at the finer net points, and then by a process which now has the name "multi-grid". Here any oscillation in residual signs was removed by a few simple point relaxations, and by taking averages over small relative regions of the resulting one-signed residuals these could effectively be transferred back to the original net, and liquidated there with easier relaxation. The results were then transferred back to the finer mesh, and one further interpolation and a relatively trivial amount of fine net relaxation produced the required results quite quickly. Unlike modern multi-grid methods we never used more than one coarser mesh for this purpose.

The latter technique did not obtain written publicity, but most of the useful devices appear in the book by Allen (1954). This, together with Southwell's last books (1946, 1956) also give a full account of the problems solved by relaxation, some non-linear, some involving eigenvalues, some with boundaries of initially unknown position, some in three dimensions and some with parabolic and hyperbolic systems. The eigenvalue techniques were rather interesting. Normally a guess at the eigenfunction gave a starting estimate of the eigenvalue with the use of Rayleigh's principle, and some relaxation was then performed. This cannot proceed too far because the equations do not have a solution at this stage, and a favourite trick was to try to arrange for displacements which made the residual at each point reasonably proportional to its displacement. The computation of a new eigenvalue estimate then gives better results and a good start for further operations. When this was very difficult a method called intensification was used, which turns out to be just the method of inverse iteration for

$$(A - \lambda I)x = 0, \tag{4.6}$$

given by

$$Ax^{(r+1)} = x^{(r)}. \qquad (4.7)$$

Occasionally the operator A − kI might be used in (4.7), not so much to increase the rate of convergence as to simplify the relaxation solution of the linear equations.

One final comment on the relaxation method is essential. The success of the method (and it was successful even with the meagre computing equipment then available), depended significantly on the ability of the human eye and brain very quickly to pick out the largest of a sequence of numbers or a cluster of such numbers, to recognise patterns of numbers and to forecast the overall effects of relaxation operations. In fact it was rather like a game of chess, and I return briefly to this point a little later.

(ii) Admiralty Computing Service

In 1943 I joined the new Admiralty Computing Service at Bath, probably the first group with the words "Computing Service" in its title. It was headed by D.H. Sadler, who was Comrie's successor as Superintendent of the Nautical Almanac Office, and it had as consultants Miller, Erdelyi and John Todd. Its workers also included E.T. Goodwin, F.W.J. Olver and H.H. Robertson, whose names are well known in the literature of numerical analysis. We solved a fair number of problems for the Admiralty, we learnt a lot about the numerical methods of Miller and Sadler, and I extended my knowledge of and capabilities with the relaxation method. Some problems were written up as reports for "Department of Scientific Research and Experiment - Admiralty Computing Service", mainly in 1945, and listed in the references are two of the problems which have particular interest for me.

The first is the evaluation of the two-variable function

$$f(x,y) = \int_0^\infty e^{-k}(J_0(kx)\cosh(ky) - 1)\ \mathrm{cosech}(k)dk \qquad (4.8)$$

at the points x = 0(0·1)5·0, y = 0(0·1)1·0. This is how the problem was presented, but we discovered that f(x,y) satisfies the elliptic equation

$$\frac{\partial^2 f}{\partial x^2} + \frac{1}{x}\frac{\partial f}{\partial x} + \frac{\partial^2 f}{\partial y^2} = 0, \qquad (4.9)$$

and that boundary values can be calculated with some interesting numerical analysis giving quite rapid techniques. I then solved the problem by relaxation methods, and the first point of interest is that this is the first publication of my use of the "difference correction" for correcting a first approximate solution on the same finite-difference mesh. (The year in which these computations were performed was either 1943 or 1944). The second interesting point is that in this very early problem (4.9) was known originally, but a mathematician deduced (4.8) quite cleverly but without knowing that a direct treatment of (4.9) is here computationally preferable. Just how much early mathematics is valuable in numerical work has always been a matter of some speculation and dispute!

The other interesting problem was the solution by Goodwin and myself of a Volterra integral equation of the first kind, with a mixture of Laplace transforms, Taylor's series, and direct numerical solution of a corresponding second-order equation in which the trapezoidal rule had attached to it several correc-

ting expressions from Gregory's quadrature formula. Here Sadler was a source of great strength, with a wealth of finite-difference knowledge of the kind contained in 'Interpolation and Allied Tables'. He published very little himself, but he was always able and willing (if not determined) to make suggestions about methods which were almost always exceedingly useful, and he had a genius for spotting errors in our computation. He insisted that all this should be done on good paper, in ink, and he delighted to peer frequently over our shoulders and triumphantly note an error before we had wasted too much sequential time!

5. Mathematics Division, National Physical Laboratory

The Admiralty Computing Service was quite successful, and its success was one of the main reasons for the setting up in 1945 of the Mathematics Division, a new division of the National Physical Laboratory. Goodwin, Fox, Olver and Robertson went in 1945 from Bath to the NPL at Teddington, J.H. Wilkinson joined in the following year, and roughly at that time we also recruited Clenshaw and Gill and Hayes and a number of others with perhaps less well-known names. The famous Turing came to contemplate building his version of the new idea of stored-program computers, and we had quite a number of junior workers on desk machines and punched-card machines. We also acquired from Germany a large differential analyser. Our duties were to help other divisions of NPL with their 'mathematical and computational' problems, to do the same for other stations of the current version of the Department of Scientific and Industrial Research, and indeed for many other government or government-type laboratories, and above all to engage in research in the theory and practice of numerical computation.

About this time there were other small groups in other government and government-type laboratories, and at several universities, particularly Cambridge and Manchester, who were also working on computer construction and use. Comrie had formed the London Scientific Computing Service, which Miller joined, but as far as the mathematics of numerical analysis was concerned the NPL group was by far the largest and the most experienced. There is no doubt that what you would call the more modern numerical analysis in the UK started with this group which, indeed, was dominant in our numerical work for at least 30 years.

The history of the NPL work has two parts, the smaller for a decade or so until the mid-fifties, in a period in which the new computer was not generally available, and the larger after the appearance of the lusty Pilot ACE computer in a form from which useful computations could be obtained. My history virtually ends with the first of these parts, in which much useful research was still performed. I mention in what follows a few of the topics and resulting publications.

Table-making continued, and indeed NPL started its own series of mathematical tables, a project for which Fox (1956) wrote a lengthy Vol. 1 which extended much of the Chebyshev theory of Lanczos and Miller for interpolation and other relevant formulae. Work on ordinary differential equations produced papers by Fox (1947, 1949), Fox and Goodwin (1949), Gill (1951, in which the effect of the new computer was already foreshadowed), and Clenshaw and Olver (1951). Clenshaw (1954, 1955, 1957) started important work on Chebyshev methods for ordinary differential equations, and Olver, after a comprehensive paper on computing the zeros of polynomials (1952), collaborated with Clenshaw (1955) on the use of economized polynomials in

mathematical tables. Fox and Goodwin (1953) continued their ACS work on integral equations with a comprehensive account of finite-difference methods for both Volterra and Fredholm equations, and Goodwin and Staton (1948) and Goodwin (1949) added to earlier work on methods for evaluating particular integrals. There was some curve-fitting by Hayes and Vickers (1951), and a little linear algebra by Fox, Huskey and Wilkinson (1948), Goodwin (1950), Fox and Hayes (1951), Fox (1950a, 1954), but the main papers for the stimulation of future work in this area came from Turing (1948) and Wilkinson (1954a, b). We did little on partial differential equations except papers on further relaxation by Fox (1947, 1950b), including the difference-correction method. Some independent workers, however, contributed significantly in this field, including of course Crank (1956) and Crank and Nicolson (1947), which produced one of the very useful stable methods for parabolic equations; and Motz (1946) and Woods (1953) did useful work on singularities in elliptic problems. Singularities in some integral equations were also treated by Young (1954).

The NPL group joined together to produce the book 'Modern Computing Methods' (1957, second edition 1961), which includes an extensive bibliography. This was one of the first quite modern books on numerical analysis, somewhat more up-to-date at that time than the very readable 'Numerical Analysis' by Hartree (1952). My book (1957) on "The numerical solution of two-point boundary problems in ordinary differential equations" put into print the work started some fifteen years earlier on relaxation methods and the "difference-correction" method. This, again, must be one of the earliest books on this topic.

And that is really the end of the "Early Numerical Analysis" story. In the middle nineteen-fifties and onwards there was a flood of books and papers on numerical analysis of all kinds and from many places, largely stimulated by the development of the stored-program computer. The NPL contribution to this feast was supplied very largely by J.H. Wilkinson. His third relevant paper (1955) was merely the first of a series which for the next thirty years transformed both the theory and the practice of virtually all problems in numerical linear algebra.

But that is another story. I end the current story by making a few comments on the effect ofthe new computing machine on our earlier work. First, in 1958 at a meeting of the Royal Society Mathematical Tables Committee, the chairman M.V. Wilkes raised the question of its role in the new computer world. This led to considerable and lengthy debate, but the extent of table-making decreased quite rapidly and the committee virtually ceased to exist around 1965. Second, the old relaxation methods were never used in the same spirit with the new computers. For the latter did not match the human eye and brain in picking out relatively quickly the largest of a sequence of numbers, or recognise useful patterns, and the new relaxation method developed by David Young and others worked in a virtually completely systematic way. This, of course, led to some useful and very interesting mathematical theories, but the modern method bears only slight relation to the original relaxation concept.

My "difference-correction" method for differential and integral equations of all kinds was also treated afresh by V. Pereyra and others, and they also made some changes, though perhaps not quite so violent as those of the relaxation story. For example, for the two-point boundary problem

$$y'' + f(x)y' + g(x)y = k(x), \quad y(a) = \alpha, \quad y(b) = \beta, \tag{5.1}$$

I replaced the differential equation by the recurrence relation

$$(1-\tfrac{1}{2}hf_r)y_{r-1} - (2-h^2g_r)y_r + (1+\tfrac{1}{2}hf_r)y_{r+1} = c(y_r), \; y(a) = \alpha, \quad y(b) = \beta, \tag{5.2}$$

where $c(y_r)$ is the difference-correction at mesh point x_r which I expressed in terms of central differences, here involving third, fourth and higher-order differences. I proposed to solve (5.2) iteratively in the form

$$(1-\tfrac{1}{2}hf_r)y_{r-1}^{(n+1)} - (2-h^2g_r)y_r^{(n+1)} + (1+\tfrac{1}{2}hf_r)y_{r+1}^{(n+1)} = c(y_r^{(n)}), \quad c(y_r^{(o)}) = 0, \tag{5.3}$$

a device very similar to the modern use of "iterative refinement" for simultaneous linear algebraic equations. I inspected the differences of $y_r^{(1)}$ to discover what orders of differences at this interval made contributions to $c(y_r^{(1)})$ for the required accuracy, whether from this point of view the interval length was satisfactory and, really quite accurately, how the interval should be changed for this purpose. All further calculations were performed at this "satisfactory" interval, starting with a new $y_r^{(1)}$ and continuing with the iterative sequence. Using only the differences at every stage which were expected to contribute to $c(y_r)$ I performed the iteration as many times as needed to reach consistency in the computed results. Some external values had to be computed and even "corrected" to produce the central differences near boundary points.

Pereyra, however, showed that whereas $y_r^{(1)}$ has global error $O(h^2)$, $y_r^{(2)}$ has global error $O(h^4)$ if $c(y_r^{(1)})$ uses only third and fourth differences, and $y_r^{(3)}$ has global error $O(h^6)$ if $c(y_r^{(2)})$ is computed using only up to sixth differences. Normally the number of differences to be used finally would be decided before the computation started, and if consistency had not been reached at this stage the process would be repeated at a smaller interval. I am not clear what the present position is, but in the early routines external values were not computed and forward or backward differences were used for at least some y_r in $c(y_r)$. Again, this new theory is very important, but the method has undoubtedly changed, at least to some extent including the fact that, as with initial value problems, (5.1) is now likely to be treated as simultaneous first-order equations with the trapezoidal rule.

Finally, the new computers were so powerful that they quickly put an effective end to the use of analogue equipment like the differential analyser for the solution of partial differential equations, and other various pieces of equipment for various problems in which the data and answers were measured by physical quantities like length, voltage, current and so on. Another analogue device was the construction of alignment nomograms which up to this time had been a regular feature of problem solving of certain kinds and had developed quite a literature.

The basic idea of the alignment nomogram can be demonstrated by a very simple example, the solution of the quadratic equation

$$a^2 + pa + q = 0, \tag{5.4}$$

which has three variables a, p and q. The nomogram depends on our ability to express (5.4) in the deter-

minantal form

$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = 0, \tag{5.5}$$

which represents the condition that in two dimensions the points (x_1, y_1), (x_2, y_2) and (x_3, y_3) are collinear. For (5.4) there are various such possibilities, of which one is given by

$$\begin{vmatrix} p^{-1} & 0 & 1 \\ 0 & q^{-1} & 1 \\ -a^{-1} & -a^{-2} & 1 \end{vmatrix} = 0. \tag{5.6}$$

The corresponding nomogram has the scale $x = p^{-1}$ on y=0, $y = q^{-1}$ on x=0, and an a-scale on the parabolic curve $x = -a^{-1}$, $y = -a^{-2}$. A line joining a "p-point" to a "q-point" intersects the "a-curve" at two, one or no points, these points giving the real roots of the quadratic equation.

We constructed a fair number of nomograms in the early nineteen-fifties, one of them I discovered from forgotten notes relating to the equation

$$\left(\frac{R^2}{AQ^2}\right)^{\frac{1}{3}} = K_T^{\frac{1}{3}} e^{-1\cdot 3R/V} - K_o^{\frac{1}{3}}, \tag{5.7}$$

where we wanted R from specified A, K_T, V, and Q and K_o which were given functions of a fifth parameter β_s. With five parameters s, t, u, v, w we need to be able to produce a determinant like

$$\begin{vmatrix} f(s,t) & g(s,t) & 1 \\ p(u,v) & q(u,v) & 1 \\ \lambda(s) & \mu(s) & 1 \end{vmatrix} = 0, \tag{5.8}$$

to provide an alignment nomogram. I failed to do this for (5.7), but introduced two other parameters

$$\alpha = A^{\frac{1}{3}} Q^{\frac{2}{3}} K_o^{\frac{1}{3}} = A^{\frac{1}{3}} f(\beta_s), \quad \beta = K_o^{\frac{1}{3}} K_T^{-\frac{1}{3}}, \tag{5.9}$$

leading to the determinants

$$\begin{vmatrix} \alpha & 0 & 1 \\ -A^{\frac{1}{3}} & 1 & 1 \\ 0 & f(1+f)^{-1} & 1 \end{vmatrix} = 0, \quad \begin{vmatrix} 0 & \beta & 1 \\ 1 & -K_o^{\frac{1}{3}} & 1 \\ (1+K_T)^{-\frac{1}{3}} & 0 & 1 \end{vmatrix} = 0, \tag{5.10}$$

and the ability to find α and β from given A, K_T and β_s. Equation (5.7) can then be expressed in the form

$$R^{\frac{2}{3}} \alpha^{-1} = e^{-1\cdot 3R/V} \beta^{-1} - 1, \tag{5.11}$$

with the relevant determinantal equation

$$\begin{vmatrix} \alpha & 0 & 1 \\ 0 & \beta & 1 \\ -R^{\frac{2}{3}} & e^{-1\cdot 3R/V} & 1 \end{vmatrix} = 0 \qquad (5.12)$$

which permits the determination of R from given α, β, and V.

The accuracy obtainable depends upon various things including the determination of appropriate scaling factors for the variables, and the literature gave this some close attention. At NPL my colleague J.G.L. Michel was our "analogue" expert, both with the differential analyser and with nomography, and he joined the other authors in producing the fourth edition of a very good book on the subject by Allock et al (1950).

Since that time I have heard no more about nomography, but of course it is quite proper that old methods should be reviewed, readapted and if necessary discarded when new equipment becomes available, and this is one of the important ways in which numerical analysis continues to make good progress in its initial task of helping the scientists in their work.

References

Admiralty Computing Service (S.R.E. Dept.) SRE/ACS 47, 1945, Tabulation of the function

$$f(x,y) = \int_0^\infty e^{-k}\{J_0(kx)\cosh(ky) - 1\}\ \mathrm{cosech}(k)\ dk.$$

---- SRE/ACS 89, 1945. Solution of integral equations occurring in an aerodynamical problem.

Airey, J.R. 1937. The converging factor in asymptotic series and the calculation of Bessel, Laguerre and other functions. Phil. Mag. 24, 521 - 552.

Aitken, A.C. 1926. On Bernoulli's numerical solution of algebraic equations. Proc. Roy. Soc. Edinb. 46, 289 - 305.

----1937. Studies in practical mathematics II. The evaluation of the latent roots and latent vectors of a matrix. Proc. Roy. Soc. Edinb. 57, 269 - 304.

Allcock, H.J., Jones, J.R. and Michel J.G.L. 1950. The nomogram (first edition 1932). London: Pitman.

Allen, D.N.de G. 1954. Relaxation methods . McGraw-Hill. New York.

Barlow's Tables 1814 (ed. Peter Barlow). (Editions, 1930, 1941 ed. L.J. Comrie) Spon: London.

Bickley, W.G. 1939. Formulae for numerical integration. Math. Gaz. 23, 352 - 359.

---- 1941. Formulae for numerical differentiation. Math. Gaz. 25, 19 - 26.

---- and Miller, J.C.P. 1936. The numerical summation of slowly convergent series of positive terms. Phil. Mag. 22, 754 - 767.

British Association Mathematical Tables, 1952 Vol. X. Bessel Functions, Part II. Cambridge University Press.

Brunt, D. 1923. The combination of observations. Cambridge University Press.

Clenshaw, C.W. 1954. Polynomial approximations to elementary functions. Math. Tab. Wash. 8, 143 - 147.

---- 1955. A note on the summation of Chebyshev series. Math. Tab. Wash. 9, 118 - 120.

---- 1957. The numerical solution of linear differential equations in Chebyshev series. Proc. Camb. Phil. Soc. 53, 134 - 149.

Clenshaw, C.W. and Olver, F.W.J. 1951. Solution of differential equations by recurrence relations. Math. Tab. Wash. 5, 34 - 39.

-------- 1955. The use of economized polynomials in mathematical tables. Proc. Camb. Phil. Soc. 51, 614 - 628.

Comrie, L.J. 1931. British Association Mathematical Tables Vol. I. Cambridge University Press.

---- 1936. Inverse interpolation and scientific applications of the National Accounting Machine. J. R. Statist. Soc. Supplement 3, 87 - 114.

Crank, J. 1956. The mathematics of diffusion. Oxford University Press.

---- and Nicolson, P. 1947. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. Proc. Camb. Phil. Soc. 43, 50 - 67.

Fletcher, A., Miller, J.C.P. and Rosenhead, L. 1946 (later edition 1962). An index of mathematical tables. London: Scientific Computing Service.

Fox, L. 1947. Some improvements in the use of relaxation methods for the solution of ordinary and partial differential equations. Proc. Roy. Soc. A 190, 31 - 59.

---- 1949. The solution by relaxation methods of ordinary differential equations. Proc. Camb. Phil. Soc. 45, 50 - 68.

---- 1950a. Practical methods for the solution of linear equations and the inversion of matrices. J.R. Statist. Soc. B, 12, 120 - 136.

---- 1950b. The numerical solution of elliptic differential equations when the boundary conditions involve a derivative. Phil. Trans. A, 242, 345 - 378.

---- 1954. Practical methods for the solution of linear equations and the inversion of matrices. Appl. Math. Ser. U.S. Bur. Stand. 39, 1 - 54. Washington: Government Printing Office.

---- 1956. The use and construction of mathematical tables. NPL series Vol.I. London: H.M. Stationery Office

---- 1957. The numerical solution of two-point boundary problems in ordinary differential equations. Oxford University Press.

---- and Goodwin, E.T. 1949. Some new methods for the numerical integration of ordinary differential equations. Proc. Camb. Phil. Soc. 45, 373 - 388.

-------- 1953. The numerical solution of non-singular linear integral equations. Phil. Trans. A, 245, 501 - 534.

---- and Hayes, J.G. 1951. More practical methods for the inversion of matrices. J.R. Statist. Soc. B, 13, 83 - 91.

----, Huskey, H.D. and Wilkinson, J.H. 1948. Notes on the solution of algebraic linear simultaneous equations. Q.J. Mech. Appl. Math. 1, 149 - 173.

Gill, S. 1951. A process for the step-by-step integration of differential equations in an automatic digital computing machine. Proc. Camb. Phil. Soc. 47, 96 - 108.

Goodwin, E.T. 1949. The evaluation of integrals of the form $\int_{-\infty}^{\infty} f(x)e^{-x^2}dx$. Proc. Camb. Phil. Soc. 45, 241 - 245.

---- 1950. Note on the evaluation of complex determinants. Proc. Camb. Phil. Soc. 46, 450 - 452.

---- and Staton, J. 1948. Table of $\int_0^{\infty} (u+x)^{-1} e^{-u^2} du$. Q.J. Mech. 1, 319 - 326.

Hartree, D.R. 1952. Numerical analysis (later edition 1957). Oxford University Press.

---- and Womersley, J.R. 1937. A method for the numerical or mechanical solution of certain types of partial differential equations. Proc. Roy. Soc. A, 161, 353 - 366.

Hayes, J.G. and Vickers, T. 1951. The fitting of polynomials to unequally spaced data. Phil. Mag. 42, 1387 - 1400.

Levy, H. and Baggott, E.A. 1934. Numerical studies in differential equations. London: Watts.

Miller, J.C.P. 1949. The construction of mathematical tables. Sci. J.R. Coll. Sci. 20, 1 -11.

Milne-Thomson, L. 1933. The calculus of finite differences (later edition 1951). London: Macmillan.

Motz, H. 1946. The treatment of singularities of partial differential equations by relaxation methods. Q.J. Appl. Math. 4, 371 - 377.

N.P.L. 1957. Modern Computing Methods (later edition 1961). London: H.M. Stationery Office.

Nautical Almanac Office. 1936. Interpolation and allied tables (last edition 1956). London: H.M. Stationery Office.

---- 1958. Subtabulation. London: H.M. Stationery Office.

Olver, F.W.J. 1952. The evaluation of zeros of high-degree polynomials. Phil. Trans. A, 244, 385 - 415.

Richardson, L.F. 1910. The approximate arithmetical solution by finite differences of physical problems involving differential equations with an application to the stresses in a masonry dam. Phil. Trans. A 210, 307 - 357.

---- 1925. How to solve differential equations approximately by arithmetic. Math. Gaz. July 1925,

---- and Gaunt, J.A. 1926. The deferred approach to the limit. Phil. Trans, A, 226, 299 - 361.

Scarborough, J.B. 1930. Numerical mathematical analysis (Later edition 1950). Oxford University Press.

Southwell, R.V. 1946. Relaxation methods in Theoretical Physics. Oxford: Clarendon Press.

---- 1956. Relaxation methods in Theoretical Physics, Vol. II. Oxford: Clarendon Press.

Sheppard, N.F. 1906. On the accuracy of interpolation by finite differences. Proc. London Math. Soc. 4, 320 - 341.

Steffenson, J.F. 1927. Interpolation (later edition 1950). New York: Chelsea.

Turing, A.M. 1948. Rounding-off errors in matrix processes. Q.J. Mech. 1, 287 - 308.

Whittaker, E.T. and Robinson, G. 1924. The calculus of observations (later edition 1944). London: Blackie.

Wilkinson, J.H. 1954a. Linear algebra on the Pilot ACE. Proc. Symp. Autom. Dig. Comput. NPL, 129 - 136. H.M. Stationery Office.

Wilkinson, J.H. 1954b. The calculation of the latent roots and vectors of matrices on the Pilot model of the ACE. Proc. Camb. Phil. Soc. 50, 536 - 566.

---- 1955. The uses of iterative methods for finding the latent roots and vectors of matrices. Math. Tab. Wash. 9, 184 - 191.

Woods, L.C. 1953. The relaxation treatment of singular points in Poisson's equation. Q.J. Mech. 6, 163 - 185.

Young, A. 1954. The application of product-integration to the numerical solution of integral equations. Proc. Roy. Soc. A, 224, 561 - 573.

Reactor Computations; Surface Representation; Fluid Dynamics

Garrett Birkhoff
Harvard University

My talk will concentrate on developments in scientific computing with which I was personally involved during the years 1945-70. It will deal mainly with advances in nuclear reactor modeling, in computerizing the representation of smooth surfaces, and in numerical fluid dynamics. It will try to bring out the dependence of these advances on classical analysis (including differential equations and numerical analysis), as well as on Mathematical Physics and empirical data from Engineering Science. To explain this dependence, I will begin by recalling some mathematical activities relating to World War II.

A Personal Retrospection of Reservoir Simulation

D. W. Peaceman
Consultant

In 1951 reservoir modeling was done with physical models, such as sand packs and electrical networks. With the primative computing equipment then available, we were able to solve a nonlinear 1-D gas flow problem and to test our new A.D.I. method. How we came to discover A.D.I. will be recounted.

With one phase and two dimensions in hand, we moved on to the modeling of two-phase imiscible displacement in two dimensions. The finite difference methods we used then still form the basis for reservoir simulators used by the industry today. I will discuss improvement that have made simulators more robust.

I will review the parade of computers we used at Humble and Exxon Production Research, from the IBM C.P.C. to the Cray 1-S. The interaction between machine capabilities, our understanding of hardware, and the kinds of problems we could solve at any given time will be stressed.

The Origins of the Mathematics of Computation

Eugene Isaacson
Courant Institute

My talk will emphasize the origins and history of the Mathematics of Computations, with some personal comments.

The prehistory and early history of Computation at the U. S. National Bureau of Standards

John Todd

Department of Mathematics 253-37
California Institute of Technology
Pasadena, California 91125

Introduction

Our ancestors were the surveyors, navigators, astronomers and table makers and electronic engineers. Speaking at Princeton it would be remiss if I did not mention Otto Neugebauer and Hermann Goldstine who have covered, respectively, the earliest computations and those in 16/19 centuries.

In a conference on the history of computing, the contributions of R. C. Archibald [1875-1955] must be recognized, as the founder in 1943 of MTAC (since 1960 Mathematics of Computation) and its editor until 1949. For accounts of his work see the notes by A. A. Bennett 4(1950), p. 1-2 and by D. H. Lehmer 10(1956), 112. In particular we note his book Mathematical Table Makers, Scripta Mathematica, New York 1948, 82 pp.

[1871-1929].

For my talk I would like to begin in 1871--this is the year in which the BAAS Math. Tables Committee began operations, which lasted at varying intensity, until 1948 when the Royal Society took over. In "Final Report of Committee on Calculation of Mathematical Tables, Advancement of Science, 7(1948), 342-347," it will be seen that some of the greatest British mathematicians, both pure and applied, were active in this project.

The development of university studies in our area, in addition to government institutions and commercial enterprises, required our attention.

In Germany the work of Carl Runge has been described in two books:

Iris Runge, Carl Runge [1856-1927] und sein wissenschaftliches Werk, Abh. Akad. Wiss. Göttingen, Math. Phy. Klasse (3) #23, 214 pages, 1949.

G. Richenhagen, Carl Runge, Von der reinem Mathematik zur Numerik. Van den Hoeck & Ruprecht Göttingen, 1985, 355 pages.

When E. T. Whittaker [1873-1956] came to the University of Edinburgh in 1912 from the Royal Observatory in Ireland, he founded a Mathematical Laboratory. For an account of Whittaker's views about the importance such organizations; see the obituaries A. Erdélyi, E. T. Whittaker, MTAC11 (1957), 53-4 and W. H. McCrea, E. T. Whittaker, JLMS32 (1957), 234-256. Whittaker maintained an interest in numerical mathematics throughout his life and in 1938 presided over a BAAS Symposium on "From function to printed table: Some aspects of the work of preparing a table of a mathematical function." His successor at Edinburgh, A. C. Aitken [1895-1967], made numerous contributions to numerical mathematics as did one of his pupils E. L. Ince [1891-1941]. Ince's work on Mathieu functions, brought them into the range of "standard" tabulated functions (see later work by G. Blanch and W. G. Bickley). Ince also contributed a volume of tables in algebraic number theory to the British Association Series--a book which at least one mathematician I know "never leaves home without."

A notable event took place in Naples in 1927 when Mauro Picone [1885-1976] established an Instituto Nazionale per le Applicazioni del Calcolo (INAC). He first characterized INAC as a "living table of functions" but later, in 1952, wrote "it is the place where the marriage between functional topology and numerical calculation has taken place." INAC was transferred to Rome in 1932.

In 1928 Alwyn Walther [1898-1967] began the development of IPM (Institut für praktische Mathematik) at Darmstadt. For an account of this we refer to

W. de Beauclair, Alwyn Walther, IPM and the development of calculator/computer technology in Germany 1930-45, Ann. Hist. of Computing 8(1986), 334-350.

It would be remiss of me also not to call attention to an overlooked paper from this university, published in the Annals. Jesse Douglas, A method of numerical solution of the problem of Plateau, Ann. of Math. (2) 29(1928), 180-188. This paper contains much good sense and deserves restudy since it relates to an important problem to which there has been new contributions recently from Meeks and Hoffman.

The Thirties

This was an important, if transitional decade in the history of our subject. All I can do is to mention some of the publications which appear to be influential for later developments. First of A. M. Turing, On computable numbers with an application to the Entscheidungsproblem, Proc. London Math. Soc (2) 42(1936/7), 230-265, A. M. Ostrowski, Über die Konvergenz und die Abründungsfestigkeit des Newtonschen Verfahren, Rec. Math. 2(1937), 1073-1098.

[Note that later Ostrowski wrote a pioneering paper in symbolic integration: Sur l'intégrabilité élémentaire

de quelques classes d'expressions, Comm. Math. Helv. 18(1946), 283-308.]

In view of the importance of graphics it is worthwhile mentioning the books of tables of Jahnke-Emde and the work of Carl Störmer which began early in the century and was surveyed in his paper: Comptes Rendus du Congre's International des Mathematiciens, Oslo 1936. Tome 1, pp. 61-75. "Programme for the quantitative discussion of electron orbits in the field of a magnetic dipole, with application to cosmic rays and kindred phenomena."

In 1936, L. J. Comrie [1893-1950] left his position as Superintendent of the (British) Nautical Almanac Office to found Scientific Computing Service in London. For an account of his work see H.S.W. Massey, L. J. Comrie, Obituary Notices of Fellows of the Royal Society 8(1952), 97-107. One of Comrie's contributions to our subject was the ingenious use (or misuse) of commercial equipment for scientific calculations, notably the NCR accounting machine. (This was the machine for which von Neumann probably wrote his first program for a problem suggested by D. H. Sadler, who succeeded Comrie as Supt. Nautical Almanac Office. See John Todd, John von Neumann and the National Accounting Machine, SIAM Review 16(1974), 526-530.) Comrie also showed how to efficiently use as a parallel machine, the (hand operated) Twin Brunsviga 13Z, e.g. to calculate r cos θ, r sin θ simultaneously. (When this machine became unavailable in WWII he showed how to couple two (electric) Marchant machines, available from USA, for use in artillery calculations.)

We mention here the work of A. J. Thompson [1885-19xx] although it covered four decades, beginning in the 1920's and culminating in the books: A. J. Thompson, Logarithmetica Britannica 2 vols, Cambridge University Press, 1952, which give logarithms to 20D of the numbers from 10^3 to 10^5. Not only did he compute these numbers on a machine built by himself, coupling four (Triumphator?) hand machines, but he set it all himself on a Monotype keyboard. All this was done in his spare time: he was a civil servant in the British General Register (Census) Office. Thompson acknowledges his debt to the Triumphator agent who, when he went out of business, passed on his entire stock of spare parts.

Another important event was the establishment in 1938 of the WPA Mathematical Tables Project in New York under the scientific control of the U.S. National Bureau of Standards with A. N. Lowan as its Director. See A. N. Lowan, The Computation Laboratory of the National Bureau of Standards, Scripta Math. 15(1949), 33-63.

WWII

During this period there was, of course, an intensification of the efforts in our subject. More mathematicians were drafted into it and some remained with it.

My own experience is briefly: After working in C. P. Snow's census of Scientists and Engineers in 1940 I saw to it that I was assigned as a Degaussing Range Officer. However this was the time of acoustic mines and my assignment was changed first to work in the Mine Design Dept. near Portsmouth. There I had my first encounter with NBS; we used the WPA tables of e^x to design delay circuits which ensured that our magnetic mines, although triggered ahead of the bow of a target would detonate under the engine room. We also had to make tables of the dip and drift of our contact mines. After observing this for some months I convinced my superiors that I could organize more effective use of mathematicians and mathematics and was transferred to the Admiralty Dept. of Scientific Research and Experiment where I had considerable freedom and support and education from my colleagues, especially Erdélyi and Sadler. The Admiralty Computing Service was organized. Accounts of its work are: D. H. Sadler and John Todd, Admiralty Computing Service, MTAC 2(1947), 289-297; Mathematics in government service and industry, Nature 157(1946, May), 571-573; A. Erdélyi and John Todd, Advanced instruction in practical mathematics, Nature 158 (1966, 16 November), 690-692.

The war time activities of mathematicians in U.S.A. has been well documented by Mina Rees and by J. Barkley Rosser; in Germany there have been comprehensive reports by W. Süss (pure mathematics 2 vols.) and A. Walther (applied mathematics, 5 volumes).

One of the successes of SCS was the location from incomplete and imperfect data of an enemy transmitter in France which was guiding bombers to targets in 1940/41. This was acknowledged in the official History of British Intelligence. Comrie reports that a Spitfire found and photographed the camouflaged transmitter within 100m. of his estimate.

Just after the end of WWII SCS published its "Index of Mathematical Tables." An enlarged second edition appeared later, A. Fletcher, J.C.P. Miller, L. Rosenhead and L. J. Comrie "An Index of

Mathematical Tables", 2 vols. 1962, Addison-Wesley. These volumes remain a monument to Comrie and S.C.S. whose staff included, among others, H. O. Hartley and J.C.P. Miller. It is also appropriate to mention here the complementary J. A. Greenwood and H. O. Hartley Guide to tables in mathematical statistics, Princeton, 1962.

I have a personal reminiscence of Comrie to tell. There was no love lost between him and the Admiralty, which I inherited, although a very temporary scientific officer, during the ACS times. He scolded me for misspelling Britannica, for instance. However things improved and we arranged to have a friendly lunch in Soho, midway between our offices. We had just ordered, in a backroom, under a skylight protected by linen glued on, when a V1 landed close by and we were enshrouded by the dirty linen. Dusty, but undamaged, we gave up lunch and returned to his office where we waited for the return of his computers, who were also lunching within range, to return, fortunately unhurt, to be given some brandy and sent home.

Post WWII

In the early postwar years national centers for applied mathematics, in particular, computing, were organized in many countries. We discuss briefly what happened in the U.S.A.

The U.S. N.B.S. was constituted in 1901. The first formal recognition of mathematics in its program occurred in 1947 with the formation of AMD=NAML. The activity in numerical mathematics continues to the present and reached a relative maximum in the early 50's. The leaders were J. H. Curtiss [1947-1953], E. W. Cannon [1954-1972], B. W. Colvin [1972-1986], F. L. Alt [1953-1954], F. E. Sullivan [1986-].

Professor Hestenes and I, at the invitation of the MAA, have completed in 1985 an extensive history (1947-1954) of the NBSINA, being published as an NBS Special Publication. I have written an article on Numerical Analysis at the NBS, SIAM Review 17(1975), 361-370, and an Obituary of J. H. Curtiss: Annals History of Computing 2(1980), 106-110. (Copies of the last two and of the paper on von Neumann and the NCR Accounting Machine are still available.) I will, therefore, confine my attention to some general remarks and to some matters which were not sufficiently emphasized in these papers.

The success of the NBS had several reasons. First, the fact that the Director, E. U. Condon, was a distinguished mathematical physicist as was his opposite number C. G. Darwin, Director of NPL. Second, the Chief of the NAML, J. H. Curtiss was ideally suited for the job and well trained in mathematics and statistics and a part of the U.S. mathematical establishment as was his father D. R. Curtiss and his brother-in-law A. W. Tucker. A third reason was the (usually) liberal policy of the U.S. Civil Service Commission about hiring non-citizens, e.g., Agmon, Fortet, Hartree, Kato, Ostrowski, Stiefel, Wielandt, Olga Taussky-Todd and myself, all of whom had previous experience in numerical mathematics. Also the NBS provided many opportunities for career development at various levels and that at the postdoctoral level was particularly successful (e.g., J. R. Rice, Marvin Marcus). Finally let me mention the experienced members of the WPA Group (Abramowitz, Blanch, Lanczos, Rhodes, Salzer, Stegun, Zucker, e.g.) and the (sometimes) generous support from other agencies of the Federal Government.

The period 1947 to 1950 was a transitional period from hand and punched card equipment to the new automatic electronic computers SEAC and SWAC. These machines had to be designed, built and then we had to learn to use them. We had to develop a philosophy for the solution of massive problems. It was, roughly, that of controlled computational experiments, i.e., to compare the theoretical results of academic problems with the experimental results obtained by the use of appropriate algorithms on the academic problems.

After the completion of the machines there was little time for experimentation. The AF, who funded SEAC, naturally wanted it for their Linear Programming Problems and the engineers who build it wanted continually to improve it. Things got worse when SEAC was commandeered by the AEC.

It was surprising that significant work was accomplished. We recall that the building of SEAC was funded by the USAF when there were delays in the construction of their UNIVAC. Alan J. Hoffman was responsible for the related activities in the CL. Apart from the solution of specific problems (some even going back to the WPA Group) a comparative study of algorithms for LP was made, which is still cited as a model of how computational experiments should be made and reported. Hoffman himself wrote a seminal report on the approximate solution of

linear inequalities and pointed out the connections with combinatorial problems, an area which was greatly advanced by J. Edmonds and then developed into a speciality of its own.

During this time steady progress was made in fulfillment of a dream of the WPA group: to publish a 'neuerer'Jahnke-Emde'. With encouragement from, among others, P. M. Morse and financial support from the NSF, AMS55 Handbook of Functions ed. M. Abramowitz and I. A. Stegun was published in 1964, unfortunately after the death of Abramowitz [1914-1958]. It was a resounding success.

During the MacCarthy era the NBS and the mathematics group did not escape severe losses, e.g., E. U. Condon resigned in 1950. Condon's successor A. V. Astin was fired in 1953 because of the ADX2 Battery Additive Affair. After public outcry he was reinstated but, despite a favorable report by the Kelly Committee, the size of the NBS operation was greatly reduced and, in particular, the INA operation at UCLA was terminated in 1954 and there was a substantial reduction in force in the Computation Laboratory: about a third of its staff, then numbering about 100. Fortunately there was quite a demand from industry and universities for people with actual computer experience.

A Numerical Analysis section (with a somewhat less ambitious program than INA) of which I was Chief was separated off from the Computation Laboratory. Abramowitz became Chief of the CL until his death in 1958.

Conclusion

In 1957 I received an invitation from Caltech to help build up their numerical analysis program and I accepted it because of my interest in teaching and the impending move of NBS to suburban Maryland. I was replaced by P. J. Davis and he, later, by Morris Newman.

This appears to be a good place for me to stop since I no longer was in direct contact with NBS and there were many organizational and personel changes.

Programmed Computing at the Universities of Cambridge and Illinois in the early fifties.

David J. Wheeler FRS
Computer Laboratory
Cambridge University
England

Abstract

The development of methods of using computers for calculations in the early fifties at Cambridge and Illinois Universities. They are the recollections of a participant.

Subject Descriptors. History of Computing, Programming methods.

Introduction

The fifties were a time of transition from calculating using hand calculating machines to computing with the aid of digital computers. At that time the computers were slow and unreliable, but relatively fast compared with hand calculations. About one in a hundred calculations go wrong when computing by hand, but about one in a million when computing by automatic computers in the early fifties. Thus the type of checking changed. The programs were written by hand and errors occurred again at the rate of a few errors per hundred instructions written down. However, the programs were corrected and faults removed from programs did not recur, although everyone seems to know of exceptions! The effects of most errors were more obvious than those occuring in hand calculation. Program looping, stopping early, or producing totally spurious results were easy to detect. Removing the errors thus located, left the more subtle ones to be found.

Early calculations on the EDSAC.

EDSAC was designed by M.V.Wilkes and W.Renwick after Wilkes had attended the lectures given at the Moore School of Electrical Engineering in July and August 1946.

The computer ran its first calculation almost forty years ago in May 1949 at the Mathematical Laboratory, Cambridge. It was a table of squares, printed in a reasonable layout. The major part of the program was for binary to decimal conversion and laying out the results. The calculation was totally automatic requiring no human intervention apart from pressing the start button after loading the program tape in the paper tape reader. It was the first calculation done fully under program control in a programmable computer.

The main effort over the next few months was to make the computer more reliable and also to make it easy to use. After all, we were all non professionals.

The design of the EDSAC was very convenient for the user. A start button activated a uniselector (stepping switch) which forced a prewired program into the store and started obeying it. This starting program, known as the initial orders, then input the program from the paper tape in the paper tape reader to the store and started the program. The first version of the starting program was replaced in August 1949 to make it more versatile and able to cope with relocation of subroutines and their parameterisation during input. Thus they could be adapted to a calculation without wasting time or space during the running of the program.

The original design of the EDSAC was to hold numbers less than two in the two's complement representation. When constructed it turned out that all numbers were less than one, we rapidly convinced ourselves that this was better for calculations! Corrections made by changing the specifications are not unknown even today.

The rounding of results was done by an explicit order. The original aim was to force programmers to consider where and how to round. However, as rounding typically slowed a computation loop by about ten per cent, this caused some effort to go into avoiding the rounding operation. It was nominally needed after each multiplication and shift order.

The very early programs were monolithic and written without the aid of subroutines as the library of subroutines was not yet written or organised. A few of these early programs were kept and used as demonstrations. I can remember a program which computed primes by means of subtraction and tests alone. It was quite a short demonstration program and had the property of visibly slowing as the potential primes became larger. I am not sure now why we thought this was an interesting property.

We then settled down to make the computer into a useful calculator. The first subroutines to go into the paper tape library were the input and output subroutines. Almost every program needed these. The methods chosen to implement functions and procedures were adapted from existing methods but with a different emphasis on speed and power. One of the defects of the EDSAC was that it had no division order. This distorted the available methods in an awkward way, favouring methods not needing division. Even if the program needed division, care was needed to select the appropriate division subroutine. To use such subroutines always took more orders than would be needed if the division had been included in the order code. Division was first implimented using the standard second order iterative process as the basis for a subroutine. This was soon supplimented by one based on a repetitive process which was shorter and faster although slightly less accurate.

I believe the most significant library subroutine was the modification of the Runge-Kutta method of solving differential equations due to Stan Gill. This was a remarkable piece of programming design, and appeared at just the right time. It was a small subroutine of sixty six orders and handled the complete solution. It minimised the use of working space, taking only 3 storage locations per differential equation while effectively accumulating the step increments to extra precision. It used four derivative evaluations per step and was of the fourth order. As the computer had only thirty five bits in a word and scaling considerations meant the available accuracy was much less, full accuracy could be attained in about one hundred steps or so. Higher order methods would not be very much faster if they had more complicated steps needing more evaluations.

There was no automatic adjustment of step length. However, it was easy to check the precision by repeating the calculation with a step of half the length. The extra precision algorithm of the library subroutine gave an extra bit to the intermediate results, so that the algebraic truncation error and the rounding error both decreased, and no awkward estimates had to be made. In the early fifties, the computers did not run on long calculations without inspection of intermediate results, so probably the time had not come for the fully automatic methods.

Bit by bit calculations were a natural technique, particularly where speed was not important but the size of a program was. An example using this technique was a method of computing logarithms by repeated squaring and doubling. The extension of this method to computing the inverse cosine was natural, and I blundered. It is clear that the subroutine would be inaccurate for small angles, and it was tested over about one hundred random numbers. One might even say, that as the program was derived using a loop invariant, it had been proven correct. Van A.Wijngaarden pointed out that the error function was spikey, losing up to half precision where the angle times a power of two was near multiples of one hundred and eighty degrees. Thus a few evaluations were not sufficient to test the subroutine. This taught me a lesson which has endured to this day.

Another process which had to be adapted for automatic calculation was finding the root of a function. Given a computable function and two arguments whose values had different signs, find the root in that range. Nearly all hand processes rely on the application of intelligence at some stage. The obvious automation is the repeated subdivision of the interval bracketing the root. This can be done by using linear interpolation to find an inside argument, evaluation of the function at that argument and replacing the limit value with the same sign by the new value. Thus the automatic program has to check for a zero functional value and stop there, else hunt for two adjacent arguments of opposite sign in the overall range.

The simple method soon slows to a first order method as the curvature, positive or negative, causes one end of the range to be nibbled away, rather than allowing the linear interpolation to be effective. The EDSAC library subroutine avoided this drastic slowdown by the following method. While a functional value was not replaced, its value was halved, except for the first time. Thus the error is roughly cubed every three evaluations. By a close attention to rounding and other essential details we can arrange that it will stop when the adjacent arguments are as close as possible and their functions have opposite sign.

Floating point subroutines were developed for the problems where programmed scaling was difficult or impossible. Interpretive subroutines were used for this purpose so that the sequence of calculations could be done more readily than by repeated calls to subroutines. These subroutines slowed the computer by an order of magnitude, so that although they made the calculation programming easier, their use was restricted.

Sines, cosines, logarithms, exponentials etc. were evaluated using economised power series rather than by the hand technique of tables and interpolation. An interpolation subroutine was available which used Neville's method, but this was rarely used.

Checking of programs

How did we ensure the results were correct? Programs had the advantage that errors which were removed did not return. Computer errors usually gave rise to obvious disasters such as looping forever, stopping, or printing crazy results. When such programs were rerun one might get the correct results as most computer errors were intermittent.

When a computer is unreliable, it has a disproportionate effect on the debugging process. It is human nature to first blame the computer even if experience shows that program errors are much more likely. People often insist on a rerun to make sure no intermittent has occurred, before analysing the data to locate the remaining faults.

The errors were removed by trial and observation. A program was run and information was collected by various means. Some was obtained by observation of numbers changing in the store. This could be done very well on EDSAC as a rotary switch enabled the content of a mercury tank to be observed. Usually one noticed if it was changing, or a negative counter approaching zero. It was not usual to obtain the precise value of a number in this way. Many programmers aranged their variables in the sixteen locations of a single tank so that they were easy to observe.

A loudspeaker gave the rhythm of a program so that discrepancies were noticed immediately without conscious effort for programs which had run before.

When a program came to a premature end, or was stopped because it was looping, a post mortem tape kept at the console was run and enabled selected portions of the store to be printed as numbers or orders. A modification to the computer enabled a telephone dial to select the portion of store that needed printing.

A stronger method of locating program errors due to Stan Gill was a trace subroutine, which printed the sequence of orders obeyed by the program, and although it slowed the program down by a factor of about twenty, was very useful for otherwise hard to find faults. The program tape had to be augmented to use such a program and usually a small extra tape called a jiffy tape was used for the purpose.

Check point subroutines were developed which enabled printing of the accumulator or other variables at selected points of a program for selected numbers of times. They were not much used. I had thought originally that such methods would be used for proving programs, but due to human nature they tended to be used as a last resort.

Corrections

Paper tape is very slow to correct or adjust. It involves copying and making changes. The tape preparation room did have equipment for this, but the work was slow and tedious. It was usual for program tapes to have either a stop order, requiring a manual restart, or else an inch or two of blank tape so that the pressing of a stop button would have an equivalent effect, before the starting directive at the end of a tape.

Thus by changing the tape in the paper tape reader to a jiffy or correction tape before the starting directive was obeyed, it was easy to modify and correct the program without repunching the entire tape. Then further corrections could be incorporated in the jiffy tape by extending that short tape. Thus the jiffy tape grew and it became worthwhile changing the main program tape -often under the incorrect assumption that the last error had been found.

Computer operation

The computer was rarely stepped through programs an order at a time as a means of locating program faults. Instead, we had testing periods when one could run a short program for a minute or two and get some information from it. As many people wished to make tests in a short time, there was plenty of advice given if anyone was using wasteful methods.

Even when operators were provided to run the computer, test runs were usually run by the authors of the programs. Production runs tended to be longer and were usually handled by the operators. The maximum length of run allowed during the day was half an hour, but a more usual duration was ten minutes.

At night the computer was handed over to various groups to use. Their competence had been assessed before they were allowed to do this. They were classified into fully and partially authorised users. This determined whether they could be in sole charge of the computer and what adjustments they could make to the computer hardware. There was no night maintenance in the early fifties so they worked until the computer broke down, and occasionly all through the night.

One of the night groups under the leadership of S.F.Boys, a theoretical chemist, did calculations of electronic wave functions lasting many tens of hours. His approach was very professional, all runs being repeated with separate program tapes. The terms of the six dimensional integration were themselves derived by algebra in the program.

As the computer began to yield useful results, a priority committee was set up to determine which submitted calculations were suitable, how they might be programmed and how much computer time they could use. Problems were submitted from most science and engineering faculties.

Computer improvements

During the life of the computer many improvements were made which enhanced its power and made it more reliable. The input speed rose from seven characters per second up to fifty characters per second. The directly coupled teleprinter working at seven characters per second was replaced by a laboratory constructed punch which went at thirty five characters per second. A B register was added to speed up order modification and magnetic tapes were connected to give an auxiliary store. A telephone dial was added to allow selective control while running. The output code was changed to a two out of five code so that simple errors were obvious in the printed results, and it required two errors of opposite types to cause a decimal digit to be printed as another.

University of Illinois

I went to Urbana, Illinois in September 1951 and stayed there for two years. During the first year the ORDVAC was finished, tested, sent to the Ballistic Research Laboratory at Aberdeen Proving Ground some seven hundred miles away in Maryland in February 1952. It was based on the I.A.S. computer at Princeton, which was under construction at that time.

The ILLIAC, an improved copy of the ORDVAC was not finished till about September 1952 so that for some time the ORDVAC was used in Urbana by sending programs and receiving results by the Teletype network.

These were more powerful computers than the EDSAC, being parallel and using cathode tube stores rather than delay lines. The store had about one thousand words rather than about five hundred words, but suffered

from a limitation in use -the read round ratio- which increased the programmers burden slightly. The engineers improved the ratio until by mid 1953 almost no inconvenience was suffered.

The programming system was similar to and derived from that of the EDSAC, although each program tape had to have its own bootstrap starting program copied on its front. The larger store size enabled more matrix calculations could be done. In particular, a program was made following a suggestion by H.H. Goldstein, based on the Jacobi method. This was used to find the eigen values and vectors of symmetric matrices. It was used a surprising amount and rather more for factor analysis than physical problems. It was a very apt program for the time.

It was a small program so that most of the program could be used to store the matrix or matrix and vectors. It simply gave the answers in all cases, with no special cases. Although it was slower than methods such as Givens etc., for the small size matrix which it could handle, the factor was nowhere near the assymptotic factor for large matrices. The program could handle up to a matrix of order forty three for eigen values and up to twenty five for both eigen values and vectors.

Another matrix method developed was for solving linear equations. The equations were reduced as they were read in so that only a triangular matrix of reduced coefficients was stored, and larger matrices could be dealt with efficiently, up to about forty three equations.

The sizes which could be handled were small by present day standards. In those days most matrices were generated by hand and this tended keep their size small.

The use of the ILLIAC was integrated into the teaching of undergraduates and became part of the curriculum in a way which was unique at the time. It became an accepted and very effective research and teaching tool, but but I will omit discussion of this as I had little first hand knowledge beyond 1953.

Unsupervised calculations

During the summer of fifty two and fifty three the ILLIAC tended to be underused. The staff and students were mostly elsewhere. This was an effect of nine month staff contracts and lack of air conditioning except in the computer room.

In the first of these summers the calculation of e to 60,000 places of decimals was done and in the second the checking of the primality of

$$2^{8191} - 1$$

by the Lucas test. In both these cases, the program was completely checked for arithmetic, operator, punch, and reader errors. It had half hour runs on a computer with mean free time between errors of about six to ten hours. The calculations took about fifty hours. These programs being checked were part of a scheme to assess the performance of the computer -therein lay the justification for this type of calculation.

I was convinced that both of these programs were correct. It was pointed out to me later, that the printed result claiming to be e was e-2, the fractional part of e, so that in this carefully checked program about nine tenths of the printed decimal digits were wrong!

This has affected my subsequent attitude to proven programs.

Preparation of program tapes

This continued to be a very tedious affair. Each program tape was composed by copying subroutines and punching the rest of the program and data. Thus users spent more time in preparing tapes than using them.

One episode I can remember in particular, was when Joe Wagstein of the NBS visited us. We explained how easy our computer was to use and he gave us a problem. The printed results were available in minutes. The reasons for this speed, which was not typical was that it was a Sunday so we had full access to the computer, and that for teaching classes we had prepared a class tape which contained the bootstrap and the important library subroutines. This class tape and a rapidly prepared jiffy tape sufficed for the problem. It was the fastest I had ever done a complete calculation.

EDSAC 2

I returned to Cambridge in September 53, and continued to evolve the use of EDSAC. After the ORDVAC and ILLIAC, it seemed a very slow and unreliable computer. There was a large amount of crystallographic and radio telescope work some of which was done by the fast fourier transform.

Rather than give more details about the first EDSAC, it is perhaps worth discussing the design of the EDSAC 2, which incorporated our experience to date and came into service in 1957.

This was a forty bit word parallel computer with two orders per word. It was made with tubes, the store used ferrite cores, the arithmetic unit was bit sliced, and the total number types of pluggable units was about 18 with nearly all the computer being made of few types. Thus it was intended to be reliable and easily maintainable.

Floating point was included to make calculations easier. The computer was micro programmed, and had a ferrite core read only memory of 768 words. The user store was 1024 words.

The fixed memory enabled many useful subroutines to be permanently available. These included sine, cosine, polar conversion, logarithms, exponentials, solution of differential equations. Later on matrix division was added to the permanently available subroutines. The fixed memory included an assembler and set of print

subroutines which enable input and output to be done elegantly and readily throughout the execution of a program. The micro program and fixed store cooperated to make a trace which enabled the flow of control in a program to be printed.

The micro program enabled designed orders to be incorporated rather than accidents of hardware. For example, the fixed point order to store the accumulater caused the rounded result to be placed in the store while the accumulater was left unchanged. The special case where rounding would cause overflow was done by choosing the nearest number in range. Floating point rounding was done equally carefully although each operation caused a packed rounded result to be left in the accumulator rather than the extra precision unpacked version.

The order code was orthogonal in the modern sense, and some orders used many micro orders to produce correct results, as in the case of division with remainder.

As both fixed and floating point was provided, there were two versions each of the functional subroutines. Because of the need for space economy, most of the calculations were done for the fixed point and rounded for the floating point version. Thus the precision of the subroutines, particularly the floating point subroutines was almost the maximum possible. The floating point representation used 32 bit fraction and 8 bit exponent. The range and precision sufficed for most calculations.

Convenience of use was one of the main objects, so that in addition to producing well designed orders and subroutines, error detection facilities were built in. All unused codes in the order code caused an immediate "report". This printed the location, offending order, content of the accumulator and modifier registers and stopped the computer. Similar reports were caused by using non standardised floating point numbers in floating point orders, input syntax errors, untested overflows etc. Before the input of a program tape, the store was normally cleared to a value -all ones- which would cause a report if used as an order or a floating point operand.

Performance

The computer could do a simple instruction in about 20 microseconds while multiplication needed about a quarter of a millisecond. The pair of paper tape readers read at the rate of 1000 characters per second while the fastest output punch could do 300 characters per second. In the early years the final output was still printed by teleprinter.

The computer was designed with ease of use as a primary design consideration. We believe we achieved this aim. The complete guide to programming and the reference guide was 64 small pages. Users were able to run simple problems after a few hours tuition. It was my favourite computer and the last one which could be designed as a whole, without running into various compatability compromises. The ease of use of EDSAC 2 delayed the advent of programming languages in Cambridge for some years.

Teaching

Both at Urbana and Cambridge, the computer was used in an "open shop" manner. Users were expected to program their own calculations, assisted where necessary by the computer staff or their colleagues. Lectures were given to train newcomers and the first summer school for training outsiders was held in Cambridge in 1950. The use of the computer spread rapidly as successful users infected their friends.

One way information spread rapidly in the early days, was by the "grapevine". While users were waiting their turn to use the computer, information was exchanged about new procedures, machine weaknesses, successful ploys and so on. This rapid "documentation" system, contributed to the success of open shop policies. Closed shop policies of restricting the use of the machine to coders, who solved the problems of others, would have been a failure at these Universities.

Review of the calculating methods used.

Early calculations done on computers could have been done by those experienced in the use of hand calculaters. There were not many with those qualifications.

The methods chosen were not particularly new but were selected for rather different properties than those needed for hand calculations. Thus binary chopping is tedious to do by hand while the use of large tables, essential for many hand calculations, was rarely used in programmed calculations In fact, while there were few computers about, many thought the sole use of computers would be to generate tables suitable for hand calculation.

The methods which turned out well suited for computers were Runge-Kutta methods for differential equations, Gaussian methods for quadrature, economised power series for functions, Jacobi for eigen values, and fast fourier methods for transforms.

The problem of getting programs correct dominated the early use of computers, and users views of their computers were determined by turn round and methods for removing errors from programs rather than the numerical analysis problems.

Another important factor was the size of program. Interesting problems tend to be near the limits of the computer, so that program space as well as running efficiency was of importance.

Later on reliability became more important than space and extra facilities could be incorporated. A good example is square root. Early subroutines failed on zero, usually looping forever. Later versions incorporated tests so that zero was dealt with correctly, and negative arguments were detected, but they were longer programs.

A factor of significance was the reliability of the library routines. In the very early days, there was a tendency to accept all for the library. This soon ran into problems. Many programs were developed under problem specialised limitations. This meant they did not work under general circumstances. Until a strong discipline was established of thorough testing and rejecting many submissions the useful subroutines were hard for users to find. This discipline appeared to be rediscovered a number of times at different locations. Nowadays, the well constructed subroutines of NAG etc. are available for serious computation.

Nevertheless, the early computers did calculations which would not have been possible otherwise. The early work at Cambridge contributed directly to three Nobel prizes.

References

R.A. Brooker and D.J. Wheeler, Floating operations on the EDSAC. M.T.A.C., vol 7, p 37 (1953)

A. Burks, H.H. Goldstein and J. Von Neumann, Preliminary Disscusion of the Logical Design of an Electronic Computing Instrument. Institute for Advanced Study, Princeton (report). (1946)

S. Gill, A process for the the step by step integration of differential equations in an automatic digital computing machine. Proc. Cam. Phil. Soc., vol 47 p 96 (1951)

S. Gill, The diagnosis of mistakes in programs on the EDSAC. Proc. Roy. Soc., A, vol 206 p 159 (1951)

Moore School of Electrical Engineering, Theory and techniques for design of electronic digital computers. Lectures given to a special course at the Moore School 8 July to 31 August 1946. Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia. (report in 4 volumes), (1947-8)

R.E. Meagher and J.P. Nash,The ORDVAC. Proc. of the Joint AIEE-IRE Computer Conference. New York, AIEE p37 (1952)

D.J. Wheeler and J.E. Robertson, Diagnostic programs for the ILLIAC. Proc. IRE 41-10 p 1320 (1953)

D.J. Wheeler, Programme organsisation and initial orders for the EDSAC. Proc. Roy. Soc. A, vol 202 p 573 (1950)

M.V. Wilkes, The EDSAC-an electronic calculating machine. J. Sci. Inst., vol 26 p 385 (1949)

M.V. Wilkes, D.J. Wheeler and S. Gill, The preparation of programs for an electronic digital computer with special reference to the EDSAC and the use of a library of subroutines. Addison Wesley Press, Cambridge, Mass. (1951)

C.R. Williams, A review of ORDVAC operating experience. Proc. of the Eastern Joint Computer Conference. New York, IRE p 91 (1953)

MATHEMATICAL SOFTWARE AND ACM PUBLICATIONS

John R. Rice
Department of Computer Sciences
Purdue University

Mathematical software started as a scientific activity almost as soon as serious scientific computing. The field was brought into focus at the symposium *Mathematical Software* held at Purdue University on April 1-3, 1970. The symposium's organizing committee was John Rice (chairman), Robert Ashenhurst, Charles Lawson, Stuart Lynn and Joseph Traub. It was sponsored by ACM and SIGNUM and financially supported by the Office of Naval Research. Mathematical software was defined then as *the set of algorithms in the area of mathematics* and it was noted that this definition is much broader than traditional numerical analysis. Even today there are large areas of mathematical software which have yet to be studied systematically or seriously (e.g., geometric algorithms).

The first chapter of the symposium proceedings, Mathematical Software [Rice, 1971] presents a brief history of the field up to that point. It is noted there that the first mathematical software published was an EDVAC machine language program to convert base 10 integers to binary; it was in Mathematical Tables and Aids to Computations (now called Mathematics of Computation) on pages 427-431 of Volume 3, 1949. Further noted is that the book [Wilkes, Wheeler and Gill, 1951] contains a thorough discussion of the mathematical software (subroutine library) for the EDSAC. The second chapter of Mathematical Software is *The Distribution and Sources of Mathematical Software* which summarizes the state of the field as of 1970. The recent book, Sources and Development of Mathematical Software [Cowell, 1984] contains as first chapter the essay *Observations on the Mathematical Software Effort* by W. J. Cody. Many of the other 13 chapters of Cowell's book contain historical remarks about specific mathematical software areas.

Chapter 3 of Mathematical Software is *The Challenge for Mathematical Software* which raises many points still completely unresolved. It concludes with recommendations for the establishment of:

A Journal of Mathematical Software
A Center or Focal Point for Mathematical Software

The implementation of the first recommendation is the focal point of this article, the other recommendation has yet to be carried out. Perhaps mathematical software is now too big for a "Center" to cover the whole field, but a focal point would still serve a very important scientific function.

ESTABLISHMENT OF THE ACM TRANSACTIONS ON MATHEMATICAL SOFTWARE

The Software Certification Workshop was held at Granby, Colorado in August 1972, sponsored by the National Science Foundation. Thirty-one people attended and one short session was devoted to a discussion of a Journal of Mathematical Software. As a result, John Rice organized a one day meeting at Argonne National Laboratory with Tom Hull, Stuart Lynn and Joseph Traub to explore the possibility seriously.

The meeting was held on November 3, 1972 with Wayne Cowell as host. By that time discussions of interest had been held with Academic Press, ACM, SIAM and SIGNUM. All aspects of the journal were discussed at Argonne and the following points of agreement were reached:

* Establishment of a Journal of Mathematical Software should be pursued even though there were 10 journals identified that claimed they would publish mathematical software papers. None of these seemed serious (now, fifteen years later, we see that most of them were not serious).
* Lloyd Fosdick would be invited to join the group (he was then editor of the Algorithms section of the Communication of the ACM).
* A professional society publisher would be preferable to a commercial publisher.

Attention was thus focused on ACM and SIAM as potential publishers. The new journal would be coordinated with the ACM Journal of Collected Algorithms (CALGO) in a formal way which might mean a cooperative arrangement between ACM and SIAM. In the next months obstacles at both ACM and SIAM became clear. First, ACM was in the midst of a financial crisis. One faction within ACM claimed that the crisis was due to subsidizing technical journals. Even though it was well known in other societies that journals generally (and

often handsomely) support other society activities, ACM's accounting system could not provide any reliable information on the profitability of ACM publications. Thus there was considerable opposition to a new journal and some who were in favor in principle were cautious because of the financial implications.

At SIAM, there was concern by the editors of two existing journals (J. of Computing and J. of Numerical Analysis) about overlapping areas. The editors of both journals thought it would be appropriate for each to simply expand and include the mathematical software area. There was also the tricky problem of dividing expenses and control with ACM's CALGO publication. Finally, there was some reluctance within SIAM to become so involved with computing.

During the spring of 1973 the definition of the proposed journal was polished (a complete "sample" issue was constructed, for example) and many operational issues decided. Most importantly, John Pasta at the National Science Foundation reacted favorably to the idea of sponsoring a conference on mathematical software whose proceedings would be the first issue or two of the new journal. This would greatly reduce financial risk.

By the summer of 1973 it became clear that the obstacles at SIAM would be very difficult to overcome. SIAM wanted to move toward mathematical software without becoming tainted with computer programs and no one could see how to do this effectively. The National Science Foundation funding for the conference Mathematical Software II seemed assured, so the efforts were focused on convincing ACM that their financial risk was tolerable. There were two other factors within ACM which supported establishing the new journal, the ACM Transactions on Mathematical Software (TOMS). First, it would remove the Algorithms section from the Communications of the ACM and this appealed to many both for financial and aesthetic reasons. Second, it would set a pattern for other specialty journals which many felt that ACM should be publishing. For example, the ACM Transactions on Computer Systems and Programming was included abstractly in ACM discussion documents in the summer of 1973.

During the winter of 1973-74 many financial analyses were made, all of which indicated that the National Science Foundation support would reduce the first year cost to ACM of the new journal to a very small amount, even with pessimistic assumptions ($5300 was the "final" figure) and thereafter the journal would be profitable (making a profit of over $15,000 in the third year). These figures were based on first year assumptions, of 333 individual subscribers and 200 institutional subscribers. The actual numbers of first year subscribers were 1072 individuals and 351 institutional, so TOMS made a substantial profit even in its first year. The launch of the ACM Transactions on Mathematical Software was formally recommended by the ACM Publications Board on May 8, 1974 and approved by the ACM Council the same week.

The conference *Mathematical Software II* was held at Purdue on May 29-31, 1974. There were over 225 attendees with 82 papers presented including 22 invited presentations. All were considered for publication in TOMS and most went through the normal refereeing procedure (a number of the papers were submitted to other journals). The conference receive $31,000 in support from the National Science Foundation of which over $16,000 went to help support the publication of 22 papers in the first issues of TOMS. The "proceedings" of Mathematical Software II had 324 pages and was strictly limited to conference attendees (only 250 copies were printed) and not made available otherwise. This was to prevent double publication of papers in the proceedings and TOMS. This has led to recurring confusion because there are books Mathematical Software [Rice, 1971] and Mathematical Software III [Rice, 1977], so people assume there is a Mathematical Software II (which there is, [Rice, 1974]) and that they should be able to obtain it (which they cannot).

ALGORITHMS

The publication of algorithms is one of the main functions of TOMS. The biggest challenge faced by the editors of TOMS is to apply normal scientific refereeing procedures to algorithms and to make them available in a reasonable way. The difficulty faced by the editors is seen right at the outset, there is wide disagreement about the definition of an algorithm. The traditional mathematical definition is : **algorithm** - *an unambiguous set of instructions for a machine.* That, it turns out, is a quite ambiguous definition and there are three colloquial definitions in widespread use as follows:

Method:	A general approach or strategy used in computing, something not defined in complete detail. Examples are divide and conquer, steepest descent, predictor-corrector, and finite differences.
Algorithm:	A set of mathematical steps for an abstract computation. Examples are the Euclidean algorithm, bubble sort, Gauss elimination (no scaling or pivoting) and the formula for the roots of a quadratic polynomial. Algorithms are thought of as short and restricted to a single "purpose".
Computer Program:	Something written in Algol, Ada, Fortran, Assembler, etc.

A little thought shows that the computer program definition is the closest to being an algorithm in the precise mathematical sense. However, the colloquial definition of algorithm is so widespread that there is continual confusion about what it means to publish "algorithms".

In this section we discuss two aspects of publishing programs that the editors of TOMS have found difficult: evaluating the performance of algorithms and distributing the algorithms. Even after 12 years of publication, TOMS (and CALGO) are the only journals that handle these aspects with a high quality of scientific publication.

The ACM algorithms started in the Communications of the ACM as short Algol programs. Thus they originated as algorithms both in the precise and colloquial senses. The first 225 ACM Algorithms were not refereed at all and many were trivial in nature.

In the 1970's two significant editorial policies were adopted. First, algorithms to be published had to have significant utilitarian value and second, algorithms had to meet high standards. Establishing standards and enforcing them has created a large burden on the editors, the referees and the authors. The acceptance rate for algorithms is less than half that of regular papers. It takes two or three times as much effort to process an algorithm and, unfortunately, it also usually takes twice as long. As one might guess, it takes relentless effort by the editors to enforce standards of style and documentation. Some authors cannot believe it when their otherwise great software is rejected because the documentation is inadequate. The result is, however, worth the effort; most (alas, we are still striving for perfection) ACM Algorithms perform a worthwhile task in a reliable, efficient manner and are easily used. These algorithms are truly valuable, at today's cost of software development, the algorithms in TOMS for a given year are worth almost $1 million. The subscribers to the ACM Collected Algorithms certainly get a bargain! The algorithms have become much longer, so long that some people say they are "programs" and not "algorithms". These words are nearly synonymous to me, but others see a large distinction. The TOMS Algorithms range from 48 to 55,560 lines of code. The longest ACM Algorithm (Algorithm 607: Text Exchange System by W. V. Snyder and R. J. Hanson) would have a listing of about 925 pages. Eighty four (out of 148) of the TOMS Algorithms have over 1,000 lines and eight have over 10,000 lines. Needless to say, few algorithms are printed in full in TOMS and even in CALGO the longer ones are given on microfiche.

Algorithm Performance

The standards for refereeing algorithms include the criterion of performance. Algorithms which perform significantly better than any previously known algorithm (for an interesting problem) are clearly valuable scientific contributions. In many instances, the sole objective of a program is to be able to solve a particular class of problems. There are no alternatives so no efficiency comparisons are made and less than high reliability might be acceptable for some difficult classes of problems. It is common, however, that competing algorithms do exist and then the TOMS referees must judge the relative performance of the algorithms. Here, all qualities come into consideration, from efficiency to accuracy to robustness to long term maintenance.

Once one gets away from elementary or simple problem areas, it becomes very difficult to say which problems a particular algorithm should solve correctly. Most of the algorithms considered in TOMS are applicable to unsolvable problem classes. The term "unsolvable" is used here in its strong, technical sense, that is, one can show no algorithm exists which can solve all the problems in the class under consideration. As a result, given an algorithm, one can usually construct problems where it fails miserably. In principle, mathematics provides a mechanism to exclude such problems. For example, one might specify that an algorithm is to be applied only to differential equations whose solution has its fourth derivative bounded by 1000. One can then hope to prove theorems about the algorithm's performance. Such an assumption is an *unverifiable hypothesis*, there is no algorithm to determine whether the assumption is satisfied. Most mathematical analysis assumptions are of this type and useless in practice.

The result of this uncomfortable situation is the development of batteries of test problems which are hoped to represent the spectrum of problems that occur in real applications. One recent paper [Shampine, 1981] in TOMS is devoted to a detailed critique of the test problem sets that are currently being used to evaluate programs for solving stiff ordinary differential equations (a particularly difficult and important class of problems). The need for care in choosing test problem sets is illustrated by the example of solving linear equations; the early practice of generating test matrices at random lead to completely misleading results about the robustness and reliability of algorithms.

A consequence of the fact that most algorithms in TOMS apply to unsolvable problems is that they contain heuristics in certain key places. For example, programs that involve the convergence of something use a heuristic test; the robustness of such software is often directly related to quality of the heuristic used. The presence of heuristics in the programs make the usual "software validation" or "program proving" techniques partially inapplicable as there is no concept of correctness of a heuristic, only performance is meaningful to discuss.

There is a large contrast in the material in TOMS and the June, 1982 issue of ACM Computing Surveys (CSUR) which is devoted to the validation and testing of software. Even the CSUR article entitled "Validation of Scientific Programs" has almost no overlap with the material in TOMS or the principle issues discussed by TOMS' algorithm authors. The CSUR paper concentrates on topics such as requirements analysis, design analysis, source code analysis and code auditing. One might say it discusses the application of good programming practices. In the performance evaluation papers and algorithm refereeing for TOMS it is assumed as a matter of course that programs are developed with good programming practices. One reason for the high rejection rate for algorithms submitted to TOMS is that, alas, this assumption is often false. However, once evidence of poor code is seen, a program is summarily excluded from further consideration — either as a TOMS algorithm or as a candidate for serious performance evaluation.

Two recent TOMS' papers evaluate software in truly difficult problem areas, optimization and solving nonlinear equations, and the results illustrate the nature of many of the problem areas involved in TOMS' papers. Both studies involved 8 algorithms (programs) regarded as being among the best available for the problem areas. In the first study [Hiebert, 1981] all but one of the programs solved the "standard" set of 36 test problems. No program could solve all, or even almost all, of the more difficult set of test problems even though every problem could be solved by some program. The second study [Hiebert, 1982] used 57 problems, mostly difficult, each in three versions (according to the scaling of the problems). The two best performing programs were able to solve only 98 of the 171 test problems (not the same 98). As one has learned to expect in such difficult problem areas, no one program is best and one needs either a set of programs to apply or to "tune" one of the programs for the particular problem at hand. Some of the test problems were not solved by any of the programs.

Table 1. Volume of Algorithms ACM distributed by type of distribution. See TOMS for detailed definitions of the types:

	1976	1981	1982	1983	1984	1985	1986*
Single algorithms							
listing	5	106	89	111	114	71	—
cards	17	37	30	20	14	6	—
diskettes	—	—	—	—	—	75	102
netlib**	—	—	—	—	—	475	1839
Quarterly issues	22	57	113	95	96	41	12
Volume issues	—	38	73	140	153	96	34
5 year tapes	—	48	38	32	46	96	74
5 year subscriptions	—	—	—	—	—	3	7

*1986 data is for first 9 months except for netlib

**netlib is a network distribution service operated at Argonne National Laboratories by Jack Dongarra and Eric Gross

Algorithm Distribution

Even before TOMS was conceived, Lloyd Fosdick had started to explore better ways of distributing algorithms to the scientific community. The practice in the Communications of the ACM of printing hundreds of lines of code was clearly inadequate. Serious study of an algorithm includes using it and it is both tedious and error prone copying code from the printed page. Further, pages of codes is particularly dull if one does not have a serious interest in the algorithm. Fosdick initiated a distribution service of machine readable forms of algorithms. He selected some algorithms for distribution and then distributed them using the ad hoc resources of his department at the University of Colorado. His experience showed that such a service was feasible, that there was a real demand for it and that there were substantial operational hurdles to face.

One of the original objectives of launching TOMS was to establish a systematic, reliable distribution service for the ACM Algorithms. There were serious obstacles. First, the ACM publication staff did not understand the issues involved, was unhappy to contract such an important function to an outside organization and was completely unable to do it internally. Second, the volume of distribution would be low enough that the service would not be an attractive commercial venture.

After some time, Ed Battiste, President of IMSL, Inc. agreed to handle the distribution service as a "public service", charging only enough to approximately recover the direct costs of the service. It then took several months of negotiation to get the ACM publications staff to agree to this arrangement. The mathematical software community owes a large debt to IMSL for their service here, they have distributed the algorithm with a high level of professionalism and they surely lose money every year in this service. The success and changing nature of this service is seen from Table 1 where the volume of the service is given for the first full year (1976) and the past six years. Algorithm distribution on microfiche was initiated and then dropped after several years due to a lack of interest.

The ACM Algorithm Distribution Service is designed to remove artificial limits on length due to printing costs. The first step when TOMS was started was to publish only excerpts in TOMS itself, full text was published in CALGO. This allowed algorithms of 10, 20 or 30 pages in length to be "published" in TOMS, but printed in full in CALGO. In the late 1970's, microfiche supplements to CALGO were initiated so that a "small" segment of an algorithm would be printed in CALGO and the remainder printed on microfiche. Thus Algorithm 607 with 55,560 lines can be published even though it takes four microfiche sheets to print. Keep in mind that the *primary* publication medium for such an algorithm is not the printed version, but the machine readable version available from the ACM Algorithms Distribution Service.

REFERENCES

Wayne R. Cowell, *Sources and Development of Mathematical Software*, Prentice Hall, Englewood Cliffs, (1984).

K. L. Hiebert, An evaluation of mathematical software that solves nonlinear least squares problems, *ACM Trans. Math. Software*, **7** (1981), 1-16.

K. L. Hiebert, An evaluation of mathematical software that solves systems of nonlinear equations, *ACM Trans. Math. Software*, **8** (1982), 5-20.

John R. Rice, *Mathematical Software*, Academic Press, New York, (1971).

John R. Rice, *Mathematical Software II* (An informal conference proceedings), Purdue University, (1974).

John R. Rice, *Mathematical Software III*, Academic Press, New York, (1977).

Lawrence F. Shampine, Evaluation of a test set for stiff ODE solvers, *ACM Trans. Math. Software*, **7** (1981), 409-420.

M.V. Wilkes, P.J. Wheeler and S. Gill, *The Preparation of Programs for an Electronic Digital Computer*, Addison-Wesley, Reading, (1951).

The Pioneer Days of Scientific Computing in Switzerland

Martin H. Gutknecht
Eidgenössische Technische Hochschule
CH-8092 Zürich

Abstract. Scientific computing was established in Switzerland by E. Stiefel, assisted by H. Rutishauser, A.P. Speiser, and others. We cover the years from the foundation of the Institute for Applied Mathematics at the ETH in 1948 to the completion of the ERMETH, the electronic computer built in this institute, in 1956/57. In this period, Stiefel's team also solved a large number of real-world computational problems on another computer, Zuse's Z4, rented by the institute. Along with this work went major contributions to numerical analysis by Rutishauser and Stiefel, and Rutishauser's seminal work on compiling programs, which was later followed by his strong commitment in ALGOL.

We have tried to include some background information and to complement H.R. Schwarz's article [Scw81] on the same subject.

1. Getting started: The foundation of the Institute for Applied Mathematics

When looking for a date marking the beginning of computer science and scientific computing in Switzerland one is soon thinking of January 1948 when the Institute for Applied Mathematics at the Swiss Federal Institute of Technology in Zurich (Eidgenössische Technische Hochschule, or, briefly, ETH) was founded under the directorship of Professor Eduard Stiefel (see Sec. 5 for Stiefel's biographical data). Up to then, Stiefel was known in the scientific world as an excellent topologist, who in his thesis written under Heinz Hopf had laid the basis for the theory of vector fields on manifolds. None of the seven papers he had published before 1948 was on numerical analysis, but in his regularly held courses in descriptive geometry he got into contact with engineers and learnt of their need for constructive and computational mathematics. Moreover, during World War II Stiefel, as an officer of the Swiss Army, had to some extent worked on computational problems. When after the war he became aware of the development of computers and algorithms in other countries, in particular the USA, he realized the scientific and economic importance of this research for a highly industrialized country, and, through his personal initiative, he achieved the foundation of the Institute for Applied Mathematics. Its aim and purpose were the introduction of scientific computing on programmable machines in Switzerland. From the beginning Stiefel was backed up in his basic decisions by a Committee for the Development of Computers in Switzerland and by the Board of Directors (Schweizerischer Schulrat) of the ETH.

At that time electronic computers were not yet on the market, but many research institutions around the world were designing and building their own machine. Some relay computers, e.g. Aiken's Mark I (1944), and at least one machine based on electron tubes, Eckert and Mauchley's ENIAC (1946), were already running. In the USA several groups of researchers competed for the biggest and the fastest machine, and the costs of some of these projects exploded. There was no chance of receiving so much money in Switzerland, therefore it was clear that in relation to these American projects a Swiss machine had to be at a Swiss scale. In fact, at the beginning Stiefel's budget was very limited, and the technical equipment of his institute consisted just of a Madas mechanical desk calculator and a Loga drum, a cylindrical instrument combining various slide rules.

But Stiefel was also a very successful administrator, who was able to acquire grant money from public and private sources and to get contracts with private industry and even with the army. In contrast to the situation in the US, the latter is quite unusual. But it is very likely that Stiefel's military career, which ended at the high rank of a colonel, was beneficial for his projects. Later, from 1958 to 1966, Stiefel also played a significant role in local politics: He was an important member of the (legislative) community council of Zurich. Thus besides being a

truly innovative and highly competent mathematician in various areas Stiefel fits also into the image of the Swiss establishment as it is colorfully painted in McPhee's "Place de la Concorde Suisse" [McP84]. Clearly, being engaged and successful in so many disjoint activities required to be well organized, and Stiefel could in fact keep things running with seemingly very little effort.

When starting the institute it was of course very important to find good collaborators. In this respect, Stiefel was again highly successful: As assistants he chose the mathematician Heinz Rutishauser (see Sec. 6 for Rutishauser's biographical data) and the electrical engineer Ambros P. Speiser[1], both former students of the ETH. Rutishauser had left the ETH three years before and was working as a high school (Gymnasium) teacher, while finishing his excellent dissertation in complex analysis in his spare time. Speiser was just getting his diploma in electrical engineering with a thesis related to computers.

2. Learning from others: The trips to the USA

Although prototype computers were being constructed in various countries, the USA were clearly ahead in computer technology. Hence, it was decided that Stiefel and his two assistants should visit the USA to acquire some of the American knowhow. The following is a brief summary of Stiefel's report [St49] from the first trip, from October 18, 1948, till March 12, 1949. Rutishauser and Speiser stayed for longer, until the end of 1949.

The first stopover was at the Mathematical Center in Amsterdam, where Dr. v. Wjingarden directed the construction of a relay computer and a mechanical integrator, but provided also a scientific computing service to the Dutch industry. Next, Stiefel spent seven weeks in New York, mainly at the IBM Watson Laboratory for Scientific Computation at Columbia University (Dr. Eckert and Dr. Thomas), where he had a chance to use a large selection of IBM computing equipment. In particular, he became familiar with IBM's Selective Sequence Electronic Calculator, a computer containing some 12'000 electron tubes. In New York, Stiefel visited also the Institute for Mathematics and Mechanics at NYU (Prof. Courant, Prof. Friedrichs) and the Computation Laboratory of the National Bureau of Standards (Dr. Lowan, Dr. Salzer); both institutions were still without electronic computers, though the NBS had already published some 30 volumes of mathematical tables, mostly produced on desk calculators and IBM punched card machines, which were wide-spread computing aids at that time.

In Washington Stiefel spent two weeks at the Office of Naval Research and the National Bureau of Standards. At the ONR, Dr. Mina Rees was then Head of the Mathematics Branch. Her continuing help in the organization of Stiefel's trip, in particular, her importance for giving him access to various computers in laboratories of the Navy and the Army, are gratefully acknowledged in Stiefel's report.

It was, of course, a must to visit Boston, where at the Harvard University Prof. H. Aiken was designing his Mark III, while Mark I was running "24 hours a day and 7 days a week", as he said. During Stiefel's three weeks' visit some of this computer time was consumed by a free boundary problem submitted by Prof. G. Birkhoff. The latter informed Stiefel also of a new numerical method, which is now called successive overrelaxation! In addition, Stiefel had discussions with Prof. S. Bergmann and his group on the use of kernel functions in conformal mapping. Certainly these discussions aroused Stiefel's interest in relaxation methods, which on one hand led in cooperation with Prof. M.R. Hestenes to the conjugate gradient method [HSt52, St52a] and, on the other hand, to a series of papers by Stiefel, Rutishauser, Engeli, and Ginsburg on relaxation methods in general, and their interrelations, see in particular [St58, EGRS].

Finally, Stiefel was for two weeks at the Institute of Advanced Study in Princeton, where he tried to learn from Prof. J. von Neumann, who was at the leading edge of both theoretical computer science and hardware design. Rutishauser was left there, monitoring the work at von Neumann's computer project while Speiser was left in Boston, working on Aiken's Mark III.

Besides the stops mentioned above, Stiefel visited shortly a number of other places, either for information about computers or for giving lectures. He also spent some time in Princeton and in Chicago working on his former research subjects, geometry and continuous groups.

Throughout his trip Stiefel was highly impressed by the wide-spread use of mathematical and numerical methods in scientific, industrial, and military research, and the confidence of government and industry in this approach. He noted that this wide-spread use was not only due to the existence of large electric and electronic computers, but rather to the different American attitude towards applied mathematics. Actually, many of the computations were still done on desk calculators, punched card machines, and analog computers.

Concerning the construction of computers Stiefel had learnt that many unexpected difficulties had appeared with large machines, in particular concerning their reliability. None of the six types of memory he

[1] Ambros P. Speiser (*November 13, 1922; dipl. El.-Ing. ETH, 1948; Dr. sc. techn., 1950; Privatdozent at ETH, 1952; Prof., 1962) became in 1955 the first director of the IBM Research Laboratory in Zurich. He is now Director of Research of Brown Boveri & Cie., Baden (Switzerland). 1965-1968 he was president of IFIP.

had seen satisfied him. In his discussions with Prof. Aiken he got assured that a relatively small and slow but reliable machine could be built with a small budget, and that such a machine could nevertheless become a very useful and cost-effective tool for Swiss science and industry. A first tentative design of this machine called ERMETH (Elektronische Rechenmaschine der ETH) was worked out by Speiser in Boston and, after his return, in Zurich. It is documented in Speiser's doctoral thesis [Sp50].

Let me anticipate that from July 1951 till February 1952 Stiefel stayed once more in the USA, now mainly in Los Angeles at the Institute for Numerical Analysis of the National Bureau of Standards and at the UCLA. In his report [St 52] he again called attention to the great support for and confidence in scientific computing and to the different situation in Switzerland, where only few mathematicians had a chance of finding a job in industry, so that some of the best chose to emigrate to the USA. (Well-known examples are W. Gautschi, P. Henrici, H.J. Maehly.) Further remarks on the few possibilities for applied mathematicians in Switzerland are also found in some of Stiefel's later annual reports until 1956; afterwards the situation started to change.

Concerning Stiefel's research at NBS it is well known that he was working with M.R. Hestenes on the conjugate gradient method [GOL87, HSt52].

On this second trip Stiefel got also new information on some of the computer projects. He noted that quite a few had been abandoned, that only three of the "superfast" machines were working regularly on mathematical problems (namely SEAC at the Computation Laboratory of the NBS in Washington, Mark III at Harvard University, and Whirlwind at MIT, which had been a secret project at the time of Stiefel's first trip). In particular, von Neumann's EDVAC was still not working. (It became operational in 1952.) So, Stiefel was confirmed in his opinion that the ERMETH should be simple and reliable.

3. Computing on the Z4

When in Spring 1949 Stiefel came back from his first trip to the USA he anticipated that the design and the construction of the ERMETH would take several years. In order to promote numerical computations with his institute he needed some other equipment that was immediately available. He first thought of renting IBM punched card machines [St49]. But then he learnt that the German Konrad Zuse had been able to save one of his relay computers, the Z4, through the devastating time at the end of World War II by hiding it in a cow stable at Hopferau in the Bavarian alps. After inspecting the Z4 there on July 13, 1949, Stiefel and Zuse worked out a lease: ETH rented the Z4 for a period of five years for a total of SFr. 30'000.

Konrad Zuse (*June 22, 1910) was a highly gifted civil engineer who had started to design and assemble a mechanical computer called Z1 in his parents' living room in Berlin in 1936. Its logical design was far ahead of the time. Zuse's basic concept, although not yet fully implemented in the Z1, included full programmability and remained the same up to the Z4. The basic number representation was already in binary floating-point. However, due to the limited accuracy of the mechanical parts the Z1 was never fully operational. But after replacing the processor by one built from relays, Zuse had a working computer, the Z2, in 1939. Two years later Zuse finished the Z3, which contained in its processor and its memory some 2600 relays and which many experts consider as the first programmable computer worldwide. The next model, the Z4 rented by Stiefel, was constructed from 1942 till 1945. It contained some 2200 relays and worked with normalized 32-bit binary floating-point numbers with 22-bit mantissa. A multiplication took 2.5 to 3 seconds. The program was read from two switchable punched tape readers. The Z4 was more powerful than the Z3, although it had again a mechanical memory (for 64 numbers). Old movie films were used as tapes, so there was a minimum of entertainment for the people operating the machine. (Although they did not have a projector!)

After 1950 Zuse kept on designing computers and some of them were fairly successful on the small German market. Zuse's work is well documented by his autobiography [Zus70] and the references listed there. The Z4, which is now exhibited in the Institute of History of Siemens in Munich, is also described in [Eng81, Scw81, Sp50a, St53, St54a]. Among the many interesting features we mention the unique handling of the value infinity and the hardware division based on Zuse's own ingenious algorithm, cf. Rutishauser et al. [RSS51]. Zuse made also a seminal early contribution to programming by formulating algorithms in his "Plankalkül" [Eng81, Zus48, Zus59].

Before its delivery to Zurich the Z4 had to be repaired and overhauled. Also, on proposal of Stiefel and his team conditional instructions were included. After its installation in August 1950, which was followed by some further servicing work, the Z4 proved extremely reliable, except for some gradually growing problems with mechanical parts, in particular the memory. Typically the Z4 was running day and night at the ETH, often unattended when working on a long job. The list of 55 projects that have been performed on it until its removal in April 1955 contain an amazing variety of subjects, e.g., a forth order PDE for the tensions in a dam, the eigenvalues of an 8×8 matrix from quantum chemistry determined by inverse iteration, a linear system with 106 unknowns, which came from a plate problem, solved by the conjugate gradient method, ODEs modelling rocket

trajectories, and so on. Some of these projects are described in the excellent survey by Schwarz [Scw81], who himself together with Dr. U. Hochstrasser was doing some of the most time-consuming computations, which were related to the design of a Swiss supersonic military aircraft [Hoc55, Scw56].

Of course, numerical experiments related to the basic numerical analysis research performed at the ETH at that time were also run on the Z4. For example, after Stiefel had returned from his second US trip, Lanczos's eigenvalue method [Ru53] and the conjugate gradient method of Hestenes and Stiefel [HS52] were coded. One must further mention Rutishauser's early investigations on the stability of numerical methods for initial value problems of ODEs [Ru52a], Rutishauser's qd-algorithm and LR-transform [Ru57], and H.J. Maehly's polynomial root finder [Mae54].

For some sparse matrix problems the code for the Z4 was extremely long (up to 6000 instructions) since there was no provision for address computation and thus the actual addresses of the nonzero elements in the matrix had to be used when calculating a sparse matrix vector product. To simplify the preparation of such codes, Rutishauser developed a program for computing these addresses and for producing the corresponding section of the code, see Schwarz [Scw81, Sec. 4] for more details. This, however, was just the beginning of his seminal work on "automatic coding" ("automatische Rechenplanfertigung"), the first peak of which is Rutishauser's Habilitation thesis [Ru52], in which he described in full detail a method for compiling the machine code for a certain problem by the computer itself from the mathematical formulas. He allowed for expressions with arbitrary levels of brackets and for loops with bounds depending on the data. Moreover, he discusses the loop unrolling (which nowadays receives much attention on vector computers). His examples include a program for solving a linear system by computing the LU decomposition column by column and then substituting forward and backward. Except from the fact that the keywords are in German, the program looks already like the body of an ALGOL procedure.

4. Constructing the ERMETH

While all this basic research and all these computations on the Z4 were going on, Stiefel's gradually growing group was also working hard on the design and the construction of the ERMETH. Speiser, since 1952 also Privatdozent (his habilitation thesis [Sp51] was on analog computers), was the technical director leading a group of five engineers and three mechanics. On the other hand, Rutishauser worked on the logical organization and its interrelation to his "automatic coding". It was in early 1953 only that it was decided to go for the electron tube technology instead of using relays. But by the end of the same year, the year when Rutishauser also worked out the qd-algorithm, the basic logical organization and the design of the arithmetical unit were close to being completed, and so was a prototype electronic memory, which was attached to the Z4 to replace the no longer satisfactory mechanical memory. However, to work out all the details of the ERMETH, to have the electronic and some of the mechanical parts manufactured by private companies, and to actually assemble the machine took another two and a half years. In July 1956 it was running for the first time, but still with a second prototype memory. In 1955 the Institute came in difficulties since Rutishauser had health problems and Speiser left for taking over the IBM Research Laboratory in Zurich. The electrical engineer Alfred Schai became the new director of the technical group completing the ERMETH; he is still the director of the Computer Center at the ETH. There were in particular problems with the large magnetic drum memory, which finally was installed in 1957. At the end of 1958 the cost for the ERMETH had accumulated to one million Swiss francs.

The ERMETH worked with 16-digit decimal words, each of which contained two instructions, one 14-digit fixed point number, or one floating-point number with 11 digit mantissa. A floating-point addition took 4 ms, a multiplication 18 ms. The magnetic drum could store 10'000 words. Hence, for the time the machine was not very fast, but it had a remarkably large memory. The machine contained some 1900 electron tubes and some 7000 germanium diodes. For more details see Schwarz [Scw81], who also discusses some of the applications and numerical investigations that were run on the machine. Schwarz moreover describes the most important development of the programming language ALGOL, in the basic design of which, I think it is fair to say, Rutishauser had a leading role. Schwarz himself wrote the ALGOL compiler for the ERMETH.

Among the contemporary articles on the ERMETH we mention [Sca57, Scl54, Sp54, S54a, Sp56, Sto54, Sto56]. There exist also a few copies of a manual [ERM58].

The ERMETH was in use at the ETH until 1963. The machine is now on display at the Technorama in Winterthur.

We conclude this article with short profiles of the two distinguished numerical analysts involved: Eduard Stiefel and Heinz Rutishauser.

5. Eduard Stiefel (1909-1978)

Biographical data: Born April 21, 1909, in Zurich. 1928-1932 student at ETH, 1932 diploma in mathematics, 1932/33 visiting positions at the universities of Hamburg and Göttingen, then assistant at ETH, 1936 lecturer. 1939 marriage with Jeannette Beltrami. 1942 Privatdozent, 1943 full professor at ETH. From 1948 director of the new Institute for Applied Mathematics at ETH. 1956/57 president of the Swiss Mathematical Society, 1958-1966 community councilman, city of Zürich, 1970-1974 president of GAMM. 1971 Dr. h.c. of the University of Louvain, 1974 Dr. h.c. of the University of Würzburg and the University of Braunschweig. † November 25, 1978.

Outline of his work: Stiefel's list of publications is published in a memorial issue of the Zeitschrift für Angewandte Mathematik und Physik, Vol. 30, No. 2 (1979). This issue also contains Stiefel's own comments on the list and a profile written by J. Waldvogel, U. Kirchgraber, H.R. Schwarz, and P. Henrici. In his comments on the bibliography, Stiefel devides his work into five periods:

1. Topology
2. Group theory and representation of groups
3. Numerical linear algebra
4. Numerical methods in approximation
5. Analytical methods in mechanics, especially celestial mechanics.

In all of these areas Stiefel made truly original and fundamental contributions. In fact, even as a newcomer to a field he was able to find a solution to some important basic problem, and in retrospect Stiefel's solution was simple and surprising at the same time.

With respect to scientific computation period 3 is the most important, but periods 4 and 5 must not be overlooked. The paramount contribution to numerical linear algebra is of course the conjugate gradient algorithm introduced in the joint paper with M.R. Hestenes [HSt52] and further investigated in a series of papers, in particular [St52a, St58]. However, one should also mention Stiefel's promotion of using variational principles for deriving the linear system from the physical problem [EGRS]. With this approach he put difference methods on a common basis with the finite element method.

Stiefel's period in approximation theory, although considered "less fruitful" by Stiefel himself, features the introduction of the single exchange version of the Remez algorithm and the proof of its equivalence with the simplex method, if the latter is applied to the discrete linear Chebyshev approximation problem [St59, St60]. The highlight of the fifth period is the introduction of the KS-transform (jointly with P. Kustanheimo) for regularizing Kepler's differential equation of celestial mechanics [KSt65].

6. Heinz Rutishauser (1918-1970)

Biographical data: Born January 30, 1918, in Weinfelden (Thurgau). 1936-1942 student at ETH, 1942 diploma in mathematics, 1942-1945 assistant at ETH, 1945-1948 Gymnasium teacher in Glarisegg and Trogen. 1949 Marriage with Margrit Wirz. 1948/49 New York and Princeton, 1949-1955 research associate at the new Institute of Applied Mathematics at ETH. 1951 Privatdozent, 1955 associate professor, 1962 full professor at ETH. From 1968 director of the computer science group at ETH. † November 10, 1970.

Outline of his work: Rutishauser's list of publications is contained in Research Report 82-01 of the Seminar für Angewandte Mathematik at ETH.

Rutishauser has come up with several of the most important ideas in numerical analysis and programming. In his Habilitation thesis [Ru52] he described the automatic compilation of a suitably formulated algorithm and thus introduced the concept of what is now known as compiler. Later his ideas on how to formulate algorithms have left traces in the design of ALGOL, for which he committed himself strongly [Ru67].

In numerical analysis Rutishauser's name is first of all forever linked with eigenvalue computations: The qd algorithm [Ru57, Ru63b, Ru76 (Appendix)] was meant for it, and so was its generalization, the LR transform, the basic principle of which reappeared later in the QR algorithm of Francis. This is also true with respect to spectral shifts, where Rutishauser found a cubicly convergent variant of the LR transform [Ru60]. Another truly original proposal is his algorithm, based on Jacobi rotations, for the reduction of band matrices to tridiagonal form [Ru63].

But Rutishauser contributed also to a number of other areas of numerical analysis. We mention his early work on the instability of methods for solving ODEs [Ru52a], his general definition and survey of "gradient methods" for linear equations [EGRS] (in this paper he also introduced a preconditioned conjugate gradient algorithm), his application of Romberg extrapolation to the notoriously difficult problem of numerical differentiation [Ru63a], his thoughts on the regularization of the nearly rank-deficient least squares problem [Ru68], his contribution to a survey of interpolation, quadrature, and approximation [SaS68 (Chapters H and I.II)], and his ideas on finding polynomial zeros [Ru69]. (These ideas have been completed in Kellenberger's dissertation [Kel71].) Finally one should mention Rutishauser's unfinished pioneering work on axioms for a reasonable computer arithmetic [Ru76 (Appendix)].

Acknowledgment. This work was initially planned as joint work with Professor P. Henrici, who unfortunately was not able to pursue this project due to a severe illness.

References

Eng81 Engeler E. et al.: Konrad Zuse und die Frühzeit des wissenschaftlichen Rechnens an der ETH. Dokumentation zur Ausstellung. Mathematisches Seminar ETH Zürich, 1981 (out of print).

EGRS59 Engeli M., Ginsburg Th., Rutishauser H., Stiefel E.: Refined iterative methods for computation of the solution and the eigenvalues of self-adjoint boundary value problems. Mitt. Inst. f. angew. Math. ETH Zürich, Nr. 8. Birkhäuser, Basel/Stuttgart, 1959.

ERM58 Gebrauchsanleitung für die ERMETH. Inst. f. angew. Math. ETH Zürich, 1958.

GOL87 Golub G.H., O'Leary D.P.: Some history of the conjugate gradient and Lanzcos algorithm: 1948-1976. (preprint)

HSt52 Hestenes M., Stiefel E.: Methods of conjugate gradients for solving linear systems. J. Res. Nat. Bureau Standards 49, 409-436 (1952).

Hoc55 Hochstrasser U.: Flatterrechnung mit Hilfe von programmgesteuerten Rechenmaschinen. Z. Angew. Math. Phys. 6, 300-315 (1955).

Kel71 Kellenberger W.: Ein konvergentes Iterationsverfahren zur Berechnung der Wurzeln eines Polynoms. Diss. ETH Nr. 4653, Zürich, 1971.

KSt Kustaanheimo P., Stiefel E.: Perturbation theory of Kepler motion based on spinor regularization. J. reine angew. Math. 218, 204-219 (1965).

Mae54 Maehly H.J.: Zur iterativen Auflösung algebraischer Gleichungen. Z. Angew. Math. Mech. 5, 260-263 (1954).

McP84 McPhee J.: Place de la Concorde Suisse. Farrar, Strauss & Giroux, New York, 1984.

RSS51 Rutishauser H., Speiser E., Stiefel E.: Programmgesteuerte digitale Rechengeräte (elektronische Rechenmaschinen). Mitt. Inst. f. angew. Math. ETH Zürich, Nr. 2, Birkhäuser, Basel/Stuttgart, 1951. Also in: Z. Angew. Math. Phys. 1, 277-297 (1950) (§§1-2), ibid. 1, 339-362 (1950) (§3), ibid. 2, 1-25 (1951) (§4), ibid. 2, 63-92 (1951) (§5).

Ru52 Rutishauser H.: Automatische Rechenplanfertigung bei programmgesteuerten Rechenmaschinen. Habilitationsschrift ETH Nr. 4. Mitt. Inst. f. angew. Math. ETH Zürich, Nr. 3, Birkhäuser, Basel/Stuttgart, 1952.

Ru52a Rutishauser H.: Ueber die Instabilität von Methoden zur Integration gewöhnlicher Differentialgleichungen. Z. Angew. Math. Phys. 3, 65-74 (1952). Engl. Transl.: National Advisory Committee for Aeronautics (NACA), Techn. Memo. 1403 (1956).

Ru53 Rutishauser H.: Beiträge zur Kenntnis des Biorthogonalisierungs-Algorithmus von Lanczos. Z. Angew. Math. Phys. 4, 35-56 (1953).

Ru57 Rutishauser H.: Der Quotienten-Differenzen-Algorithmus. Mitt. Inst. f. angew. Math. ETH Zürich, Nr. 7, Birkhäuser, Basel/Stuttgart, 1957. Contains revised versions of the papers in: Z. Angew. Math. Phys. 5, 233-251 (1954), ibid. 5, 496-508 (1954), ibid. 6, 387-401 (1955).

Ru60 Rutishauser H.: Ueber eine kubisch konvergente Variante der LR-Transformation. Z. Angew. Math. Mech. 40, 49-54 (1960).

Ru63 Rutishauser H.: On Jacobi rotation patterns. Proc. Symposia in Appl. Math. 15, 219-239 (1963).

Ru63a Rutishauser H.: Ausdehnung des Rombergschen Prinzips. Numer. Math. 5, 48-54 (1963).

Ru63b Rutishauser H.: Stabilie Sonderfälle des Quotienten-Differenzen-Algorithmus. Numer. Math. 5, 95-112 (1963).

Ru67 Rutishauser H.: Description of ALGOL 60 (Handbook for Automatic Computation, Vol. Ia). Springer, Berlin/Heidelberg/New York, 1967.

Ru68 Rutishauser H.: Once again: The least square problem. Linear Algebra Appl. 1, 479-488 (1968).

Ru69 Rutishauser H.: Zur Problematik der Nullstellenbestimmung bei Polynomen. In: Constructive Aspects of the Fundamental Theorem of Algebra, ed. by B. Dejon and P. Henrici. Wiley-Interscience, New York, 1969, 281-294.

Ru76 Rutishauser H.: Vorlesungen über numerische Mathematik (2 Vols.). Ed. by M. Gutknecht in cooperation with P. Henrici, P. Läuchli, and H.R. Schwarz. Birkhäuser, Basel/Stuttgart, 1976.

SaS68 Sauer R., Szabó I. (eds.): Mathematische Hilfsmittel des Ingenieurs, Teil III. Springer, Berlin/Heidelberg/New York, 1968.

Sca57 Schai A.: Die elektronischen und magnetischen Schaltungen der ERMETH. Scientia Electrica 3, 127-140 (1957).

Scl54 Schlaeppi H.: Entwicklung einer programmgesteuerten elektronischen Rechenmaschine am Institut für angewandte Mathematik der ETH. Z. Angew. Math. Phys. 5, 435-436 (1954).

Scw56 Schwarz H.R.: Ein Verfahren zur Stabilitätsfrage bei Matrizen-Eigenwertproblemen. Z. Angew. Math. Phys. 7, 473-500 (1956).

Scw81 Schwarz H.R.: The early years of computing in Switzerland. Annals Hist. Comput. 3, 121-132 (1981).

Sp50 Speiser A.P.: Entwurf eines elektronischen Rechengerätes unter besonderer Berücksichtigung der Erfordernis eines minimalen Materialaufwandes bei gegebener mathematischer Leistungsfähigkeit. Diss. ETH Nr. 1933, Zürich, 1950; Mitt. Inst. angew. Math., Nr. 1, Birkhäuser, Basel/Stuttgart, 1950.

Sp50a Speiser A.P.: Das programmgesteuerte Rechengerät an der Eidgenössischen Technischen Hochschule in Zürich. Neue Zürcher Zeitung, Nr. 1796 (50), 30. August 1950.

Sp51 Speiser A.P.: Ueber die Konstruktion von Rechengeräten mit linearen Potentiometern sowie die mathematischen Grundlagen der zugehörigen Kurvenanpassungen. Zürich, 1951 (unpublished, available at ETH Main Library). Shortened version: Rechengeräte mit linearen Potentiometern. Z. Angew. Math. Phys. 3, 449-459 (1952).

Sp54 Speiser A.P.: "ERMETH", Projekt einer elektronischen Rechenmaschine an der Eidgenössischen Technischen Hochschule in Zürich und bisherige Entwicklungsergebnisse. Neue Zürcher Zeitung, Nr. 1903 (79), 4. August 1954.

Sp54a Speiser A.P.: Projekt einer elektronischen Rechenmaschine an der E.T.H. (ERMETH). Z. Angew. Math. Mech. 34, 311-312 (1954).

Sp56 Speiser A.P.: Eingangs- und Ausgangsorgane, sowie Schaltpulte der ERMETH. Nachrichtentechn. Fachber. (NTF), No. 4, 87-89 (1956).

St49 Stiefel, E.: Bericht über eine Studienreise nach den Vereinigten Staaten von Amerika (18. Oktober 1948 - 12. März 1949). Zürich, 6. Mai, 1949 (unpublished).

St52 Stiefel E.: Bericht über ein Semester als Gastprofessor an der Universität von Californien und über die mathematischen Organisationen des "National Bureau of Standards" (22. Juli 1951 - 21. Februar 1952). Zürich, 26. April, 1952 (unpublished).

St52a Stiefel E.: Ueber einige Methoden der Relaxationsrechnung. Z. Angew. Math. Phys. 3, 1-33 (1952).

St58 Stiefel E.: Kernel polynomials in linear algebra and their numerical applications. Nat. Bureau Standards, Appl. Math. Ser. 49, 1-22 (1958).

St59 Stiefel E.: Ueber diskrete und lineare Tschebyscheff-Approximationen. Numer. Math. 1, 1-28 (1959).

St60 Stiefel E.: Note on Jordan elimination, linear programming and Chebyshev approximation. Numer. Math. 2, 1-17 (1960).

Sto54 Stock J.R.: An arithmetic unit for automatic digital computers. Z. Angew. Math. Phys. 5, 168-172 (1954).

Sto56 Stock J.R.: Die mathematischen Grundlagen für die Organisation der elektronischen Rechenmaschine der Eidgenössischen Technischen Hochschule. Mitt. Inst. Angew. Math. Nr. 6, Birkhäuser, Basel/Stuttgart, 1956.

Zus48 Zuse K.: Ueber den allgemeinen Plankalkül als Mittel zur Formulierung schematisch-kombinativer Aufgaben. Archiv Math. 1, 441-449 (1948/49).

Zus59 Zuse K.: Ueber den Plankalkül. Elektron. Rechenanl. 1, 68-71 (1959).

Zus70 Zuse K.: Der Computer - Mein Lebenswerk. Verlag moderne Industrie, München, 1970. 2nd ed.: Springer, Berlin/Heidelberg/New York, 1986.

CONJUGACY AND GRADIENTS

by

MAGNUS R. HESTENES

I have been invited to describe my experiences in the field of numerical analysis and to describe how these experiences influenced me in my studies of mathematics. In particular, I was invited to tell the story of the development of the conjugate gradient method for solving linear systems. I was one of the originators of this method.

At the invitation of the Mathematical Association of America, John Todd and I have written a short history of the Institute for Numerical Analysis, 1947-1954, located on the campus of UCLA. This Institute, called INA, was a Section of the National Applied Mathematical Laboratories, which formed the Applied Mathematics Division of the National Bureau of Standards, a part of the Department of Commerce. In this brief history we were concerned mainly with the mathematical aspects of this program. In particular, we were concerned about who participated in the project, what did they do, and what was their University affiliation. It is not my intention to repeat the material presented in this history except perhaps for some special items of interest.

As many of you know my specialty in mathematics is Variational Theory and Optimal Control Theory. My experiences in these fields have greatly influenced my approach to problems in numerical analysis. I shall describe certain aspects of Variational Theory, which are not only of interest in themselves but which led to a method of attack of certain computational problems.

I received my doctorate at the University of Chicago in 1932. After remaining at Chicago for a year, I left for Harvard as a National Research Fellow to work with Marston Morse. Inspired by the works of George D. Birkhoff, his mentor, Morse had become famous by his development of the Calculus of Variations in the large. Early in 1934, G.D. Birkhoff invited me to join him in writing a chapter in the Calculus of Variations. He wished to develop a new approach to the Calculus of Variations in the large. His idea was simple. It

came from the observation that every critical point of a function $F(x)$ satisfied constraints of the form

$$F'(x, h) = 0,$$

where h is held fast and x was allowed to vary. Here $F'(x, h)$ is the first variation of F, the differential of F. Unfortunately, in the general case, this procedure introduced too many singularities to be effective. However, it was very effective in the quadratic case. In quadratic case the condition $F'(x, h) = 0$ is a "conjugacy" condition although we did not use the term. As a result I wrote a long paper with Birkhoff on this subject developing these ideas for Calculus of Variations in the Small. Later, I wrote an extensive paper of the theory of "Quadratic Forms in Hilbert Space with Applications to the Calculus of Variations". In this paper the concept of conjugacy played a dominant role. I used the term "Q-orthogonality" instead of the term "conjugacy" in my writings. To see what conjugacy means in this context, may I remark that the extremals, the solutions of Euler-Lagrange equations, are the elements that are conjugate to the elements that vanish on the boundary. Thus, I was very familiar with the concept and use of conjugacy early in my career.

It is interesting to recall that, in 1936, I developed an algorithm for constructing a set of mutually conjugate directions in Euclidean Space for the purpose of studying quadric surfaces. I showed my results to Professor Graustein, a Geometer at Harvard University. His reaction was that it was too obvious to merit publication. This shows that Geometers were well versed in the concept of conjugacy. It suggests that perhaps hidden in the literature on geometry there is a method for finding the center of an ellipsoid which is equivalent to the method of conjugate gradients.

During the latter years of World War II, I was a member of the Applied Mathematics Group at Columbia University. Here I was concerned with the mathematical theory of aerial gunnery. We tested our theory with numerical computations. In one project, L.W. Cohen flew fighter planes on paper, duplicating with remarkable accuracy the results obtained by photography of actual paths of fighter planes, flying under certain gunnery

rules for attacking bombers. Cohen succeeded where others had failed. He succeeded because he wrote his algorithm in a manner so as to decrease errors which one encounters in computations.

When World War II ended, I returned to the University of Chicago. Shortly thereafter I accepted a Professorship at UCLA. Here I was approached by E. Paxson of the RAND Corporation to study the problem of steering a fighter plane so that it reached a prescribed position and direction in minimum time. This was a complicated variational problem involving differential constraints. Such problems had various names, such as, the Problem of Bolza, the Problem of Lagrange, or the Problem of Mayer. I found that the classical formulation of these problems did not fit this time optimal problem in a natural manner. Accordingly, I reformulated the variational problem so as to be more easily applicable to this minimum time problem. In doing so I had formulated a variational problem which is now known as an Optimal Control Problem. I translated the known results to fit this new formulation. The results were written up in 1949 as a RAND Report and were not published in a standard journal at that time. Later in 1965 I published a book entitled *Calculus of Variations and Optimal Control Theory*, which included the theoretical basis for this time optimal control problem. You might be interested to know that Pontryagin too was invited by his government to study the problem of aerial combat. This led to his formulation of Optimal Control Theory and Differential Games. His first necessary condition for an optimal control problem is now called *Pontryagin's Maximum Principle.* It is an extension of the standard conditions of Euler, Lagrange and Weierstrass. He established his results under weaker hypothesis than had been used heretofore. Thus, the study of the theory of aerial combat led to the development of modern theory of optimal control both here and in Russia.

Return to the time optimal problem proposed by Paxson. We obtained the equation of motion for our fighter plane and attempted to solve these equations numerically on a REAC. The REAC was an electrical analogue computer with about 3% accuracy. We tried to solve our problem as an initial value problem hoping to obtain the prescribed terminal conditions by a suitable choice of initial conditions. The results were disastrous. It turned

out that our equations were unstable in the forward direction. They were also unstable in the backward direction. However, by making many trials, we did obtain some notion of the nature of optimal paths. But this did not give us a sought after "Rule of Thumb" method for flying a plane in an optimal manner. Because of this experience I became convinced that we should look for an alternative approach to numerical solutions of variational problems of this type. In my considerations I restricted myself to simple variational problems. In particular, I chose to study the classical problem of finding surface of revolution of least area having prescribed circular boundary curves. The Euler equations to this problem normally has more than one solution satisfying prescribed boundary conditions. Only one of these solutions is minimizing. I tried two iterative methods, namely, Newton's Method and an Optimal Gradient Method. Our numerical experiments with these two methods were highly successful. In order to preserve these computations for future use, I wrote a second RAND Report in 1949 describing what we did. This report received a wide circulation among engineers and I received undo credit for devising these methods. Incidentally, with regard to the gradient method, I had to formulate an adequate definition of the concept of the gradient of an integral. To do so I introduced an inner product $\langle g, h \rangle$ on the space of variations. The gradient of $F(x)$ at a point x is a variation g such that

$$F'(x, h) = \langle g, h \rangle$$

for all admissible variations h. I found that the inner product usually used heretofore was unsatisfactory because elements of the form $x + ag$ were not admissible elements. However, I also found that there were a large class of inner products that were suitable for the problem at hand. These inner products need not be fixed but could vary with the element x with which were concerned. One such inner product is the inner product $F''(x; g, h)$ induced by the second variation of F. When this inner product is used, our gradient method becomes a version of Newton's method. Thus Newton's method can be viewed as a gradient method determined by a "preferred" inner product which varies at each step.

I also tried one other method, later called a penalty function method. In the simple case in which we minimize $f(x)$ subject to a constraint $g(x) = 0$ it proceeds as follows. Select a sequence $\{c_n\}$ converging to infinity. Obtain the minimizer x_n of the penalty function

$$F_n = f + c_n g^2.$$

Then, under favorable conditions, the sequence $\{x_n\}$ will converge to the minimizer x_0 of f subject to $g = 0$. Moreover, the sequence $\{2c_n g(x_n)\}$ converges to the Lagrange multiplier λ. In theory, this method is exellent and can be used effectively for theoretical purposes. Unfortunately, when I tried to solve a simple problem numerically by this method, I found that it had poor convergence properties due to round off errors and so I abandoned it for the time being. Besides, for variational problems with differential constraints, I knew that I would need to consider what is now known as relaxed controls and so would lead to a more complicated theory than I was willing to accept at that time. Later, at about 1969, I was invited to give a talk on computational procedures for solving optimization problems. It occurred to me at that time that a result in the folklore of Variational Theory could be used for this purpose. This result states that if x_0 minimizes $f(x)$ subject to $g(x) = 0$, then normally there is a multiplier λ and a constant c such that x_0 minimizes the function

$$F(x) = f(x) + \lambda g(x) + cg(x)^2$$

for all x near x_0 even when the constraint $g(x) = 0$ is not satisfied. Usually a relative small value of c is effective. Having chosen c, a suitable value of the multiplier λ can be found by an iterative procedure. The iteration that we shall use is obtained by observing that the gradient of F is given by the formula

$$F'(x) = f'(x) + [\lambda + 2cg(x)]g'(x).$$

This formula suggests the following iteration

Select an initial point x_1, an initial multiplier λ_1, and a suitable constant c. Having obtained x_1 and λ_1 find a minimizer x_{i+1} of the function

$$F_i(x) = f(x) + \lambda_i g(x) + c g(x)^2.$$

Then set

$$\lambda_{i+1} = \lambda_i + \alpha g(x_{i+1}) \qquad (\text{say } \alpha = 2c)$$

and repeat.

To obtain an initial estimate for x_1 and c, one can begin with the penalty function method. I called this method "a method of multipliers". This algorithm with some modifications proved to be an effective method for solving constrained minimum problems. An equivalent algorithm was also suggested by M.J.D. Powell at about the same time.

Return to the Summer of 1949. At that time I was invited to join the Institute for Numerical Analysis (INA) on a part time basis. In accepting this invitation I expected to pursue my studies of numerical methods in variational theory. However, I was diverted by Barkley Rosser who was the new director of INA. Rosser initiated a program of studying methods for solving linear equations and for finding eigenvalues and eigenvectors of matrices. He organized a seminar on this subject. The principal participants of this seminar were Barkley Rosser, George Forsythe, Cornelius Lanczos, Gertrude Blanch, Magnus Hestenes, William Karush, and Marvin Stein. Rosser and Forsythe specialized on finding solutions of linear equations. Forsythe, in particular, proceeded to classify known methods for such solutions. Hestenes, Karush, and Stein were chiefly responsible for the study of methods for finding eigenvalues and eigenvectors of matrices. Lanczos continued to refine his methods for solving eigenvalue problems. Blanch, who was in charge of numerical computations, acted as an advisor on numerical procedures. Of course, we did not limit ourselves to our specialties and participated actively on all the topics taken up in the seminar.

With regard to the study on solving linear equations, we specialized on iterative methods for solving linear equations. We did so in part because it appeared that they required less

high speed storage than other methods. Besides we found them to be interesting. We surveyed the known methods both from a theoretical point of view and from a numerical point of view. In preparing the short history of INA, which I wrote with John Todd, I found a manuscript written by Rosser and myself developing a unified theory for a large class of methods. I had forgotten that we had written this article. A summary of the contents of this article is given in the history of INA which Todd and I wrote. In this paper we discussed various algorithms for solving a linear equation

$$Ax = h$$

where A is a nonsingular $n \times n$-matrix and h was a prescribed n-dimensional vector. We used the size of the residual

$$r = h - Ax = A(x_0 - x)$$

as a measure of the closeness of x to the solution x_0 of our equation. To measure the size of r, we sometimes used the largest component of r. At other times, with * denoting transpose, we used a function of the form,

$$f(x) = \tfrac{1}{2} r^* K r = \tfrac{1}{2} x^* B x - k^* x + c,$$

where K is a positive definite symmetric matrix and

$$B = A^* K A, \quad h = A^* K h, \quad c = \tfrac{1}{2} h^* K h.$$

The solution x_0 of $Ax = h$ minimizes f and solves the equation $Bx = k$. When A is a positive definite symmetric matrix we can choose $K = A^{-1}$. Then $B = A$, $k = h$, and

$$f(x) = \tfrac{1}{2} x^* A x - h^* x + c$$

where c is an unknown constant which plays no role in our considerations. It should be noted that the minimizer x_0 of $f(x)$ is the center of the ellipsoid, $f(x) =$ constant. Thus,

the problem of solving $Ax = h$ is equivalent to that of finding the center of an ellipsoid. We observed further that the minimum point x_2 of $f(x)$ on a line

$$x = x_1 + tp$$

was the midpoint of the chord in which this line intersected the ellipsoid $f(x) = f(x_1)$.

Although it was not immediately obvious, we found that the algorithms that we studied were equivalent to one of the following type

(1) $$x_{i+1} = x_i - H_i(Ax_i - h) = x_i + H_i r_i$$

where r_i is the residual

$$r_i = h - Ax_i.$$

From this fact we concluded that, if

$$\mu = \limsup_{i \to \infty} \|I - H_i A\| < 1,$$

then the sequence $\{x_i\}$ converges linearly to the solution x_0 at the rate μ. In many cases the matrix H_i need not be constructed explicitly by the algorithm. For example, we can obtain x_{i+1} from x_i by a subroutine of the following type

> Choose m vectors $u_1, u_2, \ldots, u_m$ which span our space and selected vectors $v_1, v_2, \ldots, v_m$ such that $d_j = v_j^* A u_j$ is not zero for $j = 1, \ldots, m$. Select $y_1 = x_i$. Then, for $j = 1, \ldots, m$, set
>
> $$y_{j+1} = y_j + \alpha_j u_j, \quad \alpha_j = v_j^*(k - Ay_j)/d_j.$$
>
> Finally set $x_{i+1} = y_{m+1}$.

It can be shown that when x_{i+1} is obtained from x_i in this manner, then there is a matrix H_i such that equation (1) holds. In view of this result the Gauss Seidel method and a

large number of other standard methods can be studied simultaneously by considering an algorithm of the form (1). A discussion of our considerations of this nature can be found in the History of INA which Todd and I wrote. We omit these considerations here. However, I would like to remark that in most of the numerical cases we considered convergence was very slow. We were therefore on the lookout for more rapidly convergent algorithms. We also considered the introduction of a relaxation constant β in our algorithm but did not develop an adequate theory for this case.

One of the algorithms that we tried was a gradient method for minimizing the error function

$$f(x) = \tfrac{1}{2}x^*Ax - h^*x$$

for the case when A is positive definite. The negative gradient of f is the residual $r = h - Ax$. Accordingly, the gradient algorithm is of the form

$$x_{i+1} = x_i + a_i r_i, \quad a_i = |r_i|^2/r_i^* Ar_i,$$

where $r_i = h - Ax_i$ and $t = a_i$ is chosen to as to minimize $f(x_i + tr_i)$. We called this method, the optimal gradient method. Forsythe constructed a positive definite 6×6-matrix in a random fashion and proceeded to test the optimal gradient method numerically. He found that the method "bogged down" and that the solution could not be obtained using a reasonable number of steps. Accordingly he tried two different acceleration techniques. The first one used the relaxed equation

$$x_{i+1} = x_i + \beta a_i r_i,$$

where β is some number between 0 and 2. Values of β, such as 7, 8, and 1.2, were effective. Even $\beta = 0.2$ was better than $\beta = 1$. He also tried the following acceleration scheme suggested by Motzkin. When the algorithm bogged down he added an additional step of minimizing f along the line through x_{i-2} and x_i to obtain a new estimate x_{i+1}. This method was equally effective but somewhat more complicated to use. We discovered that

Aitken had used the second scheme earlier. Incidentally, this acceleration scheme yields one step of the conjugate gradient method described below.

Rosser returned to Cornell in the fall of 1950 and returned to INA for summer 1951 to pursue his studies of solutions of linear equations and to attend a Conference on "Solutions of Linear Equations and the Determination of Eigenvalues" to be held at INA in August 1951. In June or July 1951, after almost two years of studying algorithms for solving systems of linear equations, we finally "hit" upon a conjugate gradient method. I had the privilege of first formulating this new method. However, it was an outgrowth of my discussions with my colleagues at INA. In particular, my conversations with George Forsythe had a great influence on me. During the month of July 1951, I wrote an INA Report on this new development. When E. Stiefel arrived at INA in August to attend the conference on Solutions of Linear Equations, he was given a copy of my paper. Shortly thereafter he came to my office and said about the paper "this is my talk". It occurred that he too had invented the Conjugate Gradient Algorithm and had carried out successful experiments using this algorithm. Accordingly, I invited Stiefel to remain at UCLA and INA for one semester so that we could write an extensive paper on this subject. In the meantime C. Lanczos observed that the Conjugate Gradient Method could be derived from his algorithm for finding eigenvalues of matrices. In view of these remarks we see that there are three persons who are credited for inventing the Conjugate Gradient Method, namely, Stiefel, Hestenes, and Lanczos. However, as remarked above, this algorithm was an outgrowth of the program at INA on Solutions of Linear Equations originated by J.B. Rosser and participated upon by various members of INA, such as, G. Forsythe, W. Karush, T. Motzkin, L. Paige and M. Stein. Of these researchers, Forsythe was the most active in supplying numerical experiments for the algorithms discussed by the group. It was my privilege to invent the name "Conjugate Gradient Routine" for the new algorithm we had constructed.

The Conjugate Gradient Algorithm is based on the following property of ellipsoids:

The midpoints of parallel chords of an $(n-1)$-dimensional ellipsoid E_{n-1} lies on a $(n-1)$-plane π_{n-1} passing through the center x_0 of E_{n-1}. The $(n-1)$-plane π_{n-1} and the vectors in π_{n-1} are said to be conjugate to these chords.

Analytically, an ellipsoid E_{n-1} is the set of points x satisfying an equation of the form

$$f(x) = \tfrac{1}{2}x^*Ax - h^*x = \text{constant} \qquad (A^* = A > 0).$$

The minimizer x_0 of f is the center of E_{n-1} and solves the equation

$$Ax = h.$$

Parallel chords of E_{n-1} have a common direction vector p. A midpoint x of one of these chords minimizes f along this chord. It follows that the negative gradient

$$r = h - Ax = A(x_0 - x)$$

at such a midpoint x is orthogonal to p. That is,

$$p^*r = p^*(h - Ax) = p^*A(x_0 - x) = 0$$

or, equivalently,

$$p^*Ax = p^*h.$$

This equation represents an $(n-1)$-plane π_{n-1} through the center x_0 of E_{n-1}. Its normal is the vector Ap. Every vector q in π_{n-1} is orthogonal to Ap and is conjugate to p. The relation

$$p^*Aq = 0$$

therefore expresses the conjugacy of two vectors p and q.

Let us apply this result to the 2-dimensional case. We seek to find the center of an

ellipse. Referring to Figure 1 let x be a point on an ellipse E. Let p be a vector tangent

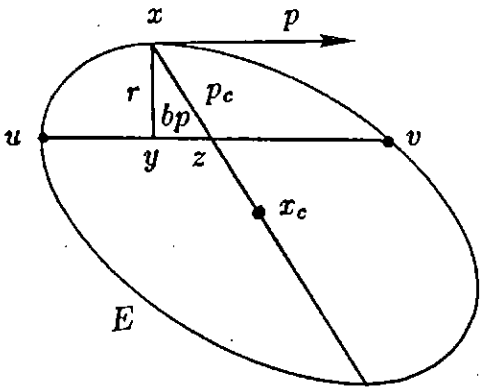


Figure 1

to E at x and let r be an inner normal of E at x. Through the tip y of r, draw a chord uv perpendicular to r. Let $z = y + bp = \frac{1}{2}(u + v)$ be the midpoint of this chord. Denote the vector joining x to z by p_c. Then $p_c = z - x = r + bp$. The vector p_c is conjugate to p. The midpoint x_c of the chord emanating from x in the direction p_c is the center of the ellipse E. The point x_c also minimizes the function

$$f(x) = \tfrac{1}{2}x^*Ax - h^*x \qquad (A^* = A > 0)$$

on this 2-dimensional space, where $f(x) =$ constant is an analytical representation of E. The geometric construction of the minimizer x_c of f can be carried out analytically as follows:

Choose a point x and compute $r = h - Ax$. Let p be a vector orthogonal to r. Compute

(2a) $$p_c = r + bp, \qquad b = -p^*Ar/p^*Ap$$

(2b) $$x_c = x + ap_c, \qquad a = p_c^*Ar/p_c^*Ap_c.$$

The point x_c minimizes $f(x)$ on our 2-plane.

This result leads us to the conjugate gradient routine. We shall give several versions of the conjugate gradient algorithm (cg-algorithm) for solving the equation

$$Ax = h,$$

where A is a positive definite symmetric matrix. The first of these is the formulation given independently by Stiefel and by Hestenes. It proceeds as follows.

Cg-algorithm I

Initial step. Select a point x_1 and compute

(3a) $$p_1 = r_1 = h - Ax_1$$

(3b) $$c_1 = p_1^* r_1, \quad d_1 = p_1^* A p_1, \quad a_1 = c_1/d_1$$

(3c) $$x_2 = x_1 + a_1 p_1, \quad r_2 = h - Ax_2 = r_1 - a_1 A p_1.$$

Iterative steps. Having obtained p_{i-1}, d_{i-1}, x_i, r_i compute

(3d) $$p_i = r_i + b_{i-1} p_{i-1} \quad \text{with} \quad b_{i-1} = p_{i-1}^* A r_i / d_{i-1}$$

(3e) $$c_i = p_i^* r_i, \quad d_i = p_i^* A p_i, \quad a_i = c_i/d_i$$

(3f) $$x_{i+1} = x_i + a_i p_i, \quad r_{i+1} = h - Ax_{i+1} = r_i - a_i A p_i.$$

Terminate when $r_{m+1} = 0$. Then $x_0 = x_{m+1}$ solves $Ax = h$.

In this algorithm the length of the vector p_i is not important. We can therefore, if we wish, introduce a scale factor σ_i for p_i. When this is done our formulas for these vectors take the form

(4) $$p_1 = \sigma_1 r_1, \quad p_{i+1} = \sigma_{i+1}(r_{i+1} + b_i p_i).$$

The scaling $\sigma_1 = 1$, $\sigma_{i+1} = (1 + b_i)^{-1}$ is particularly useful because then $p_{i+1} = h - Ay_{i+1}$ at a point y_{i+1} on the line segment joining x_i to x_{i+1}. Alternatively, we can use generalized gradients in which we have the formulas

$$p_1 = Hr_1, \quad p_{k=1} = Hr_{i=1} + b_i p_i, \quad b_i = -p_i^* Hr_{i+1}/d_i, \tag{5}$$

where H is a positive definite symmetric matrix. When these equations are used we call our algorithm a generalized *cg*-algorithm. A discussion of these and other variants of the *cg*-algorithm can be found in my book on *Conjugate Direction Methods in Optimization.*

Return to the *cg*-algorithm (3a)-(3f). Observe that equations (3d)-(3f) can be obtained from equations (2a) and (2c) by setting $x = x_i$, $r = r_i = h - Ax_i$, $p = p_{i-1}$, $p_c = p_i$, and $x_c = x_{i+1}$. It follows that the point x_{i+1} minimizes $f(x)$ on the 2-plane

$$x = x + \alpha r_i + \beta p_{i-1}.$$

This 2-plane is also determined by the points x_{i-1}, x_i, and $y_{i+1} = x_i + r_i$. The point x_{i+1} therefore minimizes $f(x)$ on this 2-plane and is the center of the ellipse in which this 2-plane cuts the ellipsoid $f(x) = f(x_i)$. Stiefel considered the direction p_i to be a relaxation of the direction r_i.

In view of this result we have the following alternative description of Cg-algorithm I.

Cg-algorithm II

Initial step. Choose x_1 and compute $r_1 = h - Ax_1$. Then find the minimizer x_2 of $f(x)$ on the line through x_1 and $y_2 = x_1 + r_1$.

Iterative steps. For $i = 2, 3, \ldots$, compute $r_i = h - Ax_i$ and find the minimum point x_{i+1} of $f(x)$ on the 2-plane through the points x_{i-1}, x_i, and $y_{i+1} = x_i + r_i$. The point x_{i+1} is the center of the ellipse in which the 2-plane cuts the ellipsoid defined by $f(x) = f(x_i)$.

Terminate when $r_{m+1} = 0$. Then x_{m+1} solves $Ax = h$.

When this algorithm is put in analytic form we obtain the following set of equations with x_1 as the initial point.

(6a) $$r_1 = h - Ax_1, \quad x_2 = x_1 + d_1 r_1, \quad r_2 = r_1 - \alpha_1 A r_1$$

(6b) $$x_{i+1} = (x_i + \alpha_i r_i - \beta_{i-1} x_{i-1})/(1 - \beta_{i-1}), \quad i > 1$$

(6c) $$r_{i+1} = (r_i - \alpha_i A r_i - \beta_{i-1} r_{i-1})/(1 - \beta_{i-1}), \quad i > 1$$

(6d) $$\alpha_i = |r_i|^2 / r_i^* A r_i, \quad \beta_{i-1} = r_{i-1}^* (r_i - \alpha_i A r_i)/|r_i|^2.$$

The scalars α_i and β_{i-1} are chosen so that r_{i+1} is orthogonal to r_i and r_{i-1}. With x_1 as the initial point, algorithms (3a)-(3f) and (6a)-(6d) generate the same points $x_2, x_3, \ldots$. Algorithm (6a)-(6d) can be found in the original paper by Hestenes. It is sometimes called GRADIENT PARTAN.

It can be shown that the point x_{i+1} minimizes $f(x)$ on the i-plane determined by the points $x_1, x_2, \ldots, x_i$, and $y_{i+1} = x_i + r_i$. This i-plane can be represented parametrically by the equation

$$x = x_1 + \gamma_1(x_2 - x_1) + \cdots + \gamma_{i-1}(x_i - x_{i-1}) + \gamma_i r_i.$$

It can be shown that, for the minimizer x_{i+1}, we have $\gamma_1 = \gamma_2 = \cdots = \gamma_{i-2} = 1$. It follows that x_{i+1} lies in the 2-plane

$$x = x_{i-1} + \gamma_{i-1}(x_i - x_{i-1}) + \gamma_i r_i$$

and so minimizes $f(x)$ on this 2-plane. In view of this result, Cg-algorithm II is equivalent to the following

Cg-algorithm III

Initial step. Select x_1 and compute $r_1 = h - Ax_1$. Find the minimizer x_2 of $f(x)$ on the line through x_1 and $x_1 + r_1$.

Iterative steps. For $i = 2, 3, \ldots$, find the minimizer x_{i+1} of $f(x)$ on the i-plane determined by the points $x_1, x_2, \ldots, x_i$, $y_{i+1} = x_i + r_i$, where $r_i = h - Ax_i$.

Terminate when $r_{m+1} = 0$. The point x_{m+1} solves $Ax = h$.

The *cg*-algorithm can also be interpreted geometrically as described in the following

Cg-algorithm IV

We seek to find the center of the $(n-1)$-dimensional ellipsoid E_{n-1} defined by the equation $f(x) = f(x_1)$. The point x_1 is on E_{n-1}. Let C_1 be a chord of E_{n-1} emanating from x_1 in the direction of the inner normal of E_{n-1} at x_1. Find the midpoint x_2 of C_1. The $(n-1)$-plane π_{n-1} through x_2 conjugate to C_1 contains the center x_0 of E_{n-1}. If $x_2 - x_0$ we are done. Otherwise π_{n-1} intersects the ellipsoid $f(x) = f(x_2)$ in a $(n-2)$-dimensional ellipsoid E_{n-2} whose center is also x_0. Accordingly, we have reduced the dimension of our space of search by 1. We now repeat the process and select a chord C_2 of E_{n-2} emanating from x_2 in the direction of the inner normal of E_{n-2} at x_2. Find the midpoint x_3 of C_2. The $(n-2)$-plane π_{n-2} in π_{n-1} conjugate to C_2 contains the common center x_0 of E_{n-2} and E_{n-1}. If $x_3 = x_0$ we are done. Otherwise, π_{n-2} intersects the ellipsoid $f(x) = f(x_3)$ in a $(n-3)$-dimensional ellipsoid E_{n-3} whose center is also x_0. Again we have reduced the dimension of our space of search by 1. Proceeding in this manner we finally obtain a chord C_m of an $(n-m)$-dimensional ellipsoid E_{n-m} whose midpoint is x_0 thereby completing our search for the center of E_{n-1}.

The following analytic version of *Cg*-method IV led to the name *Conjugate gradient algorithm.*

Cg-algorithm V

Starting with a point x_1 find the direction p_1 of steepest descent of $f(x)$ at x_1. Proceed in the direction p_1 to the point x_2 at which $f(x)$ has a minimum value. Let π_{n-1} be the $(n-1)$-plane through x_2 conjugate to p_1. Find the direction p_2 of steepest descent at x_2 of $f(x)$ in π_{n-1}. Proceed from x_2 in the direction p_2 until a point x_3 is reached at which $f(x)$ has a minimum. Let π_{n-2} be the $(n-2)$-plane in π_{n-1} conjugate to p_2 (and hence

also conjugate to p_1). Find the direction of steepest descent p_3 at x_3 of $f(x)$ in π_{n-2} and proceed to the minimum point x_4 of $f(x)$ in this direction. Proceeding in this manner we finally reach the minimum point x_0 of $f(x)$ in our original n-space. It is the solution of $Ax = h$.

We call $p_1, p_2, \ldots$ and their multiples "conjugate gradients" of $f(x)$. Except possibly for a positive scale factor, they are the vectors $p_1, p_2, \ldots$ generated by Cg-algorithm I.

There is another version of the cg-algorithm which is of interest. In this algorithm we alternate minimizations of the functions

$$f(x) = \tfrac{1}{2}x^*ax - h^*x, \quad g(x) = \tfrac{1}{2}|r|^2 = \tfrac{1}{2}|h - Ax|^2.$$

It proceeds in the manner described in the following

Cg-algorithm VI

Select a point x_1. Set $y_1 = x_1$ and compute $p_1 = -f'(y_1)$. Having obtained x_i, y_i, and $p_i = -f'(y_i)$, find the minimum point x_{i+1} of $f(x)$ on the line $x = x_i + tp_i$. Next determine the minimum point y_{i+1} of $g(x)$ on the line joining y_i to x_{i+1}. Compute $p_{i+1} = -f'(y_{i+1})$. Terminate when $x_{m+1} = y_{m+1}$ or equivalently when $f'(y_{m+1}) = 0$. The point $x_{m+1} = y_{m+1}$ is the minimum point of $f(x)$ and solves the equation $Ax = h$.

It is also of interest to note that the conjugate gradient algorithm can be put in the form (1) with H_i replaced by a_iH_i. We then have the iteration

$$x_{i+1} = x_i + a_iH_ir_i, \quad r_i = h - Ax_i$$

where H_i is a positive definite symmetric matrix. We adjoin to this an updating procedure for the matrix H_i. It has the property that $H_{n+1} = A^{-1}$. This form of the conjugate gradient algorithm is due to Davidon, who fashioned it so as to be applicable to nonlinear equations. It was modified later by Fletcher and Powell. It is now called the Davidon-Fletcher-Powell method or the variable metric method. There are several versions of this algorithm. The one that we shall present is the following

Cg-algorithm VII

Let H be a positive definite matrix. Set $H_1 = H$ and perform the following iteration with x_1 as the initial point and $r_1 = h - Ax_1$.

(7a) $$p_i = H_i r_i, \quad s_i = Ap_i, \quad q_i = H_i s_i, \quad d_i = p_i^* s_i,$$

(7b) $$\delta_i = q_i^* s_i, \quad e_i = \delta_i / d_i, \quad c_i = p_i^* r_i, \quad a_i = c_i / d_i$$

(7c) $$x_{i+1} = x_i + a_i p_i, \quad r_{r+1} = r_i - a_i s_i = h - Ax_{i+1}$$

(7d) $$H_{i+1} = H_i - (p_i q_i^* + q_i p_i^*)/d_i + (e_i + 1) p_i p_i^* / d_i.$$

Terminate when $r_{m+1} = 0$. Then x_{m+1} solves $Ax = h$. If $m = n$, we have $H_{n+1} = A^{-1}$.

Under perfect computations we have the relations

$$p_1 = Hr_1, \quad p_{i+1} = Hr_{i+1} + b_i p_i$$

so that Algorithm (7a)-(7d) is equivalent to the generalized cg-algorithm and is equivalent to Algorithm (3a)-(3f) when $H = I$. It involves more computations than the original algorithm. However, it is within it a built-in correction of roundoff errors and so usually gives better results than the original cg-algorithm when the matrix A is ill-conditioned. Extensions of this algorithm have been useful in the minimization of nonquadratic functions. There are many variations of the updating formula (7d) for H_i. For example, one can add nonnegative multiples of the matrix

$$(e_i p_i - q_i)(e_i p_i - q_i)^*$$

to H_{i+1} with altering its basic properties.

We have given seven versions of the cg-algorithm. Additional versions can be found in my book on *Conjugate Direction Methods in Optimization.* One of the first five versions given above was the original version of cg-algorithm developed at INA. I believe that it

was either Cg-algorithm IV or Cg-algorithm V but I am not certain about this. It could have been Cg-algorithm III or II because, at that time, Forsythe and I were experimenting with algorithms for minimizing $f(x)$ on i-planes for $i = 2, 3, \ldots$.

In the application of the cg-algorithm, it is often desirable to precondition the matrix A before applying the cg-algorithm. Also the cg-algorithm is sometimes used in conjunction with other algorithms for solving linear equations.

Cg-algorithm I has within it an algorithm for computing the characteristic polynomial of A. One needs only replace A by λ. This algorithm is equivalent to one developed earlier by C. Lanczos. It follows that the algorithm of Lanczos for finding eigenvalues implicitly contains the cg-algorithm although none of us recognized this fact in the seminar we conducted. When Lanczos became aware of this feature of his algorithm, he formulated an alternative version of the cg-algorithm which he called a "Method of Minimized Iterations". The connections between his algorithm and the original cg-algorithm can be found in the historical account of INA which I wrote with J. Todd.

The cg-algorithm has some useful properties. At each step the value of the error function $f(x)$ is diminished. So also is the distance of our estimate x_i from the solution x_0. This latter property may fail when generalized gradients are used. If A has multiple eigenvalues, the algorithm will terminate in less than n steps. It follows that if A has clustered eigenvalues, a good estimate of the solution is obtained early. A discussion of these and other properties of the cg-algorithm can be found in the original paper by Stiefel and Hestenes and in my book entitled *Conjugate Direction Methods in Optimization* published by Springer in 1980. We also discussed the problem of finding least square solutions for a general equation $Ax = h$ in which A may be nonsymmetric and singular. There is a vast literature on cg-algorithms and Lanczos' algorithms. References can be found in my book and in a recent paper by Gene Golub and Dianne O'Learly entitled *Some History of the Conjugate Gradient and Lanczos Algorithms* 1948-1976. This excellent paper has been submitted to the SIAM Review.

As I remarked earlier, in our seminar I was responsible for studies of methods for obtaining eigenvalues of a matrix A. We developed a gradient method for finding the eigenvalues of a symmetric matrix. It turned out that this method could be viewed as a generalization of the power method. Of course, we studied the power method and the inverse power method. We also considered the Jacobi method but did not have the computing facilities for a serious study of this method numerically. In addition we considered the problem of finding singular values of matrices. Our studies complimented the studies of Lanczos for finding eigenvalues of matrices.

B I T - a child of the computer.

Carl-Erik Fröberg

Department of Computer Science, Lund University,
P.O. Box 118, S-221 00, Lund, Sweden.

Abstract. The back-ground of the Scandinavian computer journal B I T will be outlined, in particular with respect to computational demands in science, technology, industry and defence. The history of B I T will be described and related to the evolution of computers, numerical mathematics and computer science. Some contributed papers which have had an impact on the general development will be discussed briefly.

The 19th century could perhaps be characterized as a period of preparation for the advent of the computer. It so happened that quite a few Swedish inventors played a role in this development. Scheutz, father and son, as well as Wiberg constructed mechanical devices for a somewhat automatized calculation for solving simple arithmetic problems by series of pre-determined operations. In fact, Wiberg was able to compute a logarithm table which even appeared in print. Later, Odhner built a robust mechanical, hand-driven calculator which around 1930 was followed by electromechanical calculators. All lengthy calculations had to be performed manually by this time. Let me mention a few examples from Sweden.

One such problem was to find periods of so-called internal waves in the sea. These waves are huge, up to 20-30 meters in size, but nevertheless invisible. They are generated by the moon and observed as sharp changes in the salinity. The method used was numerical autoanalysis, that is a kind of Fourier analysis of the function by itself.

During the war there was a great need for ballistic tables, and I belonged to a group involved in computing bomb tables for the Swedish Air Force. We used the classical Runga-Kutta method with air resistance represented graphically, and we had only electro-mechanical calculators at our disposal. After having computed a basic set of orbits we could produce the wanted tables by a suitable interpolation procedure. It is a sad fact that all our tables could probably have been computed in a couple of minutes on a fast modern computer. After the war I was involved in some rather lengthy computations on the deuteron concerning energy levels and quadrupole moment and also in problems on scattering.

However, in 1946 some people in the Swedish Navy and in the Academy for Engineering Sciences got interested in the progress in the United States and after having visited the key projects they reported back with great enthusiasm. It was quickly decided to offer scholarships to four young students; they were selected in the spring of 1947. They arrived already in August or September; two of them went to Harvard and MIT while two, including myself went to Princeton. As far as I am concerned I enjoyed a phantastic hospitality, particularly from Herman Goldstine with whom I established a lifelong friendship. Back home in 1948 some of us got involved in the

construction of a relay computer (BARK), completed in 1950. However, it was soon understood that there was a need for more computer facilities, and the construction of BESK under Erik Stemme was initiated. It was completed in 1953, and during a short period of time it was considered as one of the most powerful computers in the world. Clearly its structure was very much the same as that of the Princeton computer.

In 1956 a simplified copy of BESK called SMIL was completed at Lund University, built with a minimum budget of some 20,000 $. This computer was used for a large variety of problems in nuclear physics (particularly eigenvalue problems), spectroscopy, mathematics (number theory, construction of tables), and also social sciences (geographical simulations). Several problems coming from industry and different research organisations were also treated.

The interest in and use of computers created a very natural demand for conferences since the literature on the subject for obvious reasons was very scarce by this time. The first Scandinavian conference on computers was held May 1959 in Karlskrona, later known as the place where a Soviet submarine ran aground in 1981. One reason for this choice of site was the fact that the Swedish Navy played an important role in initiating the computer development, another that, especially in spring, it was a lovely place, situated on the Baltic. Preliminary discussions were held informally on the need for a Nordic journal on computer related problems, and at the next conference in Copenhagen in 1960 a more formal meeting was arranged. Niels Ivar Bech acted as chairman, and further Peter Naur of Denmark, Jan Garwick, Norway, Olli Lokki, Finland, and myself from Sweden were present. It was unanimously decided to start a Nordic journal within the computer area, to appear quarterly. The journal was intended to be international with papers published in English, German, or the Scandinavian languages. As it turned out, only about 4-5 papers have been published in German, and very soon papers in the Scandinavian languages gradually disappeared. Nowadays it is required that all papers be written in English.

The name of the journal was a longe one: Nordisk Tidskrift for Informationsbehandling, but playing around with the initials in a clever way we were able to form the name B I T. In fact, this name is most convenient because of its shortness which makes it very easy to quote papers printed in the journal. As is well known it is somewhat dangerous to suggest an activity including work since there is a great risk that the proposer is elected to carry through the project. This is exactly what happened in this case, and from the very beginning up to this time I have served as Editor of B I T. Naturally, we have also an Editorial Board with members from the Nordic countries. Peter Naur of Copenhagen has been a member right from the beginning in 1961 and Germund Dahlquist from 1962. We got financial support from the Danish company Regnecentralen under Niels Ivar Bech and from several official sources including the Nordic Research Organisations for Natural Sciences. Finally, just a few years ago we managed to become self-supporting, perhaps mostly through favorable exchange rates.

During the first decade B I T tried to learn the public to get acquainted with new developments within the computer area. It is natural that the growing crowds of people working with computer applications of different kinds felt an increasing difficulty in keeping up with the fast progress, both in hardware and in software. That left a gap which B I T tried to fill. Simultaneously we also tried to accomodate scientific papers, particularly in numerical analysis and in computer languages. Very early we opened a special column for algorithms written in ALGOL 60. As a consequence of this policy our subscribers to a large extent were private Scandinavians during the first decade. Then the situation changed slowly. The need for general information decreased because this was treated in special new publications of type Datamation and also in ordinary and weekly newspapers. Simultaneously the number of scientific contributions to B I T increased strongly, first in numerical mathematics, later also in compter science. As a result of this development the

number of Scandinavian subscribers decreased while the number of non-Scandinavian subscribers, mostly libraries of research organisations and universities increased, the net result being slightly positive. From 1980 it was clearly indicated that B I T was divided in two sections, one for Computer Science, and one for Numerical Mathematics. In spite of obvious difficulties we have been able to strike a reasonable balance between these two.

The first volume (1961) had 290 pages and was type-written and photographed. Already volume 2 was printed in an ordinary way. B I T had obviously been observed also abroad since two contributions, one from the US (Louis Fein) and one from the Netherlands (Peter Wynn) appeared already in the first volume, while several "foreign" papers (among them one by Gene Golub) were presented in volume 2. From the beginning there was a certain ambivalence with respect to papers on hardware: during the first 10 years we published a few of that kind, but finally they disappeared.

Turning to computer science there is an important subject which attracted considerable attention during the first 10-15 years, namely computer languages and compiler construction. The main interest was centered on ALGOL 60 since by that time FORTRAN was only available for users while the corresponding compilers were held secret. However, different aspects on other programming languages, e.g. COBOL, ALGOL 68, PASCAL and SIMULA, have also been treated.

It is of course hard to tell which papers have had an impact on the general development, but I think that papers by Dahlquist and others on stability problems, Enright-Hull-Lindberg on numerical methods for stiff systems and a whole bunch on Runge-Kutta methods have had a considerable influence. Finally I think it is fair to mention that we offered a special issue dedicated to Germund Dahlquist on his 60th birthday, followed by one dedicated to B I T on its 25th birthday, both with about 300 pages.

Concerning the geographical distribution of authors and subscribers we can say roughly that the Nordic countries, the rest of Europe, and USA plus Canada account for about 1/3 each in both respects. The most striking feature is the steep increase in offered contributions from Taiwan, and we have also had quite a few from mainland China. In both cases the quality has been rather good. Also some exotic countries are represented by authors: Nigeria, Singapore, Ecuador, Sudan and the Fiji Islands, just to mention a few. Even if some papers must be rejected we try to encourage the authors, and in many cases the papers can be published after a more or less thorough revision. As a mean value the time between reception of a paper and publication is nine months.

COMMENTS ON THE DEVELOPMENT OF COMPUTATIONAL MATHEMATICS IN CZECHOSLOVAKIA AND IN THE USSR

I. Babuška
Institute for Physical Science and Technology, University of Maryland

At the request of the organizing committee, I would like to share some of my observations and remembrances about the development of computational mathematics in Czechoslovakia and the USSR. My observations will be very subjective and broad in scope.

A. The Development in Czechoslovakia

1. The development until 1918

A very essential milestone in the development of science in Central Europe was the foundation of the Charles University in Prague in 1348. To my knowledge, the first mathematical text at this University was likely Algorismus Prosaycus by Krištan, from Prachatice (in Czechoslovakia) written in 1400. This text concentrates on arithmetic and so I see it as the first text on computational mathematics in Central Europe.

Many outstanding mathematicians interested in computations were, directly or indirectly, for shorter or longer periods, associated with the Charles University. Let me mention the astronomers T. Brahe (1546-1601), J. Kepler (1571-1630) and J. Bürgi (1552-1632), among others. The silver mining in Bohemia (the major mining place in Europe at this time) and the construction of a system of ponds in Southern Bohemia required significant effort and high accuracy in geodesic measurements and computations. This, together with the need of astronomy, contributed to the development of computational mathematics. Computational methods of Brahe (how to multiply numbers by additions with help of tables of sin and cos) together with the logarithmic (tables of Napier, Kepler, Bürgi), and the development of a mechanical computer by Schickart, from Tübingen in Germany (1592-1632), based at Kepler's inspiration, lead to new developments of computational mathematics. The Algebra by Bürgi was edited by Kepler, especially because it contributed to the computational techniques. Many other important developments happened in Prague, especially in connection with the University; nevertheless, I will not go into details except to emphasize that this progress was very closely related to the development of applied mathematics.

2. The period 1918-1945

After World War I, Czechoslovakia was established as a democratic republic. The development of computational mathematics was closely related to applications especially in engineering. Let me mention as an example the fields with which I am familiar, the structural mechanics, elasticity, strength of material. One outstanding scientist in this direction was Z. Bažant, professor of the Technical University in Prague. Traditionally, computational methods for the analysis of frame constructions were of great interest. Essentially, these techniques were related to the direct and iterative methods for solving systems of linear algebraic equations. These usually sophisticated methods were based rather on physical and engineering intuition than on mathematical theories, because at this time maximal simplification was needed for any computation. Some of these methods could be described today as the splitting method, block iterations, some as the method of dimensional reduction, etc.

Approximate methods for analysis of plates and shells based, for example, on Fourier method, series method, etc., were typical for solving partial differential equations. Various solution methods had the character of finite differences derived on physical grounds by "spring analyses." Let us mention that Cauchy's spring model of an elastic medium can be interpreted as finite difference scheme for Lamé-Navier equations with Poisson ratio $\nu = 1/3$. Various methods for solving nonlinear problems, eigenvalue problems, etc., were developed in connection with buckling and stability considerations in general. In mechanical engineering various methods were developed in connection with vibration problems, etc.

The first mathematical book [1, 1934] written in 1934 by two professors of mathematics at the Technical University in Prague became a widely used text. This book covered essentials of numerical analysis in a relatively accurate and detailed manner. Although this book did not brake new grounds or introduced new approaches, it became a major source of education in computational mathematics and in computational research in en-

gineering applications in Czechoslovakia.

Czechoslovakia was a highly developed industrial country. The Škoda enterprises, an industrial concern, supported a theoretical department which was heavily involved in computations. Thanks to that, Czechoslovakia had a broad and firm tradition in applied mathematics and through it in computational methods.

It is interesting to compare the scientific situation in Czechoslovakia and Poland. Without any doubts, Poland was a superpower in pure mathematics during this time; it was in the absolute forefront of the world research in developing such mathematical fields as Functional Analysis, Real Analysis, Topology, etc. On the other hand, in my opinion, the level of applied mathematics was higher in Czechoslovakia than in Poland.

In the Fall of 1938, Czechoslovakia was crippled by the Munich treaty; on March 15, 1939, Hitler occupied Bohemia and Moravia, the industrial western part of Czechoslovakia, and created a puppet state of Slovakia from the eastern part of Czechoslovakia. In other words, Czechoslovakia ceased to exist.

On November 17, 1939, Hitler closed all universities to prevent the higher education of the Czech population. Universities were closed until the end of the war and the collapse of Hitler's Germany. This, of course, had a profound effect in the development of science in general, and mathematics in particular. Although there were underground seminars and some mathematical work and some more elementary publications were somehow published, an entire generation of scientists (6-8 years period) was lost. (Some effects of this will be discussed in the following sections.)

3. The early post war period. Period of basic education

Almost immediately after the end of the war, the Universities were opened and maximal efforts started to fill the gap created by the closing of the schools for six years. Shortened studies were designed to fill the gap as quickly as possible. Basic lectures were given in theaters for 1500-2000 students. This emergency education had surprisingly good effects because of the high motivation of the students and teachers. In three to four years the major part of the educational gap was closed, especially in the education of engineers, teachers, medical personnel, etc., but could not and was not completed in the field of science and in the education of scientists.

In February 1948, the Communist party took over the government. The pattern of Soviet organization was applied in Czechoslovakia including scientific education and research. Already in 1949 the institution of "Aspirants" was established. "Aspirantura" was an organization for graduate studies in and outside the universities. Aspirants were awarded fellowships. Almost at the same time, preparations for the foundation of the Academy of Sciences (Soviet style) was started.

In mathematics the major responsibility for the education of aspirants was given to E. Čech, professor at Charles University, a well known topologist. He gathered about a dozen of the best and most promising young students, graduates from the universities, and led their scientific education. Let me mention a few names from this group which became well known in mathematics in and outside of Czechoslovakia. I. Babuška (Numerical and Applied Mathematics), M. Fiedler (Theory of Matrices), J. Kurzweil (Theory of Ordinary Differential Equations), V. Pták (Functional Analysis), O. Vejvoda (Differential Equations), M. Zlamal (Finite Element Method). Under the leadership of E. Čech, the best Czechoslovak mathematicians participated in this program. I would like to mention especially V. Knichal, V. Kořinek, professors at Charles University in Prague, F. Vyčichlo, Professor at Technical University, O. Boruvka, Professor at the University in Brno. This group of students and their teachers were a congenial, dedicated group of the highest quality. I have not seen afterwards anywhere in the world such a congenial group of students and teachers.

Professor E. Čech, although a pure mathematician with basic interest in topology and geometry, had very broad views which he imposed on the group together with his dedication, hard work and interest in every aspirant (student). E. Čech insisted that all of his "aspirants" became familiar with numerical methods. To this end, he obtained from the Soviet Union some old copies of the book of Kantorovich Krylov [2, 1936], which was well known in the Soviet Union and was translated later in the West. Because the copying machine did not exist at that time in Czechoslovakia with the exception of the ditto sheet machine, E. Čech translated and dictated it to his secretary, so that the entire book was typed and by ditto technology given to his aspirants. This and similar Čech's acts were typical of his dedication. Nevertheless, it is necessary to say that Prof. E. Čech was a highly demanding person, completely "obsessed" by mathematics (in the best sense of the word) who permanently challenged his students individually and as a group almost in a dictatorial fashion. In retrospect, one has to admire more and more his mathematics, dedication, wisdom and what he gave to "his" youngsters (with or without their consent).

E. Čech also insisted that the aspirants will get basic education in computer technology and its use. He arranged for lectures by Prof. A. Svoboda. A. Svoboda worked in the field of electronics in the United States during World War II. He returned to Czechoslovakia in 1946 and went back to the United States in 1966. A. Svoboda was the leader in the development of computers in Czechoslovakia. Under his leadership, a design and implementation of a unique relay computer was made (tubes were not available at this time). Svoboda's machine called SAPO was a triplet machine with three arithmetic units which after every operation (made simultaneously) "voted" and the majority vote was used as the answer. The programming was a 5 address system. The computer SAPO had many unique features. Unfortunately it was completed when the next generation (tubes) was already in full swing.

During this period, work seminars were routine. Teachers, as well as students, were involved in these seminars. I remember, for example, the work in a paper by Goldstine, Neumann [3, 1947] which convinced us that there was no hope that eliminatfon method could and would be used in the future for matrices larger than 100 (what a wrong conclusion!) Another paper having big impact was the one by Courant, Friedrich and Levy [4, 1927] which was analyzed in every detail; E.

Čech and others made many comments related to the connection to other fields of mathematics.

E. Čech, V. Kořinek, V. Knichal and F. Vycichlo were able to grow a new generation of very active mathematicians and fill the gap of the closing of universities by Hitler in a relatively short period of time. (Let us mention that also with an extraordinary effort, it needed eight to ten years to overcome the basic effects of this closing.)

4. Building the Mathematical Institute of the Czechoslovak Academy of Science

In the early fifties, the Mathematical Institute of the Czechoslovak Academy was established. E. Čech, V. Knichal, J. Novák, F. Vyčichlo, together with some of the previous aspirants, played a prominent role in leading the Institute. New research groups were built and another generation of young researchers educated.

In the field of Applied and Numerical Mathematics and Partial Differential Equations, I. Babuška and K. Rektorys* became very active in collaboration with Prof. F. Vycichlo.

The main emphasis in this direction of applied mathematics was mechanics of solids and partial differential equations, especially of elliptic type. The main direction was the relation between modern exact mathematics and applications with emphasis on constructive approaches which could be used for concrete solution of problems. One of the result of this effort was the book [6, 1953]. The basis of this book was the theory of analytic functions of complex variables in the spirit of the Muschelishvili theory. This philosophy of the honest mathematics in application later led to the book [5, 1966] by K. Rektorys and coworkers.

The above philosophy in its purest form, and influenced by Bourbaki, led to some effort (e.g. by V. Knichal and others) to create an axiomatic-precise system of applied mathematics. This effort did not accomplish too much.

The early post-war period (I call it 'period of education') ended roughly in 1954 when the Mathematical Institute was firmly established.

5. The Project Orlik

The project Orlik was an important milestone in the development of computational and applied mathematics in Czechoslovakia. This project was mentioned as the one of the principal achievements of the Czechoslovak Academy of Sciences on the occasion of the celebration of 30 years of its foundation, and in the publication [8, 1986] on the occasion of forty years of post-war mathematics.

The project Orlik was a large scale computational project (although still performed on desk calculators) which could be characterized as the transition from the precomputer to the computer era in Czechoslovakia, see, e.g. [9, 1986]. This project had a profound impact and was characterized by the principles which after thirty years are still the center of interest in computational and applied mathematics in the United States and elsewhere.

The research project Orlik was related to the proposed building of the largest dam in Czechoslovakia located about 40 m south from Prague on the river Vltava. The dam was of concrete gravitational type, about 400 ft high. The project Orlik was an integrated complex research in mathematics, engineering and material science (cement, concrete). The leader of the mathematical part was I. Babuška, of the engineering part Prof. L. Mejzlík (Professor of Tech. University Brno), and of the technological part, Dr. J. Jirsák. The project was a team work and included a large staff of people working on desk calculators.

The main technical problem was that the concrete releases a significant amount of heat during hardening. Simultaneously, the hardening, which depends strongly on the temperature, changes significantly the material properties, e.g. elasticity module, creap and relaxation properties, etc. This leads to the creation of significant stress state which is "frozen in" during the hardening and later could lead to dangerous and serious cracks. The effects of this type could be controlled by a proper technology of building and of material properties. The large dams in the United States used a cooling system by pipes inserted in the dams. The basic questions of the research were: a) What are the effects of various building procedures on the possible cracks? Is it necessary to use pipe cooling, etc.? Could the cracks, if any, be expected? b) How the properties of the concrete influence the undesired effects of building, later functions of the dam, etc.? Based on the research results, the dam was built without cooling by a relatively quick building schedule in blocks about 12 ft high. The dam behaved as predicted and serves well its purpose. Results of the analysis were presented at the dam world congress in 1958, and were included distinguishly in the congress reporter's report. Some technical conclusions are, e.g. in [10, 1958], [11, 1961].

The essential novelty was the emphasis on integrated approach and the reliability of the conclusions. The reliability aspects were divided in the following groups:

a. reliability of mathematical model,
b. reliability of available input data,
c. reliability of the numerical method and principles of its selections,
d. reliability of the arithmetic computations (round offs) (because minimal number of used digits were essential for computations on desk calculators).

These questions were directly and indirectly addressed in a series of theoretical and engineering papers and reports.

The problem was highly nonlinear and three-dimensional. Because three-dimensional solution was out of the question for obvious reasons, a series of two-dimensional problems were solved and combined approximately into three-dimensional ones by a sort of splitting up approach. Let me explain now some of the problems (in a simplified way).

*K. Rektorys is the author of [5] and [7], which are well known in the United States.

1) Thermoproblem with and without cooling. The basic equation which was considered was:

$$(1a)\quad c(u,\delta)\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}a(u,\delta)\frac{\partial u}{\partial x} + \frac{\partial}{\partial y}a(u,\delta)\frac{\partial u}{\partial y} + F(u,\delta)$$

$$(1b)\quad \frac{d\delta}{dt} = G(u,\delta).$$

Here u is the temperature, F the intensity of the heat created by hydration, δ a fictive time (age) in which the same amount of heat was produced as when the temperature would be fixed (about 70°F). This fictive time characterizes the state of the chemical reaction. The coefficients $c(u,\delta)$ and $a(u,\rho)$ were found so mildly dependent on u,δ that average values were used. The characterization of $F(u,\delta)$ was essential. A special care was devoted to the laboratory experiments. Finally, the above mentioned model, based on a chemical model of hydration, was accepted and a differential equation (1a,b) was designed and used. The data were obtained by the measurement of the heat release in the period $(0,t)$ under constant temperatures and in an adiabatic state. The computation of the increments in F was organized so that the total heat was exactly preserved. This was very essential for the reliability.

The technology of the building consisted in quick production of blocks about 12 ft high with time intervals T in between. The scheme is shown in Figure 1. To simplify the problem, a periodic solution (in time and space) was analyzed. It has been shown that the solution quickly approaches the state $u(t+T,x,y+d) = u(t,x,y)$, $0 \le t \le T$ and this state was numerically computed [12, 1960].

The numerical method was essentially the finite difference method with the scheme derived by the cell integration identity principles guaranteeing the balance condition. This technique was close to the technique of Marchuk's identity, elaborated later in [13, 1966].

An essential feature which was introduced much later in the finite element method under the name 'special elements' was used in the computations. In the presence of cooling pipes there was a significant heat sink. Hence, the solution was written in the form

$$u(x,y,t) = v(x,y,t) + w(x,y,t)$$

where $w(x,y,t)$ was the linear solution of a point source (more precisely single circle source) with the intensity $c(t)$ (which was the computed intensity of cooling). Function v was determined by finite difference method as explained above and the hydration heat was included in this term. (For the stationary solution exactly the method of special elements was obtained.)

2) The freezing problem. The building of the dam had to continue during winter when freezing of the concrete in the beginning phase of hardening could create a serious damage. At most the concrete is allowed to freeze for a short time at a depth of 1 - 2 in. The wooden siding for laying the concrete serves also as insulation, and the freezing occurs when the siding is moved in the next building cycle. The main approach was here to solve a stochastic problem for Equation (1) when the boundary conditions are a stochastic function - the outside temperature. The main value and dispersion for the desired information were computed. The theoretical base was described in [14, 1961]. (Let us remark that today a large research project, sponsored by NASA Lewis, solving this problem with stochastic input data is in progress.)

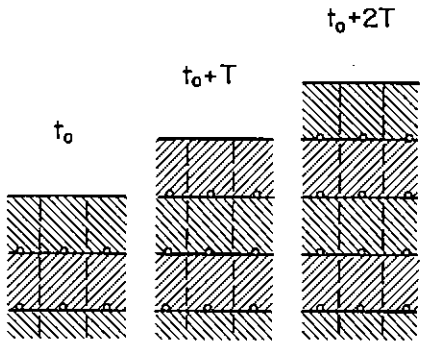


Fig. 1. Schematic state of progressed dam.

3) The elasticity problem. Given the temperature, the thermostresses were computed. The essential problem was the formulation of the problem with respect to material properties including change of elasticity module and creep (relaxation) properties, etc. A rheological model based on a description of the chemical process of hardening was designed and tested in the laboratory.

The numerical solution was based on a series of plane problems in the spirit of splitting up methods. In this phase, J. Nečas contributed significantly to this research. Among others, the theoretical papers [15, 1958] [16, 1959] are directly related to this work. The monograph [17, 1967] by J. Nečas is the only basic monograph which does not avoid unsmooth domains. This monograph and other results of J. Nečas are well known in the West. Various iterative methods were used in the connection of splitting the problem into two dimensional ones. Let us mention one of the type of Schwarz alternating algorithm. Mathematically, the main generalization used was based on the following functional analytical frame (which is today more or less standard), formulated here in the simplest form:

Let P_1, P_2 be projection operators on the subspaces $S_1, S_2 \subset H$. Then $(P_2P_2)^n$ converges pointwise to the projection onto $S_1 \cap S_2$.

4) Error control. The basic idea of the error control of the numerical method was to interpret the numerical solution as exact solution of a problem with slightly different input data. The mathematical models were verified by computation of some simple laboratory experiments. The round-off error was analyzed in a way close to that explained later in the monograph [13, 1968] by α-sequences. In the project Orlík, a team of researchers was involved. In addition to those already mentioned, I. Babuška, L. Mejzlík, J. Jirsák, E. Vitásek, J. Nečas, other researchers participated, especially K. Rektorys, M. Práger,

F. Vyčichlo. Various publications and reports, which directly or indirectly were related to the project, were published during this time.

6. The research in the optimization of the numerical methods, numerical stability and numerical methods in general

During the sixties (1964, 1967), conferences devoted to numerical mathematics were organized. Emphasis was placed on the questions of optimality of the selection of numerical method and numerical stability. These conferences, which took place in the castle Liblice, were held in a very informal working atmosphere. Leading numerical analysts and mathematicians from east and west participated. Let me mention, among others, N. S. Bachvalov, G. Golub, P. Henrici, G. I. Marchuk, F. Olver, S. L. Sobolev, A. N. Tichonov. These conferences were, in my opinion, the very first meetings in the world concentrating specifically on the questions of optimal selection of the numerical method. The various aspects of optimality, theoretical and computational were discussed. Some ideas and results related to this direction obtained in Czechoslovakia were, for example, presented in [13, 1966].

B. Computational mathematics in USSR

In this section I will make* a few subjective comments about the development of computational mathematics in USSR up to the mid 1950. For a systematic survey, we refer to [18, 1948] and [19, 1959].

The theory of approximate methods has a long tradition. For example, the idea of the Galerkin method was introduced in 1915 in [20, 1915]. The Ritz method was investigated in a series of papers of Krylov and Bogoljubov. See e.g., [21, 1917] [22, 1917], [23, 1927], [24, 1931]. The Galerkin method was investigated by various authors in the pre-war period. The book of Kantorovich and Krylov [2, 1936] is likely the first comprehensive book about the numerical solution of partial differential equations. (After the war this book was translated into many languages.)

The Faddejeva's monography [25, 1950] is likely the first comprehensive book about the methods of linear algebra. (It was later translated into English.) Michlin's work and books (e.g. [26, 1950; 27, 1952] and others) about the variational methods were important contributions to the theory of variational methods and computational approaches.

1. Variational methods

As I have already mentioned, the variational methods were investigated by many authors. The investigations addressed both the Ritz method based on a minimization of a quadratic functional as well as the Galerkin method (sometimes called methods of moments or weighted residuals) with the same or different trial and test spaces. The results related to the applications of a minimization are using the Friedrichs extension of the operator to a selfadjoint one. This direction was utilized by Michlin in many of his papers and books, and Michlin was likely the first who used the term "energy space." An important role played the analysis of the energy space and the question to what Sobolev space (in today's terminology) it is equivalent. For example, in [27, 1952] this question is analyzed for basic problems of the elasticity theory. For the mixed problem (e.g. free friction contact boundary condition) the equivalency was analyzed, e.g. in [28, 1951]. The characterization of the energy space for Poisson problem on an infinite domain was discussed in [29, 1953]. The convergence of the Ritz method in the energy space is then directly related to the best approximation. An effort was made to analyze the convergence in the stronger norms $\|u\| = (Au, Au)^{1/2}$ (see, e.g. [30, 1956]) or weaker norms as $\|\cdot\|_{L_\infty}$ (see, e.g. [31, 1941]. The convergence of the Treftz method was analyzed in detail in [26, 1950].

The Galerkin method and general method of moments (also with different trial and test functions) for integral equations were studied in many papers by Krylov and his coworkers. See, e.g. [23, 1927], [32, 1931]. In applications to differential equation, Petrov [33, 1940] used the different trial and test spaces, and the term Galerkin-Petrov method is used sometimes today. Keldys [34, 1942] applied this method to a nonselfadjoint boundary value problem for ordinary differential equation; this paper very likely was the first one establishing the convergence of the method in general setting when applied to an specific problem. The convergence of the Galerkin method was established by Michlin for the operators of the form $A = A_0 + K$ where A_0 is positive definite selfadjoint operator, and $A_0^{-1}K$ is compact in the norm $(A_0x,x)^{1/2}$. See [35, 1948], [36, 1950], [37, 1957]. In [38, 1948], a general functional analytic scheme of numerical method was discussed by Kantorovich. See e.g., [39, 1960]. The main idea is roughly the following. Let us be interested in $Kx = y$ with $x \in X$, $y \in Y$. Then the numerical method solves essentially $K_hx_h = y_h$ where h is a parameter, $h \to 0$ and $x_h \in \tilde{X}$, $y_h \in \tilde{Y}$. There is a one to one mapping φ_h of $\tilde{X}$ onto $\bar{X} \in X$ and ψ_h of $\tilde{Y}$ onto $\bar{Y} \subset Y$. Then one would like to achieve that $\varphi_h^{-1}(x_h)$ is close to the solution of the original problem. For that, one has to essentially achieve that $\varphi_hK - K_h\varphi_h$ is small. In [38, 1948] this approach was applied to a large class of illustrative problems.

Collocation method obviously can also he understood as method of moments and has been treated, e.g. in [39, 1960] in the frame of the above mentioned approach. A method which is very close to the collocation was applied in [40, 1954], [41, 1956] by Vishik. In an abstract form, the Galerkin method and nonlinear problems and a discussion of the approximate method were given by Krasnoselskij in [42, 1954] and in some of his other papers.

*I give here the references to the originals in Russian. Translations of many of these papers and books are now available.

2. Finite difference method

The basic theory of the finite difference method especially related to the stability is in the book by Rjabenkij Filippov [43, 1956]. A handbook of finite difference schemes for partial differential equations was written by Panov [44, 1951]. For hyperbolic equations there is a series of results of Ladyzenskaja and her coworkers. See, e.g. [45, 1952], [46, 1952], [47, 1953].

In the case of elliptic equations, early works are given, for example, in [48, 1952], [49, 1947]. For applications of finite difference for parabolic equation, we refer, for example, to the work by Kamynin [50, 1953].

The general eigenvalue treatment by the finite difference method is given, for example, in [51, 1954].

3. Numerical treatment of differential equations

In the previous sections some early works were presented. They played (by the subjective judgement of the author) important roles in the development of the theory of the numerical method. It is interesting to mention that although the theory of variational method was very advanced, the entire direction of the finite element method was for a long time neglected, and the main emphasis was placed on finite difference method. It seems to be characteristic the finite element method was called until recently variational finite difference method.

Finite difference method was later analyzed in the works of Samarskij, Godunov and many others, and many monographs and text books are available today. In these works the emphasis is placed on the theory. The discussions of computational aspects, numerical experimentation, analyses of the performance of the method on benchmark problems are very rare. Very likely this situation is related to the state of the computer technology in USSR. Nevertheless, the computer situation stimulated various special methodologies as splitting up methods, and various "tricky" iterative procedures which were used in scientific computations. In the area of mathematical modeling and scientific computations, important works have been done by G. I. Marchuk and his coworkers in many papers. The first of his books (see [52, 1958] is addressing modeling and computational methods in reactor analysis. It is interesting to mention that the idea of preconditioning--credited to Buljaev--is mentioned there.

I only mentioned very few papers and results; nevertheless, hopefully, they give some illustrative picture of the character of the research in the USSR in the early post-war period.

References

[1] Láska, V. Hruška, V., Theory and practice of computations, Prague, 1934 (in Czech).

[2] Kantorovich, L. V., Krylov, V. I., Approximate solutions of partial differential equations, Moscow, Leningrad, 1936, pp. 1-588 (in Russian).

[3] von Neumann, J, Goldstine, H., Numerical inverting of matrices of high order, Bull Amer. Math. Soc. 53 (1947), pp. 1021-1099.

[4] Courant, R. Friedrichs, K. O., Lewy, H., Über die partiellen Differenzengleichungern der Physik, Math. Ann. 100, 1928-1929, pp. 32-74.

[5] Rektorys, K., Survey of Applicable Mathematics, Prague, 1966.

[6] Babuška, I., Rektorys, K. Vyčichlo, F., Mathematische Elastizitätstheorie der ebenen Probleme, Academia-Verlag, Berling, 1960.

[7] Rektorys, K., Variational methods in mathematics, science and engineering, Dordricht, Boston, D. Reidel Pub. Co., 1977.

[8] Development of the mathematics in Czechoslovakia in the period 1945-1985 and further perspectives, Charles University, Prague, 1986, 1-217 (in Czech).

[9] Marek, I., Approximate and numerical methods in [8] (1985), pp. 127-143 (in Czech).

[10] Babuška, I., Mejzlík, L., Calculation and measurement of termal stresses in gravity dams, VI Congress des Grandes Barrages, New York, 1958, Question 58, pp. 1-38.

[11] Babuška, I., Mejzlík, L., Vitásek, E., Effects of artificial cooling of concrete in dam during its hardening, VII Congress des Grandes Barrages, Rome, 1961, pp. 1-13.

[12] Vitásek, E., Über die quasistationäre Lösung der Warmeleitungsgleichung, Apl. Mat. 5 (1960), pp. 109-140 (in German).

[13] Babuška, I., Práger, M., Vitásek, E., Numerical Processes in Differential equations, J. Wiley, New York, 1966.

[14] Babuška, I., On randomized solution of Laplace's equation, Časopis Pěst. Mat. 86 (1961), pp. 269-276.

[15] Nečas, J., Solution du problème biharmonique pour le coen infini I, II, Časopis Pěst. Mat., 83 (1958), 257-286., 399-424.

[16] Nečas, J., L'extension de l'espace des conditions aux limites du probleme biharmonique pour les domains a point angeloux, Czechoslovak Math. J. 9 (1959), 339-371.

[17] Nečas J., Les Méthodes directes en théorie des équations elliptiques, Prague Academia, 1967.

[18] Kantorovich, L. V., Krylov, V. I., Approximative methods, Matematika v SSSR za 30 let, Moscow, Leningrad, 1948, pp. 759-801 (in Russian).

[19] Gavurin, M. K., Kantorovich, L. V., Aproximate and numerical methods, Matematika v SSSR za sorok let, 1917-1957, Moscow, 1957, pp. 809-855 (in Russian).

[20] Galerkin, B. G., Rods and plates, Vestnik inženerov 19 (1915) (in Russian).

[21] Krylov, N. M., Sur les généralisations de la méthode de Walter Ritz, C. r. Acad. Sci. 164 (1917), pp. 853-856.

[22] Krylov, N. M., Application of the method of W. Ritz to a system of differential equations, Izv. Acad. Nauk (6) 11 (1917), pp. 521-534.

[23] Krylov, N. M., Sur différents procédes d'intégration approchée en physique mathematique, Toulouse 19 (1927), pp. 167-200.

[24] Krylov, N. M., Approximate solution of basic problems of mathematical physics, Kiev (1931) (in Russian).

[25] Faddejeva, V. N., Numerical methods of linear algebra, Moscow, Leningrad (1950), pp. 1-240 (in Russian).

[26] Michlin, S. G., Variational methods for

solving problems of mathematical physics, Uspekhi Mat. Nauk 5:6 (40) (1950), pp. 3-51 (in Russian).
[27] Michlin, S. G., Problems of the minimum of quadratic functional, Moscow, Leningrad (1952), pp. 1-216 (in Russian).
[28] Edjus, M. I., On the mixed problem of he elasticity theory, Doklady AN SSSR, 76 (1951), pp. 181-184.
[29] Michlin, S. G., Integration of Poisson equation in an infinite domain, Doklady AN SSSR 91 (1953), pp. 1015-1017.
[30] Michlin, S. G., On the Ritz method, Doklady AN SSSR 106 (1956), pp. 391-394 (in Russian).
[31] Kantorovich, L. V., About convergence of variational processes, Doklady AN SSSR 30 (1941), pp. 107-111 (in Russian).
[32] Krylov, N. M., Les méthodes de solution approchée des problémes de la physique mathematique, Paris, 1931, pp. 1-68.
[33] Petrov, G. I., Application of the Galerkin method to the problem of the stability of viscous fluid flow, Prikl. Mat. Mech 4.3 (1940), pp. 3-12.
[34] Keldyš, M. V., Galerkin method for the boundary value problems, Izv. Akad. Nauk SSSR Ser. mat. 6 (1942), pp. 309-330 (in Russian).
[35] Michlin, S. G., About the convergence of the Galerkin method, Doklady AN SSSR, 611 (1948), pp. 197-199 (in Russian).
[36] Michlin, S. G., Direct methods in mathematical physics, Moscow, Leningrad (1950], pp. 1-428 (in Russian).
[37] Michlin, S. G., Variational methods of mathematics physics, Moscow (1957], pp. 1-476 (in Russian).
[38] Kantorovich, L. V., Functional analysis and applied mathematics, Uspekhi Mat. Nauk 3:6 (28) (1948), pp. 89-185 (in Russian).
[39] Kantorovich, L. V., Akilov, G. P., Functional analysis in normed spaces, Moscow, 1960 (in Russian).
[40] Vishik, M. I., Mixed boundary value problems and their approximate solutions, Doklady AN SSSR 97 (1954), 193-196 (in Russian).
[41] Vishik, M. I., Cauchy problem for equation with operator coefficients, mixed boundary value problem for system of differential equations and approximate solution, Matem. Sbornik 39 (81) (1956), pp. 51-148 (in Russian).
[42] Krasnoselskij, M. A., Some problems of nonlinear analysis, Uspekhi Mat. Nauk 9:3 (61) (1954), pp. 57-114 (in Russian).
[43] Rjabenkij, V. C., Filippov, A. F., The stability of finite differences, Moscow (1956), pp. 1-172 (in Russian).
[44] Panov, D., Handbook for numerical treatment of partial differential equations, 5th ed., Moscow (1951), p. 1-182 (in Russian).
[45] Ladyženskaja, O. A., Solution of the Cauchy problem for hyperbolic equations by the finite difference method, Leningrad, Ucen Zap. Univ. 144, ser. mat. 23 (1952), pp. 192-246 (in Russian).
[46] Ladyženskaja, O. A., Finite difference solution of the mixed problem, Doklady AN SSSR 85 (1952), pp. 705-708 (in Russian).
[47] Ladyženskaja, O. A., The mixed problem for the hyperbolic equation, Moscow (1953), pp. 1-280 (in Russian).
[48] Ejdus, D. M., Finite difference solution of boundary value problems, Doklady AN SSSR, 83 (1952), pp. 191-194 (in Russian).
[49] Ljusternik, L. A., Remarks to the numerical solution of boundary value problem for Laplace equation and eigenvalue computation by the finite difference method, Proc. Stekl. Inst. XX (1947), pp. 49-64 (in Russian).
[50] Kamynin, L. I., The applicability of the method of finite differences to the heat problem: I. Uniqueness of the finite difference solution; II. Convergence of the finite differences, Izv. Akad. Nauk SSSR 17 (1953), pp. 163-180, 249-268. (in Russian).
[51] Ljusternik, L. A., Finite difference approximation of Laplace operator, Uspekhi. mat. nauk 9.2 (1954), pp. 3-66 (in Russian).
[52] Marchuk, G. I., Numerical method for nuclear reactors computations, Astomizdat, Moscow (1958), pp. 1-381.

Partially supported by NSF Grant DMS 8315216.

How the FFT Gained Acceptance

James W. Cooley

IBM Watson Research Center,
Yorktown Hts. NY, 10598

Introduction

The purpose of this meeting has been said to be "to bring together pioneers whose vision and research have made major contributions to specific areas of the computing field." As to my own involvement with John Tukey and the fast Fourier transform (FFT) algorithm, I am sorry to have to admit that I had no vision and did little research leading to the paper [1] which apparently was the reason for my invitation to this meeting. As for vision, I seem to have done better by paying attention to the vision of people around me.

The FFT has had a fascinating history, filled with ironies and enigmas. Even more appropriate for this meeting and its sponsoring professional society, it speaks not only of numerical analysis but also of the importance of the functions performed by professional societies.

The Role of Richard Garwin

My involvement with the FFT algorithm, or algorithms as we should probably say, started when Dick Garwin* came to the computing center of the new IBM Watson Research Center some time in 1963 with a few lines of notes he made while with John Tukey at a meeting of President Kennedy's Scientific Advisory Committee, where they were both members. John Tukey showed that if N, the number of terms in a Fourier series is a composite, $N = ab$, then the series can be expressed as an a-term series of subseries of b terms each. If one were computing all values of the series, this would reduce the number of operations from N^2 to $N(a + b)$. Tukey also said that if this were iterated, the number of operations could be reduced from N to $N\log N$. Garwin not only had the insight to see the importance of this idea but also had the drive to pursue its development and publication.

Dick told me that he had an important problem of determining the periodicities of the spin orientations in a 3-D crystal of He^3. I found out later that he was also trying to find ways of improving the ability to do remote seismic monitoring in order to facilitate agreement with Russia on a nuclear test ban and to improve our capability for long range acoustic detection of submarines. Like many others, I did not see the significance in this improvement and put the job on a back burner while I continued with some research I considered more important. However, I was told of Dick Garwin's reputation and, prodded by his occasional telephone calls (some of them to my manager), I produced a 3-dimensional FFT program. I put some effort into designing the algorithm so as to save storage and addressing by over-writing data and I spent some time working out a 3-dimensional indexing scheme that was combined with the indexing within the algorithm.

The Decision to Publish

Garwin publicized the program at first by personal contacts, producing a small but increasing stream of requests for copies. I did a write-up and a version for a program library, but did not plan publishing right away. I gave a talk on the algorithm in one of a series of seminars in our mathematics department. Ken Iverson and Adin Falkoff, the developers of APL, participated and Howard Smith, a member of the APL group, put the algorithm in APL when it was only a language for defining processes and before it was implemented on any machine. This gave the algorithm a thorough working over at the seminar.

Another participant was Frank Thomas, a mathematically-inclined patent attorney, who kept good contacts in the mathematics department. He suggested that there were patent possibilities and a meeting was called to decide what to do with it. It was decided that the algorithm should be put in the public domain and that this should be done by having Sam Winograd and Ray Miller design a device that could carry out the computation. My part of the strategy was to to publish a paper with a footnote mentioning Miller and Winograd and their device. I sent my draft copy to John Tukey, asking him to be co-author. He made some changes and emendations, and added a few references to F. Yates, G. E. P. Box, and I. J. Good. Next came the task of getting it published as quickly as possible. I offered it to *Mathematics of Computation* by sending it to Eugene Isaacson at the Courant Institute of Mathematical Sciences, where I had worked before coming to IBM. I do not know how important

*At that time, a staff member of the Watson Scientific Laboratory at Columbia University. Presently at IBM Watson Research Center, Yorktown Hts., N.Y.

my acquaintance with Eugene was or what effect it had on getting the paper published quickly. In any case, it appeared 8 months after submission in the April, 1965 issue.

I found out later about an excellent paper by Gordon Sande, a very bright statistics student of Tukey's, who was exposed to the factorization idea in one of Tukey's courses. He carried the subject further, showing how it could be used to reduce computation in covariance calculations. After hearing about our paper going out to *Mathematics of Computation,* he did not publish his in its original form. However, he published several other excellent papers [2] one of which showed that the new algorithm was not only faster but more accurate. His form of the FFT is now known as the Sande-Tukey algorithm.

Another result of Dick Garwin's efforts was a seminar run at the IBM Watson Research Center to publicize the algorithm and familiarize IBMers with it. For this, two very capable statisticians, Peter D. Welch and Peter A. W. Lewis, joined me in writing a thick research report describing the algorithm and developing some theory and applications. The three of us then published a series of papers on applications of the FFT. These papers elaborated on the theory of the discrete Fourier transfrom and showed how standard numerical methods should be revised as a result of the economy in the use of the FFT. These included methods for digital filtering and spectral analysis [3].

The IEEE ASSP Digital Signal Processing Committee

The next level of activity came with contact with the speech and signal processing people at MIT- notably Thomas Stockham, Charles Rader, Alan Oppenheim, Charles Rabiner- all of whom have gone on to become highly renowned people in digital signal processing. They had developed digital methods for processing speech, music, and images. The very great obstacle to making their methods feasible was the amount of computing required. This was the first really impressive evidence to me of the importance of the FFT. I was invited to join them and others on the Digital Signal Processing Committee of the IEEE Acoustics Speech and Signal Processing Society.

This committee ran the now famous Arden House Workshops on the FFT in 1967 [4] and in 1969 [5]. These were unique in several respects. One was that they collected people from many different disciplines: there were heart surgeons, statisticians, geologists, university professors, oceanographers, just to name a few. The common interest was in the use of the FFT algorithms and every one of the approximately 100 attending had something useful to say in his presentation. Another thing that was unique was that work was really done. People got together to formulate and work out solutions to problems. An example was where Norman Brenner, then of MIT, designed a program that computed the FFT of a sequence of interferometer data of 512,000 elements, which was larger than available high-speed storage. He did this for Mme. Connes, of the University of Paris, who returned home to perform a monumental calculation of the infra-red spectra of the planets which has become a standard reference book [6]. Others worked out algorithms for data with special symmetries.

Recent Early History of the FFT

Meanwhile, back at the research center, I started learning the history of the FFT. Dick Garwin questioned his colleague, Professor L. H. Thomas of the Watson Scientific Laboratory of Columbia University, who had an office next to his. Thomas responded by showing a paper he published in 1963 [7]. His paper describes a large Fourier series calculation he did in 1948 on IBM punched card machines: a tabulator and a multiplying punch. He said that he simply went to the library and looked up a method. He found a book by Karl Stumpff [8] that was a cook-book of methods for Fourier transforms of various sizes. Most of these used the symmetries and trigonometric function identities to reduce computations by a constant factor. In a very few places Stumpff showed how to obtain larger transforms from smaller ones, and then left it to the reader to generalize. Thomas made a generalization that used mutually prime factors and got a very efficient algorithm for his calculation.

The algorithms of Good and Thomas mentioned above have some very favorable properties, but the constraint that the factors are mutually prime does not give a number of operations proportional to or as low as $N \log N$. Tukey's form of the algorithm, with repeated factors, has the great advantage that a computer program need only contain instructions for the algorithm for the common factor. Indexed loops repeat this basic calculation and permit one to iterate up to an arbitrarily high N, limited only by time and storage.

The credit for what I would consider the first FFT- a computer program implementing this iterative procedure and really giving the $N \log N$ timing, should go to Philip Rudnick of the Scripps Institution of Oceanography in San Diego, California. He wrote to me right after the publication of the 1965 paper to say that he had programmed the radix 2 algorithm using a method published by Danielson and Lanczos in 1942 in the *Journal of the Frankin Insitute,* [9] a journal of great repute which publishes articles in all areas of science, but which did not enjoy a wide circulation among numerical analysts. Rudnick published some improvements in the algorithm [10] in 1966. I had the pleasure of meeting him and asked why he did not publish sooner. He said that his field was not numerical analysis and that he was only interested in getting a computer program to do his data analysis. Thus, we see another failure in communication and lost opportunities, the primary point of Dick Garwin's 1969 Arden House keynote address [11].

Before continuing further with the discussion of the old literature on the FFT, I would like to point out two important concepts in numerical algorithms which had been stated long ago but did not have very much impact until they were demonstrated by the implementation of the FFT on electronic computers. The first is the divide-and-conquer approach. If a large N-size problem requires effort that increases like N^2, then it pays to break the problem into smaller pieces of the same structure. The second important concept is the asymptotic behavior of the number of operations. Obviously this was not significant for small N and, by habit of thought, people failed to see the importance of early forms of the FFT algorithms even where they would have been very useful.

I can illustrate this point by going back to the Danielson and Lanczos paper [9] They describe the numerical problem of computing Fourier coefficients from a set of equally-spaced samples of a continuous function. It is not only a long laborious calculation, but one is also faced with the problem of verifying accuracy. Errors can arise from mistakes in computing or from undersampling the data. Lanczos pointed out that although his use of the symmetries of the trigonometric functions, as described by Runge, reduced computation by a significant factor, one still had an N^2 algorithm. In a previous reading of this paper, I obtained and published [12] the mistaken notion that Lanczos got the doubling idea from Runge. In fact, he only attributes the use of symmetries to Runge, citing papers published in 1903 and

1905 which I could not find. The Stumpff paper [8] gave a reference to Runge and König [13], which does contain the doubling algorithm and which appears to have been a standard textbook in numerical analysis. Thus, it appears that Lanczos independently discovered the clever doubling algorithm and used it to solve the problems of computational economy and error control. He says, in the introduction to [9] on page 366, "We shall show that, by a certain transformation process, it is possible to double the number of ordinates with only slightly more than double the labor." He goes on to say:

> In the technique of numerical analysis the following improvements suggested by Lanczos were used: (1) a simple matrix scheme for any even number of ordinates can be used in place of available standard forms; (2) a transposition of odd ordinates into even ordinates reduces an analysis for $2n$ coefficients to two analyses for n coefficients; (3) by using intermediate ordinates it is possible to estimate, before calculating any coefficients, the probable accuracy of the analysis; (4) any intermediate value of the Fourier integral can be determined from the calculated coefficients by interpolation. The first two improvements reduce the time spent in calculation and the probability of making errors, the third tests the accuracy of the analysis, and the fourth improvement allows the transform curve to be constructed with arbitrary exactness. Adopting these improvements the approximation times for Fourier analyses are: 10 minutes for 8 coefficients, 25 minutes for 16 coefficients, 60 minutes for 32 coefficients, and 140 minutes for 64 coefficients.

The matrix scheme in (1) reduces the data to even and odd components so that real cosine and sine transforms are computed. The rest of the process makes use of the symmetries of the sines and cosines, similar to the methods of Runge. After this, he uses the doubling algorithm. Step (2) is what we have been calling the twiddle factor multiplication and in Step (3) he does the butterfly calculation but observes accuracy by comparing the two inputs: the Fourier coefficients of the sub-series. Thus, it appears that Lanczos had the FFT algorithm and if he had an electronic computer, he would have been ready to write a program permitting him to go to arbitrarily high N. It may seem strange to us, then, to see his remark on Page 376, "If desired, this reduction process can be applied twice or three times."

This is an outstanding example of the difference in point of view between different generations of numerical analysts. Here was the doubling algorithm, capable of doing Fourier transforms in $N \log N$ operations, described in detail. It seems to be appreciated as much as a method for checking accuracy as for reducing computing. The authors did not foresee the possibility of automating the procedure. In fact, in the very beginning of the Danielson and Lanczos paper, it is presented as an economical way of doing the computation without using a mechanical analyzer which was available at the time. Then they published it in the *Journal of the Franklin Institute* where it was unnoticed until Philip Rudnick, who was not a numerical analyst, revived it but ignored the opportunity to show it to the world. Lanczos later published his *Applied Analysis* in 1956 [14] with only a few words and a footnote (page 239) referring to the Danielson and Lanczos paper. I find no references at all in his later books including his 1966 book, *Discourse on Fourier Series* [15].

Gauss and the FFT

After learning of the above early papers, I wrote what I thought to be the very early history of the FFT algorithm [12] going back to Runge and König. Some years later, while working on his book, [16], Herman Goldstine told me of a paper by Gauss [17] that contained the FFT algorithm. I got a copy of the paper, which was in a neo-classic Latin that I could not read. The formulas and a slight recognition of parts of words indicated he was doing a kind of Lagrangian interpolation that leads to the basic FFT algorithm. I put this aside as a very interesting post-retirement activity.

A few years later, some old signal processing friends, Don Johnson and Sidney Burrus at Rice University, told me that they put a very bright energetic graduate student, Michael Heideman on the trail of Gauss and the FFT. He not only translated the Gauss article but found and described many others who wrote of FFT methods, between Gauss and my early references. [18].

Conclusion

This story of the FFT can be used to help one appreciate the important functions of professional societies such as the ACM and SIAM. Some recommendations one can make are:

- It is obvious that prompt recognition and publication of significant achievements is an important goal.
- Careful attention to a review of old literature may offer some rewards. Furthermore, awards for outstanding achievements should lead to a review of the old literature.
- Communication between mathematicians, numerical analysts, and workers in a very wide range of applications can be very fruitful.
- Do not publish papers in neo-classic Latin.

References

[1] J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series", *Mathematics of Computation,* Vol 19, p. 297 (April 1965).

[2] W.M.Gentleman and G. Sande,"Fast Fourier Transforms for Fun and Profit," 1966 Fall Joint Computer Conf., AFIPS Proc. Vol 29, Washington, D.C.: Spartan, 1966, pp.563-578

[3] J. W. Cooley, P. A. W. Lewis and P. D. Welch, "The Application of the Fast Fourier Transform Algorithm to the Estimation of Spectra and Cross-Spectra," *J. Sound Vib.,* Vol. 12(3), pp. 339-352, July 1970.

[4] Special Issue on Fast Fourier Transform and Its Application to Digital Filtering and Spectral Analysis, *IEEE Trans. Audio Electroacoustics,* Vol. AU-15, pp. 43-117, June 1967.

[5] Special Issue on Fast Fourier Transform *IEEE Trans. Audio Electroacoustics,* Vol. AU-17, No. 2, pp 65-186, June 1969.

[6] J. Connes, P. Connes, et J.-P. Maillard, "Atlas des Spectres dans le Proch Infrarouge de Vénus, Mars, Jupiter et Saturne," *Éditions du Centre de la Recherche Scientifique* 15, quai Anatole France Paris VII', 1969

[7] L. H. Thomas, "Using a Computer to Solve Problems in Physics," *Applications of Digital Computers,* Ginn and Co., Boston, Mass., 1963.

[8] Karl Stumpff, *Grundlagen und Methoden der Periodenforschung,* Springer, Berlin, 1937. Karl Stumpff, *Tafeln und Aufgaben zur Harmonischen Analyse und Periodogrammrechnung,* Springer, Berlin, 1939.

[9] G. C. Danielson, and C. Lanczos, "Some Improvements in Practical Fourier Analysis and Their Application to X-ray Scattering From Liquids," *J. Franklin Inst.,* Vol 233, nos. 4 and 5, pp. 365-380 and 435-452, April and May, 1942.

[10] Philip Rudnick, "Note on the Calculation of Fourier Series,", *Math. Comp.,* Vol. 20, No.3, pp 429-430, July 1966.

[11] R. L. Garwin, "The Fast Fourier Transform as an Example of the Difficulty in Gaining Wide Use for a New Technique," Special Issue on Fast Fourier Transform, *IEEE Trans. Audio Electroacoustics,* Vol. AU-17, No. 2, pp 69-72, June 1969.

[12] J. W. Cooley, P. A. W. Lewis and P. D. Welch, "Historical Notes on the Fast Fourier Transform," *IEEE Trans. Audio Electroacoustics,* Vol. AU-15, pp 76-79, June 1967.

[13] C. Runge and H. König,"Band XI, Vorlesungen Über Numerisches Rechnen," *Die Grundlehren der Matematischen Wissenschaften* Verlag von Julius Springer, Berlin, 1924

[14] C. Lanczos, *Applied Analysis,* Prentice Hall, Inc. Englewood CLiffs, N.J. 1956

[15] C. Lanczos, *Discourse on Fourier Series,* Oliver and Boyd, Edinburgh and London, 1966

[16] H. H. Goldstine, *A History of Numerical Analysis from the 16th Through the 19th Century,* Springer-Verlag, New York, Heidelberg, and Berlin, pp. 249-253.

[17] C.F.Gauss, "Nachlass: Theoria interpolationis methodo nova tractata," Carl Fricdrich Gauss, Werke, Band 3, Gottingen: Koniglichen Gesellschaft der Wissenschaften, 1866, pp. 265-303.

[18] M. T. Heideman, D. H. Johnson, and C. S. Burrus, "Gauss and the History of the Fast Fourier Transform," *The ASSP Magazine* Oct. 1984, Vol. 1, No. 4.

Early Contributions to Numerical Analysis*

J. Barkley Rosser
University of Wisconsin

Invited to talk on early contributions to numerical analysis and other contributions, both institutional and mathematical, of interest to computer scientists.

Comments on the Development of Numerical Analysis From Classical Analysis

R. S. Varga
Kent State University

The interplay between classical analysis and numerical analysis will be illustrated from the history of the University of Michigan Summer Schools, the Gatlinburg Meetings and the foundation of of Numerische Matematik.

THE DEVELOPMENT OF ODE METHODS: A SYMBIOSIS BETWEEN HARDWARE AND NUMERICAL ANALYSIS

C. W. Gear
R. D. Skeel

Department of Computer Science
University of Illinois at Urbana-Champaign

Abstract

The history of the numerical solution of ordinary differential equations is surveyed from its origins three centuries ago up to the early 1970s. The increasing demands for the solution of ODEs, especially for exterior ballistics and celestial mechanics, has been a primary stimulus of and a significant influence on the early development of computers starting with the analog differential analyzers and continuing to the first wired-program digital computers—whose form foreshadowed future developments in parallel computers. At the same time the hardware has, of course, affected the algorithms used, but this has resulted in surprisingly few innovations in numerical techniques.

1. Hand Calculation

We begin with hand calculation because it is interesting in its own right and because it is important to appreciate what was known about numerical methods before the use of computers.

Analog devices for specific calculations date from at least the start of the 15th century (Goldstine[1] , p. 5) and for general calculations from 1620 when Gunter invented a forerunner of the slide rule (Goldstine, *op cit*, p. 4). However, any engineer trained before the introduction of the inexpensive four-function calculator knows that a slide rule is not a particularly useful device for numerical integration. The first digital arithmetic tool was built at the about same time by Schickard (Goldstine, *op cit*, p. 6) and reinvented in 1642 by Pascal (1623–1662) who built a digital adder/subtracter. Thirty years later, Leibniz (1646–1716) built a digital machine that surpassed Pascal's by being able to also perform multiplication and division. However, it seems that practical calculating machines were not available until the mid-19th century (Randell[2], pp.2, 3).

Moulton [3] states that "Newton in his *Principia* was the first to find approximate solutions of differential equations by numerical processes" and goes on to say, "The successors of Newton ... applied the method to problems in celestial mechanics to which more general methods are not adapted. For example, if a comet passes near Jupiter ..., its motion can be most conveniently followed during the interval by numerical processes." This must be a reference to one of the very important calculations in the history of science, namely the predicted delay in the return of Halley's comet of 1682 by Clairaut, Lalande, and Lepaute in 1748. Lalande wrote[4] , "During six months we calculated from morning to night, sometimes even at meals; the consequence of which was, that I contracted an illness which changed my constitution for the rest of my life. The assistance rendered by Madame Lepaute was such that without her we should never have dared to undertake this enormous labour; in which it was necessary to calculate the distance of each of the two planets, Jupiter and Saturn, from the comet, separately for every successive degree, for 150 years." The differential equations they solved[5] were not for the orbit itself but rather for the perturbations due to the two large planets. However, logarithms were probably the only calculating aids they had. The result was a prediction that the comet would reach perihelion in April 13, 1749, which was in error by only 31 days. Sagan and Druyan[6] state that this "powerfully supported ... the Newtonian view that we live in a clockwork universe" and quote Laplace as saying that "the regularity which astronomy shows us in the movements of the comets doubtless exists also in all phenomena." The next return of Halley's, in 1835, was predicted with an

This work was supported by the Department of Energy under contract DOE DEFG02-87ER25026.

error of only 5 days, and the prediction for 1910 was off by only 2.7 days[7]. This won a 1000 mark prize for P. H. Cowell and A. C. D. Crommelin, who took into account the influence of the 7 planets from Venus outward to Neptune. Cowell[8] is known for the formula

$$y_{n+1} - 2y_n + y_{n-1} = \frac{h^2}{12}(f_{n-1} + 10f_n + f_{n+1})$$

for the special second order ODE $y'' = f(y,t)$.

It is Leonhard Euler[9] in 1768, according to Goldstine[10], who "is basically responsible for the present–day methods." His chapter on *De Integratione Aequationum Differentialium per Approximationem* not only gives a description of the "Euler" or "polygon" method for the general problem

$$\frac{dy}{dx} = V(x,y)$$

but in paragraph 660 gives a general description of the step–by–step Taylor series method. Several examples are given for the Taylor method, the first being $V(x,y) = x^n + cy$, but no numerical results are given. The Euler method was the basis of the first existence proof for ODEs given by Cauchy a century later.

The higher derivatives needed for the Taylor series method can become very complicated. G. W. Hill[11] in 1878 gives a recursion that simplifies these calculations for the gravitational force potential. He was interested in calculating the position of the moon, important in navigation for the determination of longitude, using two second order ODEs. A sixth order Taylor method was used to generate numerical tables and graphs, and Jacobi's integral was used as a check.

The so–called Adams–Bashforth and Adams–Moulton formulas were both derived[12] by John Couch Adams (1819–1892) in 1883 to assist an investigation by Bashforth of capillary action[10]. Earlier, Adams had shared in the discovery of Neptune by calculating its position based on the motion of Uranus. In the work with Bashforth a fixed stepsize was used with its value sufficiently small so that fifth order differences were negligible. The process for a scalar equation was to predict y, evaluate the first derivative f, and then perform a single Newton–Raphson correction without reevaluating f.

The implicit Adams formula was employed in a fairly sophisticated way by Sir George H. Darwin[13], also of Cambridge, in 1897 in an effort to calculate periodic orbits for a restricted three body problem. Jacobi's integral is used to reduce the problem to three coupled first order ODEs with arclength as the independent variable. Darwin makes no reference to the work of Adams but derives the implicit Adams formula as a straightforward application of the calculus of finite differences. Using the symbol ΔE^{-1} for backward differences, he derives the generating function for the coefficients of the backward differences of the f values. Variable stepsize is used with doubling accomplished by using every other derivative value and halving by interpolating the derivative values. Darwin remarked that the ratio of the largest increment to the smallest was 32 for some of the orbits because of sharp bends in the orbits. For most of the calculation the 4th order formula was used with the stepsize determined by the size of the second and third order differences of the derivative values. The integration is started using low order formulas with small stepsizes. Also the order is lowered from four to three just after going through a "quasi–cusp." Little is said about prediction , but the corrector iteration is said to be repeated until convergence, which is "usually rapid". Darwin gives a detailed description of the computational process including a "schedule for computation" which depended heavily on the use of 5–figure tables of logarithms with some use of 4–figured tables, and he gives pages and pages of numerical results. Also he mentions "the prodigious amount of work involved" and the early death of his first computer, as well as acknowledging the Royal Society for providing two–thirds of the expenses of these computations.

Forest Ray Moulton (1872–1952), a professor of astronomy at the University of Chicago, spent April to June 1918 "computing the the trajectories of projectiles as a basis for the construction of range tables" for the U. S. Army. This experience resulted in the publication of his book *New Methods in Exterior Ballistics*[14] in 1926, which describes in great detail methods for computing ballistics tables, including anti–aircraft tables. The process was to solve using 5–place tables a simple nonlinear system that accounts for gravity and air resistance and then to solve using 4–place tables and larger stepsizes complicated linearized equations for corrections that account for minor factors such as the rotation of the earth. The simple nonlinear equations are given by

$$\frac{d^2x}{dt^2} = -F\frac{dx}{dt},$$

$$\frac{d^2y}{dt^2} = -F\frac{dy}{dt} - g$$

where

$$F = \frac{G(v)e^{-ay}}{C},$$

$$v^2 = \left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2,$$

and G is given by tables of empirical data. The 4th order implicit Adams formula was used with the 4th order explicit Adams as predictor. The stepsize was chosen on the basis of the difference between the predicted value of f_{n+1} and its corrected value. With no reference to previous use of these formulas, Moulton does a derivation for uniform stepsize using Δ_k as the symbol for the kth backward difference. Step doubling and halving is performed as by Darwin; however for starting his 4th order scheme he used a block implicit method, an idea that has reappeared many times since. One chapter of the book is devoted to convergence theory. In 1930 he wrote *Differential Equations*[3] with little of additional interest. There is in this book the idea of Adams method being based on the replacement of $f(x(t),t)$ by an interpolating polynomial, concern with the choice of initial stepsize, and a reference to Darwin[13].

William E. Milne in a 1926 paper[15,16] discusses several methods based on numerical integration including the well known 4th order implicit Milne–Simpson formula. A 4th order explicit formula is proposed as a predictor and an appropriate multiple of the predictor–corrector difference is taken as an estimate of the (local truncation) error — the *Milne device*. This technique has seen wide use for the Adams method, for which it can be rigorously justified; however it is not valid for the Milne–Simpson method because of weak instability. Apart from the problem of error estimation the poor stability of these methods becomes a problem when computations are performed on a large scale. Thus increases in computing speed have led to greater concern and study of numerical stability and the abandonment of methods like those of Milne.

Carl Runge (1856–1927), an applied mathematician, seems to be the first to derive the very popular type of method based on resubstitution. His 1895 paper[10,17] derives two popular 2nd order 2–stage methods, one based on the midpoint rule and the other on the trapezoid rule. He also derives a 3rd order 4–stage method of short–lived interest. Collatz[18] gives interesting biographical information and a photograph.

In 1900 Karl Heun[19] introduces a restricted class of what we know as Runge–Kutta methods and determines the coefficients for about a dozen formulas. His list includes the three formulas of Runge that we have just mentioned. He also manages to construct a 4th order 8–stage formula, but the most interesting is his 3rd order 3–stage formula:

$$k_1 = hf(y_n),$$

$$k_2 = hf(y_n + \frac{1}{3}k_1),$$

$$k_3 = hf(y_n + \frac{2}{3}k_2),$$

and

$$y_{n+1} = y_n + \frac{1}{4}k_1 + \frac{3}{4}k_3.$$

Heun then goes on to discuss systems of equations, error analysis, and graphical methods for solving ODEs.

It was Wilhelm Kutta (1867–1944) in his 1901 paper[20] who introduced the general class of explicit Runge–Kutta methods as we know it today and wrote down the nonlinear equations for the parameters that must be solved in order to attain a given order of accuracy. He solves these equations for the 1–parameter family of 2nd order 2–stage methods, the 2–parameter family of 3rd order 3–stage methods, and the 2–parameter family of 4th order 4–stage methods. As a special case he obtains the very famous "classical" Runge–Kutta formula based on Simpson's rule as well as a 4th order 4–stage formula based on the 3/8–ths rule, which he calls Kutta's method. The formula given in the previous paragraph he calls Heun's method. Also, he obtains a 5th order 6–stage formula.

Runge–Kutta–type methods were devised for general and special second order ODEs by Nyström[16,21] in 1925.

To give some idea of the scale of the computations performed, we quote p.125 of Collatz[18] concerning the Norwegian mathematician Carl Störmer:

> In order to confirm his theory of the aurora borealis, Störmer and his colleagues spent several years calculating numerous orbits of electrons in the earth's magnetic field .... The computed orbits were reproduced very closely by ... experimental work. ... 4500 working hours were needed for 120 orbits.

Störmer([22], 1907) is known for a family of explicit formulas for special second order ODEs, the simplest of which is

$$y_{n+1} - 2y_n + y_{n-1} = h^2 f_n.$$

Another large–scale computation was that of L. J. Comrie[2] who in 1929 used a punched card system to calculate future positions of the moon and in the process punched half a million cards.

Hand calculation was very important until the 1960s (and reemerged in the 1970s with the invention of the handheld calculator). The practical details of hand calculation are found in many numerical analysis books, such as those by Collatz[18] and Hartree[23]

In conclusion we see that the use of numerical methods was, out of necessity, quite sophisticated at the time of the introduction of computers. Thus we have that in 1947 Sir Richard Southwell is reported[24] to have said that "Human beings get a feeling for their problems as they work with them; they develop intuitions which cannot be automated or communicated to a cold, heartless computer." And he was right, for twenty years at least. The use of automated computing machinery lead to the use of simplistic numerical methods which continues to this day among many who do simulations. As Hartree[23] explains in the first numerical analysis book to consider seriously the use of computers, "with an automatic machine it may be best to obtain the same results by a simple process involving a large number of steps to save the time that would be taken in planning, programming, and coding a less simple method using fewer numerical steps."

2. Analog Computation in ODEs

When, in 1822, Babbage invented the difference engine, it would appear that digital technology was in a better position to cope with the numerical solution of ODEs than analog computers. Indeed, although Babbage failed to implement his ideas in the available technology, a machine based on his ideas was completed by a Swedish printer, Scheutz, in 1853 (see Goldstine[1], p 16). The difference engine was capable of calculating successive values of a polynomial by constructing a difference table. Had such machines become commonly available, it would have been surprising if they had not been adapted to the solution of ODEs since it was a short step from them to the digital differential analyzer. However, the planimeter (a device for measuring the area bounded by a simple curve) was invented shortly before that time by Hermann and improved by Maxwell and Professor James Thompson. According to Goldstine, Thompson did not present his idea to the Royal Society for almost a decade because no one saw any use for it until his brother, Lord Kelvin, discussed the problem of a tide–calculating machine. Kelvin used the invention to construct a machine to compute Fourier coefficients by quadrature. He then went on to plan its application to the solution of the general linear second order ODE:

$$\frac{d}{dx}\left[\frac{1}{F(x)}\frac{dy}{dx}\right] + y = 0. \tag{2.1}$$

He considered the use of two of Thompson's mechanical integrators to compute successive iterates of

$$\frac{d}{dx}\left[\frac{1}{F(x)}\frac{dy_{m+1}}{dx}\right] + y_m = 0 \tag{2.2}$$

from

$$y_{m+1} = \int F(x)[c - \int y_m\, dx]\, dx.$$

This requires the coupling of the output of the inner integrator to the input of the outer integrator. Unfortunately, the mechanical integrator of the time had no power gain, so that technology was insufficient to support the idea until Vannevar Bush[25] and his colleagues at MIT independently developed the idea half a century later. It is interesting to speculate what might have heen the developments had Babbage's difference engine heen slightly more successful and available to Kelvin. Instead, Goldstine says about Kelvin's harmonic analyzer: "Here we see for the first time an example of a device which can speed up a human process by a very large factor, as Kelvin asserts. That is why Kelvin's tidal harmonic analyzer was important and Babbage's difference engine was not." Certainly, an analog integrator was far better suited to the harmonic analysis problem, a quadrature of a product of two functions, one of which was sinusoidal, than any simple adaptation of Babbage's difference engine could have been. Among other problems, the construction of a sinusoidal function is mechanically nearly trivial but digitally computationally intensive. The planimeter seemed to be popular in the first part of this century, e. g.[26], for use in graphical techniques in order to perform the integration of a Picard iteration after plotting the current approximation to the derivative.

In his discussion of the integration of Eq. (2.1) using the iteration given by Eq. (2.2) Kelvin wrote, "After thus altering, as it were, y_1 into y_2 by passing it through the machine, the y_2 into y_3, by a second passage through the machine, and so on, the thing will, as it were, become refined into a solution which will be more and more nearly rigorously correct the oftener we pass it through the machine. If y_{i+1} does not sensibly differ from y_i, then each is sensibly a solution." (Quoted from Goldstine[1], p. 50.) Such a device would still involve considerable human intervention, although Kelvin was "feeling satisfied, feeling I had done what I wished to do for many years." But at this point, he saw that the iteration could be avoided. "Compel agreement between the function fed into the double machine and that given out by it." He then showed, according to Hartree[27] how, in principle, this can be done by making a second interconnection between the two integrators so that the output of the second is used continuously as the integrand of the first, and that this interconnected system of integrators evalnates a solution of the equation directly, so that the general differential equations of the second order with variable coefficients may be solved by a machine in a single process. Thus, the analog computer for the solution of ODEs was designed, and it would appear that the ideas in the design were to have an initial impact on the organization of the digital computer.

In 1930 Bush built the first working differential analyzer, and hy 1940 there were over half a dozen full size differential analyzers in use ([1], p. 97). More intriguing[28], however, was the "Hartree Differential Analyzer" built of Meccano parts at the University of Manchester, which according to Hartree ([1], p. 95) gave an accuracy of 2% and was used for serious computation. However, limited precision was a problem even for the best differential analyzers in applications such as astronomy. Moreover, until 1942 the "differential analyzer was programmed manually, with a wrench in one hand and a gear in the other"[29], the process often taking a day or two[1].

3. First Digital Computers and Digital Differential Analyzers

It was the numerical solution of differential equations[2] in his thesis work that in 1937 lead Howard Aiken, a physics instructor at Harvard, to plan an automatic computing machine. The largely mechanical 50–foot long Mark I was demonstrated in 1943 and served from May 1944 to 1959. The solntion of ODEs (by the Runge–Kntta method) was one of five suggested scientific applications. In the same period Bell Telephone Laboratories built a series of computers using electromagnetic relays. Model III (1944–1958) was called the "Ballistic Computer" because it solved fire control problems. A copy of the Model V built in 1947 went to the Ballistic Research Labs at the Aberdeen Proving Grounds. Alt[30] states that systems of ODEs "have so far furnished the main portion of prohlems for the machine. Both Picard's method and step–by–step methods have been tried, and the latter have so far been found more efficient. As an example, in a system of order five each step required about three minutes .... The machine can be directed to change the length of step. .... the machine can handle systems beyond capacity of differential analyzers and the ENIAC." The stepsize was adjusted according to the number of corrector iterations needed for convergence.

One of the early digital computers, the ENIAC: Electronic Numerical Integrator And Computer, was initially programmed using a technique similar to the plug–hoard wiring of the early card calculating machines. Any of its 20 registers (or memory cells) could be used as an accumulator, much as each integrator in a differential analyzer is used as an accumulator to "accumulate" the integral of a particular integrand in time. Just as the integrators in an analog computer operate in parallel, so did the arithmetic on each of the accumulators. In fact, it appears that the ENIAC was designed with the intention of solving ordinary differential equations by imitating the techniques used on analog computers, hecause the construction of the ENIAC was preceded by a "Report on an Electronic Diff.* Analyzer (2 April 1943)" by Mauchly, Eckert, and Brainerd where the "*" means either "erential" or "erence." The historical importance[29] of the ENIAC was its electronic hardware, which made it 10 times faster than a differential analyzer and 100 times faster than a human computer. The very first digital electronic computer of Atanasoff for solving linear systems of equations was also according to Burks[29] modeled after the differential analyzer.

The ENIAC was funded for the purpose of preparing firing and bombing tables, in particular for the aiming of anti–aircraft guns[2,29]. (Also the ORDVAC built at the University of Illinois went to the Ballistics Research Laboratory.) According to Goldstine[10] the method used in the first solutions of differential equations on the ENIAC was the Heun method. This was a second order 3–stage metbod, not the better known method described in Section 1. The program was written by Burks[29] to integrate a trajectory.

It is interesting to note that after the ENIAC had been in operation and it had been observed that it took considerable time to prepare a "program" and incorporate it in wiring, the machine was modified so that a sequence of instructions could be read as sequence of two–digit numbers from a function table and executed serially. In order to keep the logic simple, one register was dedicated for use as an accumulator and the rest were relegated to use as memory registers. Hartree commented in a set of lectures given in Illinois in 1948[27] that "It seems likely that it will increase the scope and value of the Eniac as a general purpose machine, but that the older form of control may be better for extensive work on comparatively simple problems such as the step–by–step integration of ordinary differential equations for which the ENIAC was originally designed." Was the ease of preparation more due to the tedium of wiring the program, or was it due to the difficulty of programming for parallel operations? It may have been that

the problems of parallelism in digital computation were first encountered nearly 40 years ago.

Special purpose digital machines for the integration of ODEs continued to be built. However, unlike ENIAC, these DDAs used short wordlength fixed point binary arithmetic. The first of these, MADDIDA[31] was built in 1950. It used the Euler method with a stepsize so small that only the last bit or two changed with each step, and it employed a "residue register" to minimize the accumulation of rounding errors – more specifically, for each operation of an integration step a running sum of the rounding errors was maintained and each time that operation was performed the result was rounded in such a way that the running sum was minimized. (It should be appreciated that the effects of the rounding errors are not simply additive and so this technique does not completely eliminate the accumulation of rounding errors.) However, because of their slowness DDAs never became as widely used as analog differential analyzers, let alone general purpose digital computers. Apparently[32] , they have seen some use in special purpose applications such as real–time control systems (and graphics), and also Adams and Heun integration have been used in some DDAs.

4. Early ODE Programs: the Effect of Small Memories

The early stored program computers had extremely small memories. The EDSAC had 512 words of 17 bits, while the first ILLIAC had 1024 words of 40 bits. Because of this, space for both code and intermediate results was at a premium, so the codes had to be simple and methods which required little temporary storage were utilized. The Runge–Kutta–Gill (RKG) method was in that class. It is a particular case of a fourth–order Runge–Kutta method in which the coefficients were chosen by Stanley Gill[33] so that it was not necessary to store all of the intermediate derivatives at each step. In a general explicit Runge–Kutta implementation, we compute

$$k_i = hf(Y_i)$$

where

$$Y_i = y_n + \sum_{j=1}^{i-1} \beta_{ij} k_j$$

and save each of the k_i for i from 1 to q. Finally we compute

$$y_{n+1} = y_n + \sum_{i=1}^{q} \gamma_i k_i.$$

This requires $q \times s$ storage cells, where s is the number of components in the system.

In the RKG method, the β_{ij} were chosen so that

$$\beta_{ij} = \beta_{i-1,j}$$

for $j < i-1$ which meant that

$$Y_i = Y_{i-1} + \beta_{i,i-1} k_{i-1}$$

and hence that prior values of k_j did not have to be saved. The RKG method was implemented on the EDSAC[34] and later implemented by Wheeler, when he was visiting Illinois, as code #27 for the ILLIAC I and was apparently its first ODE solver. ILLIAC was first operational on Labor Day, 1952. The extant documentation indicates that this code was "machine tested" but is dated June 1952 so may have been tested on the ORDVAC, or on the ILLIAC prior to its full operation. It was revised for the ILLIAC in October 1953 as code #114, apparently because of a change in the operating system(!). Then, in January 1954 it was renumbered as code F1 in the "Reorganization of the ILLIAC Library", a library naming system[35] whose initial letter indicated the general class of code. At the time, there were 81 programs extant in the library, although the numbering had reached 128. Only 23 of these subroutines were concerned with the numerical solution of problems, and an additional 17 with the evaluation of elementary functions. At that time, the library contained three subroutines for numerical solution of ODEs. Program F2 was a Milne method for initial value problems written by Gene Golub and placed in the library in October 1953. It used a feature of the operating system called an *interlude*. Because of the lack of memory, sections of code could be executed during load time (the "interlude") so they did not occupy memory space during run time. The Milne code computed the required starting values prior to t_0 from user–supplied values of the first and second derivatives at t_0. This was done in the interlude. Routine FA1 was a boundary value problem solver as a "Floating–point auxiliary." It was called that because floating point was itself handled by a subroutine that interpreted a pseudo–code, so a "subroutine" written in the interpretive language was then called an auxiliary subroutine. Although the interpreter also provided indexing (which was not available in the 1950–designed ILLIAC), only 8 of the 40 numerical programs were floating–point auxiliaries, the remainder used the 40–bit fixed–point arithmetic.

The number of instructions was kept to a minimum in the early programs. The independent variable, t, was treated as an additional dependent variable with the defining equation $dt/dt = 1$. This added one memory location for the derivative value, but no code to compute it since it could be specified as a constant at load time. It also added the space for the k_i values. This was a saving over the space that would have been needed for the instructions to treat t separately, since four sets of LOAD, ADD, and STORE would have been needed for the general four–stage RK method, although the technique added several slow multiplications to each step. Since on machines of that time, multiplication was about a tenth of the speed of addition (for the ILLIAC I, fixed–point addition was 72 micro–seconds and fixed–point multiplication averaged about 700 micro–seconds), this represented a considerable space–time trade–off.

5. Adaptive Programs

The development of adaptive codes for ODEs had to await an increase in memory capacity, either primary or secondary. We have attempted to determine where the first adaptive codes were developed, but without success. Here we mention a few of the developments of which we are aware.

There were undoubtedly many variable–stepsize one–step methods written. A method was described by Merson in a 1957 report[36] and came to be known as the Runge–Kutta–Merson method. At Illinois in 1958, Nordsieck wrote the variable stepsize routine F6 which used the "classical" fourth–order formula Runge–Kutta method on the point set (0, 0.5, 0.5, 1.0) and the error estimator

$$E = y_{n+1} - y_n - hk_3$$

which was held to be no larger than $2^{-[3e/4]}$ where e was the "number of bits of accuracy" required. (This was a fixed point code.) As in many codes of the time, the stepsize was restricted to powers of 2 "to reduce round–off error." It also saved time in the multiplication by h in machines of the time, since multiplication by a multiplier with many zero bits was faster due to the reduced numbers of adds in the add–and–shift implementation.

Two years prior to that, D. E. Muller (of "Muller's method" for rootfinding) had coded routine F5 which used the RKG method to integrate until a condition was satisfied. The condition was a zero value of a specified dependent variable.

One of the early variable–stepsize multistep methods was the Nordsieck modification of Adams method in which he stored scaled derivatives. Fred Krogh has pointed out to us that the idea was essentially developed much earlier in a paper by Thomas[37] presented at a September 1950 ACM meeting in which he discussed the implementation of variable–stepsize Adams methods using divided differences. He proposed a method for computing the derivatives of a function from the divided difference table as follows. Suppose that we have a divided difference table for the approximations y_i of $y(t_i)$ where the t_i are not necessarily equally spaced. Suppose that Δ_n^j is the j–th divided difference at t_n, that is,

$$\Delta_n^j = \frac{\Delta_n^{j-1} - \Delta_{n-1}^{j-1}}{t_n - t_{n-j}}$$

To calculate the first q derivatives of y at t_n, we introduce a set of additional points $t_{n+i} = t_n$ for $i = 1, 2, \cdots, q$ and calculate Δ_{n+i}^i for the q–th degree polynomial passing through y_{n-i}. Since this polynomial has a constant q–th divided difference, the desired Δs can be calculated from

$$\Delta_{n+j}^{k-1} = \Delta_{n+j-1}^{k-1} + \Delta_{n+j}^{k}(t_{n+j} - t_{n+j-k})$$

for $1 \le j < k \le q$, where $\Delta_{n+j}^q = \Delta_n^q$. Since Δ_{n+i}^i is the i–th divided difference evaluated at the same points t_n, it is exactly the i–th derivative of the polynomial divided by $i!$. From these derivatives, Thomas proposed to use Taylor's series to advance the solution over one time step chosen so that the error estimate was constant. This was proposed in the context of an explicit method such as Adams–Bashforth, and the divided differences were to be calculated for the derivative, f. After y_{n+1} had been calculated by Taylor's series, a value of f_{n+1} could be evaluated and the divided difference table could be extended one more line. He then went on to point out that it was not necessary to store differences, rather the "divided derivatives" could be computed directly. This is essentially the Nordsieck implementation of Adams method, although it appeared that Nordsieck[38,39] was unaware of Thomas' work when he devised his method that appeared in Illinois code F7 in August 1961. One other interesting remark in Thomas's 1952 paper states that the method " is convenient for differential equations in which the derivative is given implicitly." However, although he was at the Watson Scientific Computing Lab, Columbia University, there is no indication that Thomas was thinking of a computer implementation of his method, and it seems that his method was never actually used.

Thomas proposed using a variable stepsize but fixed order method: Adams methods implemented in hand calculations with a fixed stepsize and a difference table naturally lent themselves to variable order because it was relatively simple to add or drop

differences according to their size. This was the basis of what is probably the first variable order code development for multistep methods by Krogh. It was written in late '66 and '67, and was presented at the 1968 IFIP meeting[40] in a session that may have been the first numerical analysts' meeting on stiff equations. Krogh's code used modified divided differences, an implementation that is generally viewed as the best for a general multistep code today. It handled higher order ODEs, provided output at arbitrary points, and used a corrector formula of one order higher than the predictor formula in order to obtain a longer interval of absolute stability.

6. Stiff ODEs

The earliest paper on stiff differential equations, by Curtis and Hirschfelder[41] described the use of the BDF methods. What they were interested in doing was to find smooth solutions to problems whose Jacobian had very large eigenvalues, which is not quite what we mean by a stiff problem today. Soon after, Mitchell and Craggs[42] found that these BDFs were not (zero–)stable for orders greater than 6. The next significant development was the seminal paper of Dahlquist[43] which defined A–stability and showed the classical result that the order of an A–stable multistep method cannot exceed two. This result was later extended to a larger class of methods in the Daniel–Moore conjecture[44] which essentially says that the order of an A–stable method cannot exceed twice its "degree of implicitness", that is, the number of derivatives involved implicitly in each step where each higher derivative counts as an additional derivative, as does a derivative at an additional point. The result was shown for implicit Runge–Kutta methods by Ehle[45]. Although it is outside of the time period we are discussing, mention should be made of the beautiful results of Hairer, Norsett, and Wanner[46] on order stars which proved this conjecture for all cases. Because of the order limitation implied by A–stability, people looked for less restrictive stability requirements that would permit higher–order, useful methods. A(α)–stability was defined by Widlund[47] to mean that the stability region included a wedge of half–angle α symmetric about the real axis in the left–half plane. A–stability corresponded to $\alpha = \pi/2$, but Widlund showed that for any smaller α it was possible to find methods of up to fourth order. (This was later extended to arbitrary order[48,49].) As α approaches $\pi/2$ the coefficients of such methods become large as the error coefficient must approach infinity if the order exceeds two. A different approach was taken by Gear[50,51], who defined "Stiff Stability" to mean that the method was stable in a half–plane to the left of a negative real value and in a finite region from there up to the origin. Most important about both approaches was the fact that non–A–stable methods were explored, and that these methods were realized to be the most effective for general stiff problems.

Until the late 60's, stiff equations were often being solved on analog computers. Electrical engineers had problems of sufficient size that they could only be handled on digital computers, so there were a number of papers on methods for stiff equations beginning to appear in the literature[52], but many problems arising in chemistry were still sufficiently small that analog computers were adequate. It was due to the use of analog computers in the then Applied Mathematics Division of Argonne National Laboratory that the first author (henceforth identified by the first person singular) became involved in stiff ODEs. I was a summer visitor in 1965 and 1966 (a program that encouraged a lot of interactions between numerical analysts and computer scientists) and in 1965 had extended the Nordsieck ideas to higher–order methods. In 1966, a person using the analog computer at Argonne to solve a set of seven equations describing a chemical kinetics problems made the statement "you people will never be able to handle these type of problems with your digital computers." This was enough of a challenge to encourage me to search for methods. The concept of stiff stability was more of an afterthought: at the time it was realized that the problem required a method that was stable in most of the left half plane and especially around the origin. Multistep methods were examined, and since the stability far from the origin was determined by the polynomial $\sigma(\xi) = \sum \beta_i \xi^{k-i}$, the "most stable" such polynomial, ξ^k, was investigated. Only later was it realized that these were just the BDF methods used by Curtis and Hirschfelder (but dismissed by Henrici[53] for the very valid reason that they had large error coefficients and were not even zero–stable for orders exceeding six). At the same time, Krogh was studying the problem and also settled on the BDF methods independently in a report that was unfortunately never published since he received a copy of my report[50] before completing his. A version of his report with subsequent revisions was printed later as a TRW internal report[54] but never appeared in the open literature. To return to Argonne in the summer of 1966, I wrote a preliminary version of a stiff integrator using fixed–order BDF methods, but with stepsize control using a Nordsieck vector implementation. The chemistry problem was run successfully, although the proponent of the analog computer was not prepared to accept the answers from a digital calculation "until they match the results obtained from the analog computer". They did, when free of programming errors, and, to the chemists delight, they produced extremely good "mass balance" results, the mass balance being a linear invariant of the system representing the number of each atom present in the reaction. It was realized that linear invariants were preserved within round–off error by the class of methods being considered (see Gear[55] for a discussion) so that preservation of mass balance was meaningless. Nonetheless, for a while, there was considerable debate whether or not the mass balance equations should be used to eliminate variables or as a check on the computation. At one point during the Argonne visit, a program error caused a change of several percent in the answers, but the mass balance remained good to the ten digits printed, and it was somewhat difficult to convince the user that the answers were wrong.

After the Argonne visit I embedded the method in a package, ODESSY: Ordinary Differential Equation Solver System, written with three graduate students but never published. It accepted a set of ODEs in a symbolic form, translated them to a machine language subroutine and differentiated them to obtain a machine language subroutine for the Jacobian needed in a stiff integration method, and then proceeded to integrate them automatically. Stiff methods were used in all cases because no one had yet thought of changing methods in midstream and stiff methods would work on nonstiff equations at a modest penalty. The order had to be varied because it did not seem reasonable to expect the user to choose the order. The package was of little value because it was too restrictive — it was impossible to append programs to compute the coefficients of the system or use tabular data, for example. For this reason, users at Kirtland AFB were unable to make effective use of the program, and I removed the integrator from the package and made it into a subroutine for their use. Work would have stopped there (and have been of relatively limited value) had it not been for the suggestion of George Forsythe in 1969 when I was on sabbatical leave from Illinois at Stanford and the Stanford Linear Accelerator Center. I was preparing the draft manuscript for a book[56] in a Prentice–Hall series edited by Forsythe. Forsythe suggested that a book would be much more valuable if it included working programs, so I spent part of that year rewriting the earlier code for publication in the algorithms section of *CACM*[57] and in the book. DIFSUB was, by today's standards, a poorly written code since it went to great lengths for speed. At the time, subroutine calls were expensive (particularly on the IBM 360 series on which it was first implemented), and so no internal subroutines were used: the equivalent was achieved with assigned GO TOs. Furthermore, may computer systems were still using early FORTRAN II compilers, and so the code was written in a subset of Fortran. The impact of this on the internal allocation of working space lead to convoluted code.

Analog computers remained an important tool for chemical kinetic problems for some time. Around 1970 there was a meeting in Boston between chemists and numerical analysts to discuss the use of digital methods in chemical kinetics. At the time there was a proposal to build a large analog computer with a price tag of several million dollars to solve some high–altitude kinetic problems. However, digital methods were accepted in time to avoid this effort.

Bibliography

[1] H. H. Goldstine. *The computer from Pascal to von Neumann.* Princeton University Press, Princeton, New Jersey, 1972.

[2] B. Randell (ed.). *The Origins of Digital Computers, Third Edition.* Springer–Verlag, Berlin, 1982.

[3] F. R. Moulton. *Differential Equations.* Macmillan, New York, 1930.

[4] P. Moore and J. Mason. *The Return of Halley's Comet.* W. W. Norton, New York, 1984.

[5] F. R. Moulton. *An Introduction to Celestial Mechanics, second edition.* Macmillan, New York, 1914.

[6] C. Sagan and A. Druyan. *Comet.* Random House, New York, 1985.

[7] R. S. Richardson. *Getting Acquainted with Comets.* McGraw–Hill, New York, 1967.

[8] P. H. Cowell and A. C. D. Crommelin. "Investigation of the motion of Halley's Comet from 1759–1910," In: *Greenwich Observations 1909.* Neill, Bellevue, England, 1910.

[9] F. Engel and L. Schlesinger. *Leonhardi Euleri Opera Omnia, Ser. 1, Vol. XI.*, Leipzig, 1913.

[10] H. H. Goldstine. *A history of numerical analysis from the 16th through the 19th century.* Springer Verlag, New York, 1977.

[11] G. W. Hill. "Researches in the lunar theory, chapter II," *Amer. J. of Mathematics* (1878), Vol. I, pp. 245–260.

[12] F. Bashforth and J. C. Adams. *An Attempt to Test the Theories of Capillary Action.*, Cambridge, 1883.

[13] G. H. Darwin. "Periodic orbits," *Acta Mathematica* (1897), Vol. 21, pp. 99–242.

[14] F. R. Moulton. *New Methods in Exterior Ballistics.* University of Chicago, 1926.

[15] W. E. Milne. "Numerical integration of ordinary differential equations," *Am. Math. Mo.* (1926), Vol. 33, pp. 455–460.

[16] W. E. Milne. "Step–by–step methods of integration," In: *Numerical Integration of Differential Equations,* W. E. Milne A. A. Bennett and H. Bateman, ed. Dover, New York, 1956, pp. 71–87.

[17] C. D. T. Runge. "Über die numerische Auflösung von Differentialgleichungen," *Math. Annalen* (1895), Vol. 46, pp. 167–178.

[18] L. Collatz. *The Numerical Treatment of Differential Equations, Third Edition.* Springer–Verlag, Berlin, 1960.

[19] K. Heun. "Neue Methode zur approximativen Integration der Differentialgleichungen einer unabhängigen Veränderlichen," *Zeit. Math. u. Phys.* (1900), Vol. 45, pp. 23–38.

[20] M. W. Kutta. "Beitrag zur näherungsweisen Integration oder Differentialgleichungen," *Zeit. Math. u. Phys.* (1901), Vol. 46, pp. 435–453.

[21] E. J. Nyström. "Uber die numerische Integration von Differentialgleichungen," *Acta Soc. Sci. Fen.* (1925), Vol. 50, No. 13, pp. 1–55.

[22] C. Störmer. "Sur les trajectoires des corpuscules électrisés", Archives des Sciences physiques et naturelles, Geneva, July–October 1907, pp. 63–pp.

[23] D. R. Hartree. *Numerical Analysis.* Oxford Univ. Press, 1952.

[24] G. Birkhoff. "Computing developments 1935–1955, as seen from Cambridge, U.S.A.," In: *A history of Computing in the Twentieth Century*, J. Howlett N. Metropolis and Gian–Carlo Rota, ed. Academic Press, New York, 1980, pp. 21–30.

[25] V. Bush. "The differential analyzer: a new machine for solving equations," *Journal of the Franklin Institute* (Oct 1931), Vol. 212, pp. 447–488.

[26] H. Levy and E. A. Baggott. *Numerical Solution of Differential Equations.* Dover, New York, 1950.

[27] D. R. Hartree. *Calculating Instruments and Machines.* University of Illinois Press, Urbana, IL, 1949.

[28] G. R. Stibitz. "D. R. Hartree, Calculating Instruments and Machines," *Math. Tables Aids to Computation* (January 1950), Vol. 4, p. 114.

[29] A. W. Burks. "From ENIAC to the stored–program computer: two revolutions in computers," In: *A History of Computing in the Twentieth Century*, J. Howlett N. Metropolis and Gian–Carlo Rota, ed. Academic Press, New York, 1980, pp. 311–344.

[30] F. L. Alt. "A Bell Telephone Laboratories' computing machine — II," *Math. Tables and Aids to Computation* (April 1948), Vol. 3, pp. 69–84.

[31] T. C. Bartee, I. L. Lebow and I. S. Reed. *Theory and Design of Digital Machines.* McGraw–Hill, New York, 1962.

[32] R. B. McGhee and R. N. Nilsen. "The extended resolution digital differential analyzer: a new computing structure for solving differential equations," *IEEE Trans. on Computers* (Jan. 1970), Vol. C–19, No. 1, pp. 1–9.

[33] S. Gill. "A process for the step–by–step integration of differential equations in an automatic digital computing machine," *Proceedings of the Cambridge Philosophical Society* (1950), Vol. 47, pp. 96–108.

[34] M. V. Wilkes, D. J. Wheeler and S. Gill. *The preparation of programs for an electronic digital computer.* Addison–Wesley Press, Cambridge, Mass, 1951.

[35] S. Gill. "Reorganization of the Illiac library", University of Illinois Digital Computer Laboratory Internal Report #55, Urbana, IL, Jan 1954.

[36] R. H. Merson. "An operational method for the study of integration processes", Proceedings of a symposium on data processing, Weapons Research Establishment, Salisbury, South Australia, 1957.

[37] L. H. Thomas. "The integration of ordinary differential systems," *Ohio State University Engineering Experiment Station News* (June 1952), Vol. 24, pp. 8–9,31–32.

[38] A. Nordsieck. "On numerical integration of ordinary differential equations", University of Illinois Coordinated Science Laboratory Report # R–127, Urbana, IL, Oct 1961.

[39] A. Nordsieck. "On numerical integration of ordinary differential equations," *Mathematics of Computation* (Jan 1962), Vol. 16, pp. 22–49.

[40] F. T. Krogh. "A variable order multistep method for the numerical solution of ordinary differential equations," In: *Information Processing 68*, A. J. H. Morrell, ed. North Holland, Amsterdam, 1969, pp. 194–199.

[41] C. F. Curtiss and J. O. Hirschfelder. "Integration of stiff equations," *Proceedings U. S. National Academy of Science* (1952), Vol. 38, pp. 235–243.

[42] A. R. Mitchell and J. W. Craggs. "Stability of difference relations in the solution of ordinary differential equations," *Math. Comp.* (1953), Vol. 7, pp. 127–129.

[43] G. Dahlquist. "A special stability problem for linear multistep methods," *BIT* (1963), Vol. 3, p. 27.

[44] J. W. Daniel and R. E. Moore. *Computation and theory in ordinary differential equations.* Freeman and Co., 1970.

[45] B. L. Ehle. "High order A–stable methods for the numerical solution of systems of differential equations," *BIT* (1968), Vol. 8, pp. 276–278.

[46] G. Wanner, E. Hairer and S. P. Norsett. "Order stars and stability theorems," *BIT* (1978), Vol. 18, pp. 503–517.

[47] O. Widlund. "A note on unconditionally stable linear multistep methods," *BIT* (1967), Vol. 7, pp. 65–70.

[48] R. D. Grigorieff and J. Schroll. "Uber A(α)–stabile Verfahren hoher Konsistenzordnung,".

[49] A. K. Kong. "A search for better linear multistep methods for stiff problems", Report R–77–899, Dept. of Computer Sci., Univ. of Illinois at Urbana–Champaign, Dec. 1977, 101 pps.

[50] C. W. Gear. "The numerical integration of stiff differential equations", University of Illinois Department of Computer Science Report #221, Urbana, IL, Jan 1967.

[51] C. W. Gear. "The automatic integration of stiff ordinary differential equations," In: *Information Processing 68*, A. J. H. Morrell, ed. North Holland, Amsterdam, 1969, pp. 187–193.

[52] W. Liniger. "Optimization of a numerical method for stiff systems of ordinary differential equations", IBM Research Report # RC–2198, Yorktown, NY, 1968.

[53] P. Henrici. *Discrete variable methods in ordinary differential equations.* Wiley, New York, 1962.

[54] F. T. Krogh. "The numerical integration of stiff differential equations", TRW Report 99900–6573–R000, Redondo Beach, CA, Mar 1968.

[55] C. W. Gear. "Maintaining solution invariants in the numerical solution of ODEs," *SIAM J. of Scientific and Statistical Computing* (July, 1986), Vol. 7, pp. 734–743.

[56] C. W. Gear. *Numerical initial value problems in ordinary differential equations.* Prentice–Hall, Englewood Cliffs, New Jersey, 1971.

[57] C. W. Gear. "Algorithm 407 – DIFSUB for solution of ordinary differential equations," *Communications of Association for Computing Machinery* (Mar 1971), Vol. 14, pp. 185–190.

A HISTORICAL REVIEW OF ITERATIVE METHODS

David M. Young*

The University of Texas

*The preparation of this paper was supported in part by The National Science Foundation through Grant DCR - 8518722, by the Department of Energy through Grant A505-81ER10954, and the U.S. Air Force Office of Scientific Research and Development through Grant AF-85-0052. The U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so for U.S. Government purposes.

1. Introduction

Originally, as suggested by the title, it was intended that this paper would give a broad historical review of iterative methods. However, it soon became apparent that in the available time and space it would be necessary to focus on a much narrower topic, namely, the history of the development of the successive overrelaxation method (SOR method) and of polynomial acceleration techniques for speeding up the convergence of basic iterative methods.

To begin the discussion a brief summary of the highlights of the SOR theory will be given in Section 2. This will be followed in Section 3 by a description, from my perspective as a graduate student at Harvard working under the direction of Garrett Birkhoff, of the development and analysis of the SOR method. Section 4 is devoted to polynomial acceleration techniques including Richardson's method, Chebyshev acceleration and second-degree methods. The close relation which sometimes holds between these methods and certain forms of the SOR method is described. This in some sense "rounds out" the theory of both types of methods.

2. Review of SOR Theory

In this section we review some of the highlights of the theory of the SOR method. For a more complete coverage of the SOR theory see, e.g., Young [1971].

Let us consider the problem of solving the linear system

$$Au=b \tag{2.1}$$

where A is a given real nonsingular NxN matrix and b is a given column vector. We assume that the diagonal elements of A do not vanish. Letting D be the diagonal matrix whose diagonal elements are the same as those of A we can rewrite (2.1) in the form

$$u=Bu+c \tag{2.2}$$

where

$$\begin{cases} B=I-D^{-1}A \\ c=D^{-1}b \end{cases} \tag{2.3}$$

The Jacobi method is defined by

$$u^{(n+1)}=Bu^{(n)}+c \tag{2.4}$$

To define the successive overrelaxation method (SOR method) we define the strictly lower triangular matrix L and the strictly upper triangular matrix U such that B=L+U. The SOR method is defined by

$$u^{(n+1)}=\omega\{Lu^{(n+1)}+Uu^{(n)}+c\}+(1-\omega)u^{(n)} \tag{2.5}$$

or, equivalently by

$$u^{(n+1)}=L_\omega u^{(n)}+k_\omega \tag{2.6}$$

where

$$k_\omega=\omega(I-\omega L)^{-1}c$$

$$L_\omega=(I-\omega L)^{-1}(\omega U+(1-\omega)I) \tag{2.7}$$

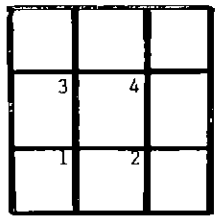
The main result of the SOR theory is a relation between the eigenvalues $\{\lambda_i\}$ of L_ω and the eigenvalues $\{\mu_i\}$ of B. This relation holds if A is consistently ordered (CO). We give a definition of a CO matrix in terms of graph theory. To do this we construct an undirected graph of A. This graph consists of points $P_1,P_2,\ldots,P_N$, where N is the order of A, and edges P_iP_j. The edge P_iP_j, which is a line joining P_i and P_j, belongs to the graph if $a_{i,j}\neq 0$ or $a_{j,i}\neq 0$.

A matrix is said to have Property A if every simple closed path has an even number of edges. A matrix is CO if for every simple closed path there are as many edges P_iP_j with $i<j$ as there are with $i>j$. It can be shown that for any matrix with Property A we can permute the rows and corresponding columns of A to obtain a CO matrix. Evidently, if A is CO then A has Property A.

As an example, consider the model problem involving Poisson's equation

$$u_{xx}+u_{yy}=-1 \tag{2.8}$$

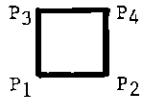
on the unit square $[0,1]x[0,1]$ with zero boundary values. Using the standard 5-point difference equation with h=1/3 we get the mesh



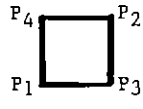
and, with the indicated ordering of the mesh points, the linear system

$$\begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix} = \begin{bmatrix} 1/9 \\ 1/9 \\ 1/9 \\ 1/9 \end{bmatrix}$$

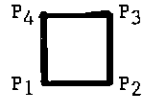
In this example the matrix has Property A and is consistently ordered since its graph is



With the "red-black" ordering the graph is



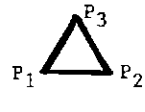
and the matrix is CO. However the matrix corresponding to the "to-and-fro" ordering whose graph is



has Property A but is not CO. Finally we note that for the matrix

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

the graph is



and the matrix does not have Property A.

The key eigenvalue relation for the SOR theory is given by

$$(2.9) \qquad \lambda+\omega-1=\omega\mu\sqrt{\lambda}$$

Here λ is an eigenvalue of L_ω and μ is an eigenvalue of B. If A is also symmetric and positive definite (SPD) then a bound on the spectral radius $S(L_\omega)$ can be found by solving (2.9) for each eigenvalue μ or B. (Actually, the eigenvalues of B are real and $S(B)<1$, and the largest value of $|\lambda|$ corresponds to the eigenvalue S(B) of B.) We can minimize $S(L_\omega)$ by letting $\omega=\omega_b$, where

$$(2.10) \qquad \omega_b = \frac{2}{1+\sqrt{1-S(B)^2}}$$

The value of $S(L_{\omega_b})$ is given by

$$(2.11) \qquad S(L_{\omega_b})=\omega_b-1=r$$

For the model problem defined above we have $S(B) = \cos \pi h \sim 1-\frac{1}{2}\pi^2h^2$ and

$$(2.12) \qquad \omega_b = \frac{2}{1-\sin\pi h} \qquad S(L_{\omega_b}) = \frac{1-\sin\pi h}{1+\sin\pi h} \sim 1-2\pi h$$

The number of iterations needed for convergence with the Jacobi method is $O(h^{-2})$ while with the SOR method with $\omega=\omega_b$ it is $O(h^{-1})$. Thus there is an order-of-magnitude improvement.

Suppose now that A is a "red-black" matrix of the form

$$(2.13) \qquad A = \begin{bmatrix} D_R & H \\ K & D_B \end{bmatrix}$$

where D_R and D_B are square diagonal matrices. It can be shown that

$$(2.14) \qquad \| L_{\omega_b}^n \|_{D^{1/2}} = r^n\{ n(r^{1/2}+r^{-1/2})+[n^2(r^{1/2}+r^{-1/2})^2+1]^{1/2}\}$$

$$\approx 5n\ r^n$$

where $r = \omega_b-1$. Normally one would expect r^n instead of $5nr^n$. The presence of the factor of n slows the convergence of the SOR method somewhat and is caused by the existence of a principal vector of grade 2 for L_{ω_b} for the eigenvalue r. The derivation of (2.14) was made possible by the availability of formulas for the eigenvectors and principal vectors of L_{ω_b} in terms of the eigenvectors of B; see Young [1971].

3. Early Work on The SOR Method

In 1948 when I was a graduate student at Harvard in search of a thesis topic, I sought advice from Garrett Birkhoff. I had originally thought that I might work on Lie groups, but Birkhoff suggested that instead I work on "relaxation methods". He gave me several references including the book by Sir Richard Southwell [1946] and the report by Shortley, Weller and Fried [1940] which appeared as Bulletin No. 107 of the Engineering Experiment Station of Ohio State University.

Relaxation Methods

I found the book by Southwell rather hard to understand, but after studying it and some other papers, I was able to get some idea what relaxation methods were all about. Actually the term "relaxation methods" in the broad sense referred to a procedure for obtaining an approximate numerical solution to a problem involving a partial differential equation, usually elliptic. This procedure includes both the replacement of the given problem by a discretized problem involving a partial difference equation and also the solution of the discretized problem as a system of linear algebraic equations. In the narrower sense of the term, and the sense in which I believe it is now understood, "relaxation methods" simply refer to an iteration procedure for solving a system of linear algebraic

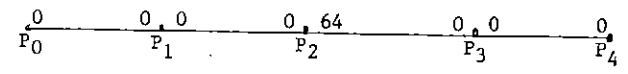
equations.

I will now give my perception of relaxation methods. I am somewhat nervous about doing so in the presence of Professor Leslie Fox who worked with Southwell in actually applying the method to practical problems.

Consider the problem of solving the linear system

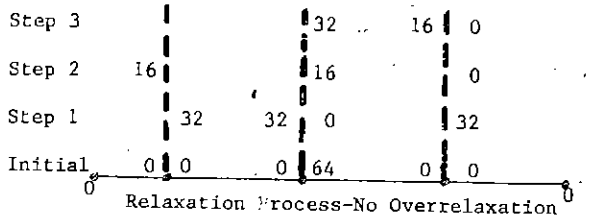
$$(3.1)\qquad \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}=\begin{bmatrix} 0 \\ 64 \\ 0 \end{bmatrix}$$

We relate this to a one-dimensional problem involving 5 mesh points with 3 interior points as indicated below.

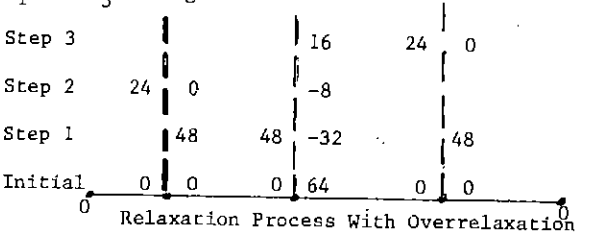


Letting the boundary values and the initial solution guesses be zero compute the residuals $r_i=(b-Au)_i$ for each interior point obtaining $r_1=0, r_2=64, r_3=0$. Mark the initial quesses to the left of each interior point and the initial residuals to the right.

We now are ready to carry out the relaxation process. Increments are indicated to the left of each point and cumulative residuals are shown to the right. In the example shown below on step 1 we add an increment of 32 to u_2 reducing r_2 to 0 but increasing r_1 an r_3 to 32. On step 2 we add an increment of 16 to u_1 reducing r_1 to 0 but increasing r_2 to 32. The process can be continued until all r_i are negligible. At that point we accumulate the increments and get final values of the u_i. Then comes the heartbreak step - check whether the actual residuals based on the u_i are correct! If there is an error it probably means that all the work to date has been wasted. I must confess that this happened all-to-frequently when I tried to use relaxation methods.



The convergence of the relaxation process can often be speeded up by "overrelaxing". We suppose we overrelax at u_2 by 50% but do not overrelax at u_1 and u_3. We get



Note that, in some sense, the residuals tend to cancel out. After 3 steps we are much better off than we were when no overrelaxation was used.

Systematic Relaxation Procedures

Partly because of my lack of success in getting the relaxation process to converge before the occurence of numerical errors I was interested in the possibility of more systematic procedures which could be adapted to the then emerging high-speed computers. The paper by Shortley et al [1940] was concerned with the Liebmann method, (Liebmann [1918]), an iterative method for solving the discrete analogue of Laplace's equation. The Liebmann method is a special case of the Gauss-Seidel method. The method can be regarded as a systematic form of relaxation where one chooses a fixed ordering of the equations and relaxes the residuals (without overrelaxation) one at a time, repeatedly sweeping through the region. The same is done with the SOR method except that one over-relaxes at each step by the factor ω.

The paper by Shortley et al [1940] gave an analysis of the convergence properties of the Liebmann method in terms of the eigenvalues and eigenvectors of certain matrices. There was also a discussion of the role of principal vectors of the iterative matrix which were not eigenvalues. Several very interesting conjectures were presented which stimulated my interest. Besides this, estimates for the rate of convergence of the Liebmann method were given.

Not too long after I began my work, Sir Richard Southwell visited Birkhoff at Harvard. One day when he, Birkhoff and I were together, I told him what I was trying to do. As near as I can recall, his words were

> "any attempt to mechanize relaxation methods would be a waste of time."

This was somewhat discouraging, but my propensity of making numerical errors was so strong that I knew that I would never be able to solve significant problems except by machines. Thus, though discouraged, I continued to work.

Richardson's Method

Besides the Liebmann method, I also studied a method of L. F. Richardson [1910] which can be written in the form

$$(3.2)\qquad u^{(n+1)}=u^{(n)}+\gamma_{n+1}(b-Au^{(n)})$$

Here the parameters $\gamma_1,\gamma_2,\ldots$ are to be chosen to speed up the convergence. One approach for choosing the $\{\gamma_i\}$ would be to choose an integer m and use the parameters in a cyclic order $\gamma_1,\gamma_2,\ldots,\gamma_m,\gamma_1,\gamma_2,\ldots$. If A is SPD then the problem of choosing the best values of the $\{\gamma_i\}$ is equivalent to that of minimizing Δ where

$$(3.3)\qquad \Delta = \operatorname*{Max}_{m(A)\le\nu\le M(A)} \prod_{i=1}^{m} |1-\gamma_i\nu|$$

where m(A) and M(A) are the smallest and largest eigenvalues of A respectively.

It is now well known that the optimum values of the $\{\gamma_i\}$ can be found by the use of Chebyshev polynomials; this is discussed in Section 4. Unfortunately I was not acquainted with Chebyshev polynomials, nor was I able to find "good" $\{\gamma_i\}$ which would have resulted in substantially more rapid convergence than the Liebmann method.

Besides considering the Liebmann method I also considered a variant of Richardson's method where $\gamma_1=\gamma_2=\ldots=\gamma$ for some fixed γ and where new

values are used as soon as available. I called this method the "Richardson-Liebmann method". If $a_{11}=a_{22}=\ldots=a_{N,N}=c$ and if $\gamma=\omega/c$ then the Richardson-Liebmann method reduces to the SOR method with relaxation factor ω.

A key breakthrough was the observation, that for certain small linear systems derived from the model problem the eigenvalues of the Gauss - Seidel method are the squares of those of the Jacobi method. A proof was then developed for a general region assuming that the mesh points were numbered in a "consistent" ordering. The relation was also found to be true for more general elliptic equations. The same methods were used to show that a relation could also be obtained between the eigenvalues of the SOR method and the eigenvalues of the Jacobi method.

Garrett Birkhoff had previously called my attention to a paper of Hilda Geiringer [1949]. This paper was concerned with iterative methods for solving general linear systems -- not merely those arising from the solution of partial differential equations. This more general point of view motivated an attempt to define as general a class of matrices as possible so that the basic SOR eigenvalue relation holds. As a result, the concepts of consistently ordered matrices and matrices with Property A were developed.

Another paper which proved to be most useful to me was that of Temple [1938]. In this paper it was shown that the problem of solving the linear system $Au=b$ with A SPD was the same as that of minimizing the quadratic form $Q(u)=\frac{1}{2}(Au)-(b,u)$. This quadratic form was used to show that for many problems, the spectral radius $S(B)$ of the Jacobi method is a monotone function of the size of the region. This is often useful in estimating the optimum value of ω for a non-rectangular region.

In 1949, a paper by Snyder and Livingston appeared in MTAC, one of the few outlets for numerical analysis research results at that time. This paper described a procedure written in Univac machine language for solving Laplace's equation on a rectangle using the Liebmann method. I modified the program to handle Richardson's method and the Richardson-Liebmann method. A couple of years later when I went to the Aberdeen Proving Ground, I was able to finally to collaborate in writing a computer program based on the SOR method; see Young and Lerch [1953]. It was truly exciting to see the machine carry out the iterative process and also to see that the observed convergence properties of the SOR method were close to those predicted by the SOR theory.

Garrett Birkhoff was extremely helpful to me in my research efforts, especially in providing guidance and encouragement, pointing out references, and, of course, reading numerous drafts of the thesis. The name "successive overrelaxation method" was suggested by him and I feel it was an excellant choice.

After leaving Harvard I often returned during summers to work with Birkhoff and, on occasion, with Dick Varga, Bob Lynch, and others. Other years we would often work at the Argonne National Lahoratory.

I would like to recall one anecdote. As mentioned earlier, when the optimum value of ω is used with the SOR method, there is a principal vector of grade 2 associated with the largest eigenvalue. This slows the convergence. I was able to actually find the principal vector and use it to obtain a bound on the 2-norm of the error as a function of the iteration number n. However, I was not able to find such a bound for the ∞-norm. I was afraid that when I mentioned this to Birkhoff, he would insist that I find a bound for the ∞-norm. I was also concerned that if he were to decide this it might not be too easy to convince him to change his mind. However, when I saw him, I didn't explain the situation very well. He may have thought I was arguing <u>for</u> the ∞-norm. In any case, he appeared to be somewhat irritated and told me in no uncertain terms that the ∞-norm was old-fashioned and that I definitely should use the 2-norm. I made a silent sigh of relief and did not mention the subject again.

Almost simultaneous with the completion of my thesis was the appearance of the paper by Stanley Frankel [1950] in MTAC (now "Mathematics of Computation" (MOC)). Frankel carried out a complete analysis of the SOR method, which he referred to as the "extraporated Liebmann method" for the Laplace equation on the rectangle. He also gave an analysis of the second-order Richardson method -- see the discussion in Section 4.

I had considerable difficulty in getting my thesis (Young [1950]) published. This was in part due to the scarcity of periodicals which would even consider numerical analysis papers. Another reason was the difficulty I had in condensing it. It was very painful to be required to throw out some of the results which I felt were interesting.

The original thesis was 150 pages long. (Incidentally, since xeroxing was unknown, all formulas had to be filled in by hand on all 3 copies.) By May 1951, the paper had been condensed to 75 pages and submitted to the Transactions of the American Mathematical Society. The referee, Hilda Geiringer, correctly pointed out that the paper was "far from ready for publication." She made a number of very useful criticisms and suggestions. After much agony and discarding of material, the paper was reduced to 15 pages and resubmitted. Some time later, I was told that I had cut out too much and some expansion was needed to make the paper intelligible. A final iteration increased the length to 20 pages and the paper (Young [1954]) finally appeared in 1954 - four years after the thesis was written. It can truly be said that without Garrett Birkhoff's continued interest and encouragement, the paper would never have seen the light of day!

4. Polynomial Acceleration

In this section we give a brief history of polynomial acceleration procedures for speeding up the convergence of certain basic iterative methods. By a <u>basic iterative method</u> we mean a one-step method of the form

$$u^{(n+1)}=Gu^{(n)}+k \tag{4.1}$$

where for some nonsingular matrix Q we have $G=I-Q^{-1}A$ and $k=Q^{-1}b$. Examples of basic iterative methods are the Richardson basic iterative method, where $Q=I$, and the Jacobi method, where $Q=D$. We assume throughout this discussion that I-G is similar to an SPD matrix and hence all eigenvalues of G are real and less than unity.

A procedure for accelerating the convergence of the basic iterative method (4.1) is a

polynomial acceleration procedure if for some sequence of polynomials $P_0(x), P_1(x), P_n(x)\ldots$ such that $P_n(1)=1$ for all n we have

$$u^{(n)}-\bar{u}=P_n(G)(u^{(0)}-\bar{u}) \tag{4.2}$$

where $\bar{u}$ is the true solution of (2.1). The acceleration is linear if $P_0(x), P_1(x)\ldots$ do not depend on $u^{(0)}, u^{(1)}, \ldots$. Richardson's method, discussed in Section 3, is a polynomial acceleration procedure for the Richardson basic iterative method

$$u^{(n+1)}=(I-A)\,u^{(n)}+b \tag{4.3}$$

Given a set of polynomials $P_0(x), P_1(x), \ldots$ such that $P_n(1)=1$, for all n, one can easily show from (4.2) that

$$u^{(n)}=u^{(0)}+Q_{n-1}(G)\delta^{(0)} \tag{4.4}$$

where $\delta^{(0)}=Cu^{(0)}+k-u^{(0)}$ and $Q_{n-1}(x)=(x-1)^{-1}(P_n(x)-1)$

One can compute $u^{(1)}, u^{(2)}, \ldots$, given $u^{(0)}$, if one is given the coefficients $\{\alpha_{n,k}\}$ of the $\{P_n(x)\}$. Also, if one is given the roots of the $\{P_n(x)\}$ for each n one can compute $u^{(1)}, u^{(2)}, \ldots$ by a variable extrapolation procedure similar to Richardson's method (3.2). However, if the polynomials $\{P_n(x)\}$ satisfy a recurrence relation it is often best to use a related recurrence relation for the $\{u^{(n)}\}$.

The "optimum" polynomials $\{P_n(G)\}$ are scaled Chebyshev polynomials, and can be shown to satisfy a three-term recurrence relation, see e.g. Varga [1957], Blair et al [1959], and Golub and Varga [1961]. One can use this relation to derive the following form which is given by Hageman and Young [1981].

$$u^{(n+1)}=\rho_{n+1}\{\gamma(Gu^{(n)}+k)+(1-\gamma)u^{(n)}\}+(1-\rho_{n+1})u^{(n-1)} \tag{4.5}$$

where $\gamma, \rho_1, \rho_2, \ldots$ are given functions of the smallest eigenvalue and the largest eigenvalue of G. We say that (4.5) defines a (nonstationary) second-degree procedure.

Frankel [1950] considered a precedure for accelerating the convergence of the Richardson basic iterative method (4.3). This procedure, which he called the "second-order Richardson method" can be regarded as a stationary second-degree method applied to the basic iterative method. By a stationary second-degree method applied to the basic iterative method (4.1) we mean a method of the form

$$u^{(1)}=\hat{\gamma}(Gu^{(0)}+k)+(1-\hat{\gamma})u^{(0)} \tag{4.6}$$

$$u^{(n+1)}=\rho\{\gamma(Gu^{(n)}+k)+(1-\gamma)u^{(n)}\}+(1-\rho)u^{(n-1)}, n=1,2,\ldots$$

where $\hat{\gamma}, \rho$, and γ are fixed. It can be shown that for any choice of $\hat{\gamma}$, the optimum values, of ρ and γ are related to the optimum values of γ and $\rho_1, \rho_2, \ldots$ for the Chebyshev procedure (4.5). Thus γ is the same for both cases and $\rho_\infty = \lim_{n\to\infty}\rho_n$ Moreover, the asymptotic convergence rate is the same for (4.6) as it is for (4.5). These results are given by Frankel [1950] for Richardson's method. For the more general case see Golub [1959], Golub and Varga [1961], Young [1972], Young and Kincaid [1972] and Kincaid [1974].

In some cases there is a close relation between second-degree methods applied to the Jacobi method and certain generalizations of the SOR method. Thus, let us assume that the linear system (2.1) is "red-black", i.e. that it can be written in the form

$$\begin{pmatrix} D_R & H \\ K & D_B \end{pmatrix}\begin{pmatrix} u_R \\ u_B \end{pmatrix}=\begin{pmatrix} b_R \\ b_B \end{pmatrix} \tag{4.7}$$

where D_R and D_B are square diagonal matrices. We refer to the equations corresponding to D_R and D_B as the "red" equations and the "black" equations respectively. The modified SOR method (MSOR method) involves using the SOR method with relaxation factors $\omega_1, \hat{\omega_1}, \omega_2, \hat{\omega_2}, \ldots$ where ω_1 is used for the red equations on the first iteration, $\hat{\omega_1}$ is used for the black equations on the first iteration, etc.

Frequently-used choices of the $\{\omega_i\}$ and $\{\hat{\omega_i}\}$ for the MSOR method are, (see e.g. Young [1971], Chapter 10)

(1) The ordinary SOR method

$\omega_1=\hat{\omega_1}=\omega_2=\hat{\omega_2}=\ldots=\omega_b$ where $\omega_b=2/(1+\sqrt{1-S(B)^2})$.

(2) The modified Sheldon method

$\omega_1=1, \hat{\omega_1}=\omega_2=\hat{\omega_2}=\ldots=\omega_b$

(3) The Cyclic Chebyshev Semi-Iterative Method, see Varga [1962] and Golub and Varga [1961]

$\omega_1=1 \qquad \hat{\omega_1}=2(2-S(B)^2)^{-1}$

$\omega_{k+1}=(1-\frac{S(B)^2}{4}\omega_k)^{-1} \quad \hat{\omega}_{k+1}=(1-\frac{S(B)^2}{4}\omega_{k+1})^{-1}$

$(k=1,2,\ldots)$

If the second degree methods (4.5) and (4.6) are applied to the Jacobi method one obtains the sequence

$$\begin{bmatrix} u_R^{(0)} \\ u_B^{(0)} \end{bmatrix} \;\Big|\; \begin{bmatrix} u_R^{(1)} \\ u_B^{(1)} \end{bmatrix} \quad \begin{bmatrix} u_R^{(2)} \\ u_B^{(2)} \end{bmatrix} \;\Big|\; \begin{bmatrix} u_R^{(3)} \\ u_B^{(3)} \end{bmatrix} \quad \begin{bmatrix} u_R^{(4)} \\ u_B^{(4)} \end{bmatrix}$$

$\longleftarrow 1 \longrightarrow$ $\longleftarrow 2 \longrightarrow$

The subvectors in the boxes do not depend on the subvectors not in boxes. Moreover the vectors $(u_R^{(1)}u_B^{(2)})^T, (u_R^{(3)}u_B^{(4)})^T, \ldots$ correspond to the MSOR method. In particular Chebyshev acceleration corresponds to the Cyclic Chebyshev Semi-Iterative method, the case $\hat{\gamma}=\gamma$ corresponds to the modified Sheldon method and the case $\hat{\gamma}=\rho_\infty\gamma$ corresponds to the ordinary SOR method.

It is relatively easy to analyze the convergence properties of the second-degree method by the use of the appropriate polynomials; see e.g. Young and Kincaid [1972]. Such an analysis shows that Chebyshev acceleration is considerably faster than the stationary second-degree method with $\hat{\gamma}=\rho_\infty\gamma$. This helps explain why the Cyclic Chebyshev Semi-Iterative method is faster, as measured by the reduction of the norm of the error, than the ordinary SOR method. On the other hand, it should be noted that the spectral radius of the matrix corresponding to the ordinary SOR method is smaller than that of the Cyclic Chebyshev Semi-Iterative method.

The relations between the second-degree methods and the SOR method given above in some sense round out the theory. Such relations could

have been, and probably were, suspected by some researchers in the early 1950's. I must confess to suspecting some such relation when working on Richardson's method and noting the similarity between its asymptotic rate of convergence and that of the SOR method. I cannot remember when I received confirmation of these suspicions or from whom - it could have been when I heard a talk given in 1959 by Abe Taub, based on Blair et al [1959], when I read Golub's thesis [1959], or later when I read the paper by Golub and Varga [1961].

5. Extensions and Recent Developments

As stated earlier I had originally planned to cover a great deal more of the history of iterative methods. A number of topics which might have been included, given unlimited time and space, are listed below. Some of these will be reviewed in a later paper.

Extensions of the SOR method and theory

block SOR, SOR for p-cyclic matrices, generalized consistently ordered matrices, Kahan's generalization of the theory to the case where A is a Stieltjes matrix, the SSOR method

Other basic iterative methods

ADI methods, methods based on approximate factorization of matrices (PDE oriented and matrix oriented)

Conjugate gradient acceleration

the conjugate gradient method and the preconditioned conjugate gradient method

Adaptive parameter determination

automatic procedures for estimating upper and lower bounds for the eigenvalues of the basic iteration matrix and splitting parameters such as ω for the SSOR method

Methods for nonsymmetric systems

Complex Chebyshev, generalized conjugate gradient methods such as OROTHODIR, ORTHOMIN and ORTHORES, Lanczos methods, normal equations and generalized normal equations

General purpose software packages

The ITPACK, ELLPACK and PCG packages

Other methods

multigrid methods, methods for vector and parallel systems

References

Blair, A., Metropolis., von Neumann, J., Taub, A. H., and Tsingori, M. [1959], A study of a numerical solution of a two-dimensional hydrodynamical problem, Math. Tables Aids Comput. 13, 145-184.

Flanders, D., and Shortley, G. [1950], Numerical determination of fundamental modes, J. Appl. Phys. 21, 1326-1332.

Frankel, S. P. [1950], Convergence rates of iterative treatments of partial differential equations, Math. Tables Aids Comput. 4, 65-75.

Geiringer, H. [1949], On the solution of systems of linear equations by certain iterative methods, Reissner Anniversary Volume, Contributions to Applied Mechanics, 365-393. Edwards, Ann Arbor, Michigan.

Golub, G. H. [1959], The use of Chebyshev matrix polynomials in the iterative solution of linear systems compared with the method of successive overrelaxation, doctoral thesis, University of Illinois, Urbana, Illinois.

Golub, G. H., and Varga, R. S. [1961], Chebyshev semi-iterative methods, and second-order Richardson iterative methods. Numer. Math., Parts I and II, 3, 147-168.

Hageman, L. A. and Young, D. M. [1981], Applied Iterative Methods, Academic Press, New York.

Kincaid, D. R. [1974] On complex second-degree iterative methods, Siam J. Numerical Analysis, 11, 211-218.

Lanczos, C. [1952], Solution of systems of linear equations by minimized iterations, J. Res. Nat. Bur. of Standards 49, 33-53.

Liebmann, H. [1918], Die Angenährte Ermittleung harmonischer Functionen and Konformer Abbildungen, Sitz-Bayer. Akad. Wiss. Math.-Phys. Klasse, 385-416.

Markoff, W. [1891], Über Polynome, die in einem gegeben Intervalle moglichst wenig von Null abweichen, Math. Ann. 77 (1916): 213-258 (translation and condensation by J. Grossman of Russian article published in 1892).

Richardson, L. F. [1910], The approximate arithmetical solution by finite differences of physical problems involving differential equations with an application to the stresses in a masonry dam, Philos. Trans. Roy. Soc. London Ser. A 210, 307-357.

Shortley, G. [1953], Use of Tschebyscheff polynomial operators in the numerical solution of boundary value problems, J. Appl. Phys. 24, 392-296.

Shortley, G. H., Weller, Royal, and Fried, Bernard [1940], Numerical solution of Laplace's and Poisson's equations, The Engineering Experiment Station Bulletin No. 107, vol. IX, No. 5, revised January 1942.

Snyder, F. and Livingston, H., [1949], Coding of a Laplace boundary value problem for the UNIVAC, Math. Tables and Other Aids to Computation, III, 341-350.

Southwell, R. V., [1946], Relaxation Methods in Theoretical Physics, Oxford University Press.

Temple, G. [1938], The general theory of relaxation methods applied to linear systems, Proc. Roy. Soc. A 169, 476-500.

Varga, R. S. [1957], A comparison of the successive overrelaxation method and semi-iterative methods using Chebyshev polynomials, J. Soc. Indus. Appl. Math. 5, 39-46.

Varga, R. S. [1962], Matrix Iterative Analysis. Prentice-Hall, Englewood Cliffs, New Jersey.

Young, D. M. [1950], Iterative methods for solving partial difference equations of elliptic type, doctoral thesis, Harvard Univ., Cambridge, Massachusetts.

Young, D. M. [1954], Iterative methods for solving partial difference equations of elliptic type. Trans. Amer. Math. Soc. 76, 92-111.

Young, D. M. [1954a], On Richardson's method for solving linear systems with positive definite matrices, J. Math. Phys. XXXII, 243-255.

Young, D. M. [1956], On the solution of linear systems by iteration, Proc Sixth Symp. in Appl. Math. Amer. Math Soc. vol VI, 283-298, McGraw-Hill, New York

Young, D. M. [1971]. Iterative Solution of Large Linear Systems. Academic Press, New York.

Young, D. M. [1972], Second-degree iterative methods for the solution of large linear systems, J. of Approx Theory, 5, 137-148.

Young, David M., and Kincaid, David R., [1972], Linear stationary second-degree methods for the solution of large linear systems: Report CNA-52, Center for Numerical Analysis, The University of Texas, Austin, Texas.

Young, David M., and Francis Lerch [1953], The numerical solution of Laplace's equation on ORDVAC, Ballistic Research Laboratories Memorandum Report No. 708, Aberdeen Proving Grounds, Md.

Young, D. M., and Warlick, C. H. [1953], On the use of Richardson's method for the numerical solution of Laplace's equation on the ORDVAC, Ballistic Research Labs. Memorandum Report No. 707, Aberdeen Proving Grounds, Maryland.

HISTORICAL COMMENTS ON FINITE ELEMENTS

J. Tinsley Oden

The University of Texas at Austin

1. INTRODUCTION

Finite elements; perhaps no other family of approximation methods has had a greater impact on the theory and practice of numerical methods during the twentieth century. Finite element methods have now been used in virtually every conceivable area of engineering that can make use of models of nature characterized by partial differential equations.

Why have finite element methods been so popular in both the engineering and mathematical community? I feel that a principal reason for the success and popularity of these methods is that they are based on the weak, variational, formulation of boundary and initial value problems. This is a critial property, not only because it provides a proper setting for the existence of very irregular solutions to differential equations (e.g. distributions), but also because the solution appears in the integral of a quantity over a domain. The simple fact that the integral of a measurable function over an arbitrary domain can be broken up into the sum of integrals over an arbitrary collection of almost disjoint subdomains whose union is the original domain, is a vital property. Because of it, the analysis of a problem can literally be made locally, over a typical subdomain, and by making the subdomain sufficiently small one can argue that polynomial functions of various degrees are adequate for representing the local behavior of the solution. This summability of integrals is exploited in every finite element program. It allows the analysts to focus their attention on a typical finite element domain and to develop an approximation independent of the ultimate location of that element in the final mesh.

The simple integral property also has important implications in physics and in most problems in continuum mechanics. Indeed, the classical balance laws of mechanics are global, in the sense that they are integral laws applying to a given mass of material, a fluid or solid. From the onset, only regularity of the primitive variables sufficient for these global conservation laws to make sense is needed. Moreover, since these laws are supposed to be fundamental axioms of physics, they must hold over every finite portion of the material: every finite element of the continuum. Thus once again, one is encouraged to think of approximate methods defined by integral formulations over typical pieces of a continuum to be studied.

2. THE ORIGIN OF FINITE ELEMENTS

When did finite elements begin? It is difficult to trace the origins of finite element methods because of a basic problem in defining precisely what constitutes a "finite element method". To most mathematicians, it is a method of piecewise polynomial approximation and, therefore, its origins are frequently traced to the appendix of a paper by COURANT [1943] in which piecewise linear approximations of the Dirichlet problem over a network of triangles is discussed. Also, the "interpretation of finite differences" by POLYA [1952] is regarded as embodying piecewiese-polynomial approximation aspects of finite elements.

On the other hand, the approximation of variational problems on a mesh of triangles goes back much further: 92 years. In 1851, SCHELLBACH [1851] proposed a finite-element-like solution to Plateau's problem of determining the surface S of minimum area enclosed by a given closed curve. SCHELLBACH used an approximation S_h of S by a mesh of triangles over which the surface was represented by piecewise linear functions, and he then obtained an approximation of the solution to Plateau's problem by minimizing S_h with respect to the coordinates of hexagons formed by six elements (see WILLIAMSON [1980]). Not quite the conventional finite element approach, but certainly as much a finite element technique as that of COURANT.

Some say that there is even an earlier work that uses some of the ideas underlying finite element methods: LEIBNIZ himself employed a piecewise linear approximation of the Brachistochrone problem proposed by BERNOULLI in 1696 (see the historical volume, LEIBNIZ [1962]). With the help of his newly developed calculus tools, LEIBNIZ derived the governing differential equation for the problem, the solution of which is a cycloid. However, most would agree that to credit this work as a finite element approximation is somewhat stretching the point. LEIBNIZ had no intention of approximating a differential equaiton; rather, his purpose was to derive one. Two and a half centuries later it was realized that useful approximations of differential equations could be determined by not necessarily taking infinitesimal elements as in the calculus, but by keeping the elements finite in size. This idea is, in fact, the basis of the term "finite elements".

There is also some difference in the process of laying a mesh of triangles over a domain on the one hand and generating the domain of approximation by piecing together triangles on the other. While these processes may look the same in some cases, they may differ dramatically in how the boundary conditions are imposed. Thus, neither SCHELLBACH nor COURANT, nor for that matter SYNGE who used triangular meshes many years later, were

particularly careful as to how boundary conditions were to be imposed or as to how the boundary of the domain was to be modeled by elements, issues that are now recognized as an important feature of finite element methodologies. If a finite element method is one in which a global approximation of a partial differential equation is built up from a sequence of local approximations over subdomains, then credit must go back to the early papers of HRENNIKOFF [1941], and perhaps beyond, who chose to solve plane elasticity problems by breaking up the domain of the displacements into little finite pieces, over which the stiffnesses were approximated using bars, beams, and spring elements. A similar "lattice analogy" was used by McHENRY [1943]. While these works are draped in the most primitive physical terms, it is nevertheless clear that the methods involve some sort of crude piecewise linear or piecewise cubic approximation over rectangular cells. Miraculously, the methods also seem to be convergent.

To the average practitioner who uses them, finite elements are much more than a method of piecewise polynomial approximation. The whole process of partitioning of domain, assembling elements, applying loads and boundary conditions, and, of course, along with it, local polynomial approximation, are all components of the finite element method.

If this is so, then one must acknowledge the early papers of GABRIEL KRON who developed his "tensor analysis of networks" in 1939 and applied his "method of tearing" and "network analysis" to the generation of global systems from large numbers of individual components in the 1940's and 1950's (KRON [1939]; see also KRON [1953], [1955]). Of course, KRON never necessarily regarded his method as one of approximating partial differential equations; rather, the properties of each component were regarded as exactly specified, and the issue was an algebraic one of connecting them all appropriately together.

In the early 1950's, ARGYRIS [1954] began to put these ideas together into what some call a primitive finite element method: he extended and generalized the combinatoric method of KRON and other ideas that were being developed in the literature on system theory at the time, and added to it variational methods of approximation, a fundamental step toward true finite element methodology.

Around the same time, SYNGE [1956] described his "method of the hypercircle" in which he also spoke of piecewise linear approximations on triangular meshes, but not in a rich variational setting and not in a way in which approximations were built by either partitioning a domain into triangles or assembling triangles to approximate a domain (indeed Synge's treatment of boundary conditons was clearly not in the spirit of finite elements, even though he was keenly aware of the importance of convergence criteria and of the "angle condition" for triangles, later studied in some depth by others).

It must be noted that during the mid-1950's there was a number of independent studies underway which made use of "matrix methods" for the analysis of aircraft structures. A principal contributor to this methodology was LEVY [1953] who introduced the "direct stiffness method" wherein he approximated the structural behavior of aircraft wings nsing assemblies of box beams, torsion boxes, rods and shear panels. These assuredly represent some sort of crude local polynomial approximation in the same spirit as the HRENNIKOFF and McHENRY approaches. The direct stiffness method of LEVY had a great impact on the structural analysis of aircraft, and aircraft companies throughout the United States began to adopt and apply some variant of this method or of the methods of ARGYRIS to complex aircraft structural analyses. During this same period, similar structural analysis methods were being developed and used in Europe, particularly in England, and one must mention in this regard the work of TAIG [1961] in which shear lag in aircraft wing panels was approximated using basically a bilinear finite element method of approximation. Similar element-like approximations were used in many aircraft industries as components in various matrix-methods of structural analyses. Thus the precedent was established for piecewise approximations of some kind by the mid-1950's.

To a large segment of the engineering community, the work representing the beginning of finite elements was that contained in the pioneering paper of TURNER, CLOUGH, MARTIN, and TOPP [1956] in which a genuine attempt was made at both a local approximation (of the partial differential equations of linear elasticity) and the use of assembly strategies essential to finite element methodology. It is interesting that in this paper local element properties were derived without the use of variational principles. It was not until 1960 that CLOUGH [1960] actually dubbed these techniques as "finite element methods" in a landmark paper on the analysis of linear plane elasticity problems.

The 1960's were the formative years of finite element methods. Once it was perceived by the engineering community that useful finite element methods could be derived from variational principles, variationally based methods significantly dominated all the literature for almost a decade. If an operator was unsymmetric, it was thought that the solution of the associated problem was beyond the scope of finite elements, since it did not lend itself to a traditional extremum variational approximation in the spirit of RAYLEIGH and RITZ.

Many workers in the field feel that the famous Dayton conferences on finite elements (at the Air Force Flight Dynamics Laboratory in Dayton, Ohio, U.S.A.) represented landmarks in the development of the field (see PRZEMINIECKI et al. [1966]). Held in 1965, 1968, 1970, these meetings brought specialists from all over the world to discuss their latest triumphs and failures, and the pages of the proceedings, particularly the earlier volumes, were filled with remarkable and innovative accomplishments from a technical community just beginning to learn the richness and power of this new collection of ideas. In these volumes one can find many of the premier papers of now well-known methods. In the first volume alone one can find mixed finite element methods (HERRMANN [1966], Hermite approximations (PESTEL [1966]), C^1-cubic approximations (BOGNER, FOX, and SCHMIT [1966]) hybrid methods (PIAN [1966]) and other contributions. In later volumes, further assaults on nonlinear problems and special element formulations can be found.

Near the end of the sixties and earlv seventies there finally emerged the realization that the method could be applied to unsymmetric operators without difficulty and thus problems in fluid mechanics were brought within the realm of application of finite element methods; in particular, finite element models of the full Navier-Stokes equations were first presented during this period (ODEN [1969], ODEN and SOMOGYI [1969], ODEN [1970]).

The early textbook by ZIENKIEWICZ and CHANG [1967] did much to popularize the method with the practicing engineering community. However, the most important factor leading to the rise in popularity during the late 1960's and early 1970's was not purely the publication of special formulations and algorithms, but the fact that the method was being very successfully used to solve difficult engineering problems. Much of the techology used during this period was due to BRUCE IRONS, who with his colleagues and students developed a multitude of techniques for the successful implementation of finite elements. These included frontal solution technique (IRONS [1970]), the patch test (IRONS and RAZZAQUE [1972]), isoparametric elements

(ERGATOUDIS, IRONS and ZIENKIEWICZ [1966]), and numerical integration schemes (IRONS [1966]) and many more. The scope of finite element applications in the 1970's would have been significantly diminished without these contributions.

3. THE MATHEMATICAL THEORY

The mathematical theory of finite elements was slow to emerge from this caldron of activity. Many of the works on "variational finite difference methods" which appeared in the mid-to-late 1960's actually captured the essence of convergence requirements of finite element methods. Thus, the 1965 work of FENG KANG [1965] on such methods, published in Chinese and unknown to the western world for over a decade, is regarded by many as containing the first proof of convergence of finite-element methods. The mathematical theory of finite elements, which addressed mathematical issues connected with purely finite element schemes, began around 1968 and several papers were published that year on the subject. One of the first papers in this period to address the problem of convergence of the finite method in a rigorous way and in which a-priori error estimates for bilinear approximations of a problem in a plane elasticity are obtained, is the often overlooked paper of JOHNSON and McCLAY [1968], which appeared in the *Journal of Applied Mechanics*. This paper correctly developed error estimates in energy norms, and even atempted to characterize the deterioration of convergence rates due to corner singularities.

Also in 1968 there appeared the important mathematical paper of ZLÁMAL [1968] in which a detailed analysis of interpolation properties of a class of triangular elements and their application to second-order and fourth-order linear elliptic boundary-value problems is discussed. This paper attracted the interest of a large segment of the numerical analysis community and several very good mathematicians began to work on finite element methodologies. In the same year, CIARLET [1968] published a rigorous proof of convergence of a finite element approximation of a class of linear two-point boundary-value problems in which piecewise linear shape functions were used. By using a discrete maximum principle he was able to prove L^∞ estimates. We also mention the work of OLIVEIRA [1968] on convergence of finite element methods which established corrected rates-of-convergence of certain problems in appropriate energy norms.

By 1972, finite element methods had emerged as an important new area of numerical analysis in applied mathematics. Mathematical conferences were held on the subject on a regular basis, and there began to emerge a rich volume of literature on mathematical aspects of the method applied to elliptic problems, eigenvalue problems, and parabolic problems. A conference of special significance in this period was held at the University of Maryland in 1972 and featured a penetrating series of lectures by IVO BABUŠKA (see BABUŠKA and AZIZ [1973]) and several important mathematical papers by leading specialists in the mathematics of finite elements, all collected in the volume edited by AZIZ [1972].

One unfamiliar with aspects of the history of finite elements may be led to the erroneous conclusion that the method of finite elements emerged from the growing wealth of information on partial differential equations, weak solutions of boundary-value problems, Sobolev spaces, and the associated approximation theory for elliptic variational boundary-value problems. This is a natural mistake, because the seeds for the modern theory of partial differential equations were sown about the same time as those for the development of modern finite element methods, but in an entirely different garden.

In the late 1940's, LAURENT SCHWARTZ was putting together his theory of distributions around a decade after the notion of generalized functions and their use in partial differential equations appeared in the pioneering work of SOBOLEV. A long list of other names could be added to the list of contributors to the modern theory of partial differential equations, but that is not our purpose here. Rather, we must only note that the rich mathematical theory of partial differential equations which began in the 1940's and 50's, blossomed in the 1960's, and is now an integral part of the foundations of not only partial differential equations but also approximation theory, did not lead naturally to the variational methods of approximation such as finite elements, but grew independently and parallel to that development for almost two decades. It was a happy accident, or perhaps an unavoidable occurrence, that in the late 1960's these two independent subjects, finite element methodology and the theory of approximation of partial differential equations via functional analysis methods, united in an inseparable way, so much so that it is difficult to appreciate the fact that they were ever separate.

The 1970's must mark the decade of the mathematics of finite elements. During this period, great strides were made in determining a-priori error estimates for a variety of finite element methods, for linear elliptic boundary-value problems, for eigenvalue problems, and certain classes of linear and nonlinear parabolic problems; also, some preliminary work on finite element applications to hyperbolic equations was done. It is both inappropriate and perhaps impossible to provide an adequate survey of this large volume of literature, but it is possible to present an albeit biased reference to some of the major works along the way.

An important component in the theory of finite elements is an interpolation theory: how well can a given finite element method approximate functions of a given class locally over a typical finite element? A great deal was known about this subject from the literature on approximation theory and spline analysis, but its particularization to finite elements involves technical difficulties. One can find results on finite element interpolation in a number of early papers, including those of ZLÁMAL [1968], BRAMBLE and ZLÁMAL [1970], BABUŠKA [1970, 1971], and BABUŠKA and AZIZ [1972]. But the elegant work on Lagrange and Hermite interpolations of finite elements by CIARLET and RAVIART [1972a] must stand as a very important contribution to this vital aspect of finite element theory. A landmark work on the mathematics of finite elements appeared in 1972 in the remarkably comprehensive and penetrating memoir of BABUŠKA and AZIZ [1972] on the mathematical foundations of finite element methods. Here one can find interwoven with the theory of Sobolev spaces and elliptic problems, general results on approximation theory that have direct bearing on finite element methods. The fundamental work of NITSCHE [1975] on L^∞ estimates for general classes of linear elliptic problems must stand out as one of the most important contributions of the seventies. STRANG [1972], in an important communication, pointed out "variational crimes", inherent in many finite element methods, such as improper numerical quadrature, the use of nonconforming elements, improper satisfaction of boundary conditions, etc., all common practices in applications, but all frequently leading to exceptable numerical schemes. In the same year, CIARLET and RAVIART [1972b,c] also contributed penetrating studies of these issues. Many of the advances of the 1970's drew upon earlier results on variational methods of approximation based on the Ritz method and finite differences; for example the fundamental Aubin-Nitsche method for lifting the order of convergence to

lower Sobolev norms (see AUBIN [1967] and NITSCHE [1968]) used such results. In 1974, the important paper of BREZZI [1974] used such earlier results on saddle-point problems and laid the groundwork for a multitude of papers on problems with constraints and on the stability of various finite element procedures. While convergence of special types of finite element strategies such as mixed methods and hybrid methods had been attempted in the early 1970's (e.g. ODEN [1973]), the BREZZI results, and the methods of BABUŠKA for constrained problems, provided a general framework for sudying virtually all mixed and hybrid finite elements (e.g. RAVIART [1975], RAVIART and THOMAS [1977], BABUSKA, ODEN and LEE [1977]).

The penetrating work of SCHATZ and WHALBIN [1976] on interior estimates and problems represented notable contributions to the growing mathematical theory of finite elements. The important work of DOUGLAS and DUPONT (e.g. [1970], [1973]; DUPONT [1973]) on finite element methods for parabolic problems and hyperbolic problems must be mentioned along with the idea of elliptic projections of WHEELER [1973] which provided a useful technique for deriving error bounds for time-dependent problems.

The 1970's also represented a decade in which the generality of finite element methods began to be appreciated over a large portion of the mathematics and scientific community and it was during this period that significant applications to highly nonlinear problems were made. The fact that very general nonlinear phenomena in continuum mechanics, including problems of finite deformation of solids and of flow of viscous fluids could be modeled by finite elements and solved on existing computers was demonstrated in the early seventies (e.g. ODEN [1972]), and, by the end of that decade, several "general purpose" finite element programs were in use by engineers to treat broad classes of nonlinar problems in solid mechanics and heat transfer. The mathematical theory for nonlinear problems also was advanced in this period, and the important work of FALK [1974] on finite element approximations of variational inequalities should be mentioned.

It is not too inaccurate to say that by 1980, a solid foundation for the mathematical theory of finite elements for linear problems had been established and that significant advances in both theory and application into nonlinear problems existed. The open questions that remain are difficult ones and their solution will require a good understanding of the mathematical properties of the method.

4. PERSONAL REFLECTIONS AND ACKNOWLEDGEMENTS

I remember very well my own introduction to finite elements. I had read thoroughly the work of AGYRIS and others on "matrix methods in structural mechanics" and had developed notes on the subject while teaching graduate courses in solid mechanics in the early 1960's, but none of the literature of the day had much impact on my university research at the time, if the research of anyone in the university community. The aircraft industry was actively developing the subject during this period and was far ahead of universities in studying and implementing these methods.

Then, in 1963, I had the good fortune to enter the aerospace industry for a brief period of time and to meet and begin joint work with GILBERT BEST, who had been charged with the responsibility of developing a large general-purpose finite element code for use in aircraft structural analysis. Only the two of us worked on the project, but by fall 1963 we had produced some quite general results and one of the early working codes on finite elements. This code had features in it that were not fully duplicated for more than a decade. I still have copies of our elaborate report on that work (BEST and ODEN [1963]).

It was BEST who demonstrated to me the strength and versatility of the method. In our work, noted above, we developed mixed methods, assumed stress methods, hybrid methods, we explored algorithms for optimization problems, nonlinear problems, bifurcation and vibration problems, and did detailed tests on stability and convergence of various methods by numerical experimentation. We developed finite elements for beams, plates, shells, for composite materials, for three-dimensional problems in elasticity, for thermal analysis, and linear dynamic analysis. Some of our methods were failures; most were effective and useful. Since convergence properties and criteria were not to come on the scene for another decade, our only way to test many of the more complex algorithms was to code them and compute solutions for test problems.

I went on to return to academia in 1964 and among my first chores was to develop a graduate course on finite element methods. At the same time, I taught mathematics and continuum mechanics, and it became clear to me that finite elements and electronic computing offered hope of transforming nonlinear continuum mechanics from a qualitative and academic subject into something useful in modern scientific computing and engineering. Toward this end, I began work with graduate students in 1965 that led to successful numerical analyses of problems in finite-strain elasticity (1965, 1966), elastoplasticity (1967), thermoelasticity (1967), thermoviscoelasticity (1969), and incompressible and compressible viscous fluid flow (1968, 1969). These works, many summarized in ODEN [1972], include early (perhaps the first) uses of Discrete-Kirchhoff elements, incremental elasto-plastic algorithms, conjugate-gradient methods for nonlinear finite element systems, continuation methods, dynamic relaxation schemes, Taylor-Galerkin algorithms (then called "finite-element based Lax-Wendroff schemes"), primitive-variable formulations in incompresible flow, curvilinear elements, and penalty formulations; all these subjects have been resurected in more recent times and have been studied in far more detail and better style and depth than was possible in the 1960's.

While my later work, work in the '70's and '80's, was influenced by the competent mathematicians (and friends) who developed the subject during the period (BABUŠKA, CIARLET, STRANG, DOUGLAS, NITSCHE, and many others), the work and guidance of G. BEST was basic to may interest in this subject, and I dedicate this note to him.

I should also add that portions of this paper are excerpts from an article to appear in the Handbook of Numerical Analysis, edited by J.L. LIONS and P.G. CIARLET, North Holland Publishing Co., Amsterdam. I am grateful to North Holland for granting permission to use this material in the present volume.

SELECTED BIBLIOGRAPHY

ARGYRIS, J.H. [1954]: "Energy Theorems and Structural Analysis", *Aircraft Engineering*, Vol. 26, pp. 347-356 (Oct.), 383-387, 394 (Nov.)

ARGYRIS, J.H. [1955]: "Energy Theorems and Structural Analysis", *Aircraft Engineering*, Vol. 27, pp. 42-58 (Feb.), 80-94 (March), 125-134 (April), 145-158 (May)

ARGYRIS, J.H. [1966]: "Continua and Discontinua", *Proceedings, Conference on Matrix Methods in Structural Mechanics*, Preziemiencki et al. (Eds.), AFFDL-TR-66-80, (Oct. 26-28, 1965), Wright-Patterson AFB, Ohio, pp. 11-190.

AUBIN, J.P. [1967]: "Behavior of the Error of the Approximate Solutions of Boundary-Value Problems for Linear Elliptic Operators by Galerkin's Method and Finite Differences", *Annali della Scuola Normale di Pisa*, Series 3, Vol. 21, pp. 599-637.

AZIZ, A.K. [1972]: Editor of **The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations**, Academic Press, N.Y.

BABUŠKA, I, J.T. ODEN, and J.K. LEE [1977]: "Mixed-Hybrid Finite Element Approximations of Second-Order Elliptic Boundary-Value Problems", *Computer Methods in Applied Mechanics and Engineering*, Vol. 11, pp. 175-206.

BABUŠKA, I. [1976]: "Finite Element Methods for Domains with Corners", *Computing*, Vol. 6, pp. 264-273.

BABUŠKA, I., and A.K. AZIZ [1972]: "Survey Lectures on the Mathematical Foundation of the Finite Element Method", Aziz, A.K. Ed., **The Mathematical Foundation of the Finite Element Method with Applications to Partial Differential Equations**, Academic Press, N.Y., pp. 5-359.

BABUŠKA, I. [1971]: "Error Bounds for the Finite Element Method", *Numerische Math.*, Vol. 16, pp. 322-333.

BEST, G., and J.T. ODEN [1963]: "Stiffness Matrices for Shell-Type Structures", *Engineering Research Report No. 233*, General Dynamics, Ft. Worth, Texas.

BRAMBLE, J.H., and M.ZLAMAL [1970]: "Triangular Elements in the Finite Element Method", *Mathematics of Computation*, Vol. 24, No. 112, pp. 809-820.

BREZZI, F. [1974]: "On the Existence, Uniqueness, and Approximation of Saddle-Point Problems Arising from Lagrange Multipliers", *Revue Française d'Automatique, Informatique et Recherche Operationelle*, 8-R2, pp. 129-151.

BOGNER, F.K., R.L. FOX, and L.A. SCHMIT, Jr. [1966]: "The Generation of Interelement, Compatible Stiffness and Mass Matrices by the Use of Interpolation Formulas", *Proceedings, Conference on Matrix Methods in Structural Mechanics*, Przemieniecki et al. (Eds.), pp. 397-444.

CIARLET, P.G. [1968]: "An $O(h^2)$ Method for a Non-Smooth Boundary-Value Problem, *Aequationes Math.*, Vol. 2, pp. 39-49.

CIARLET, P.G., and P.A. RAVIART [1972a]: "General Lagrange and Hermite Interpolation in $\mathbb{R}^n$ with Applications to the Finite Element Method", *Archive for Rational Mechanics and Analysis*, Vol. 46, pp. 177-199.

CIARLET, P.G., and P.A. RAVIART [1972b]: "Interpolation Theory over Curved Elements with Applications to Finite Element Methods", *Computer Methods in Applied Mechanics and Engineering*, pp. 217-249.

CIARLET, P.G., and P.A. RAVIART [1972c]: "The Combined Effect of Curved Boundaries and Nuermical Integration in Isoparametric Finite Element Methods", in **The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations**, Ed. A.K. Aziz, Academic Press, N.Y., pp. 409-474.

CLOUGH, R.W. [1960]: "The Finite Element Method in Plane Stress Analysis", *Proceedings 2nd ASCE Conference on Electronic Computation.*

COURANT, R. [1943]: "Variational Methods for the Solution of Problems of Equilibrium and Vibration", *Bull. Am. Math. Soc.*, Vol. 49, 1-23.

DOUGLAS, J. and T. DUPONT [1973]: "Superconvergence for Galerkin Methods for the Two- Point Boundary Problem via Local Projections", *Numerische Math.*, Vol. 21, pp. 220-228.

DOUGLAS, J. and T. DUPONT [1970]: "Galerkin Methods for Parabolic Problems", *SIAM J. Numerical Analysis*, Vol. 7, No. 4, pp. 575-626.

DUPONT. T. [1973]: "L^2-Estimates for Galerkin Methods for Second-Order Hyperbolic Equations, *SIAM J. Numerical Analysis*, Vol. 10, pp. 880-889.

ERGATOUDIS, I, B.M. IRONS, and O.C. ZIENKIEWICZ [1966]: "Curved Isoparametric Quadrilateral Finite Elements", *Int. J. Solids and Structures*, Vol. 4, pp. 31-42.

FALK, S.R. [1974]: "Error Estimates for the Approximation of a Class of Variational Inequalities", *Mathematics of Computation*, Vol. 28, pp. 963-971.

FENG KANG [1965]: "A Difference Formulation Based on the Variational Principle" (in Chinese), Appl. Mathematics and Comp. Mathematics, Vol. 2, No. 4, pp. 238-262.

HERRMANN, L.R. [1966]: "A Bending Analysis for Plates" Proceedings, *Conference on Matrix Methods in Structural Mechanics*, Przemieniecki et al. (Eds.), pp. 577.

HRENNIKOFF, H. [1941]: "Solutions of Problems in Elasticity by the Framework Method", *J. Appl. Mech.*, A 169-175.

IRONS, B., and A. RAZZAQUE [1972]: "Experience with the Patch Test for Convergence of Finite Elements", in **The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations**, Ed. A.K. Aziz, Academic Press, N.Y., pp. 557-587.

IRONS, B. [1970]: "A Frontal Solution Program for Finite Element Analysis", *Int. J. Num. Meth's. Eng.*, Vol. 2, No. 1, pp. 5-32.

IRONS, B. [1966]: "Engineering Applications of Numerical Integration in Stiffness Methods", *AIAA Journal*, Vol. 4, No. II, pp. 2035-3037.

JOHNSON, M.W., Jr., and McLay, R.W. [1968]: "Convergence of the Finite Element Method in the Theory of Elasticity", *J. Appl. Mech.*, Series E, Vol. 3, 5, No. 2, pp. 274-278.

KRON, G. [1953]: "A Set of Principles to Interconnect the Solutions of Physical Systems", *J. Appl. Phys.*, 24, 965-980.

KRON, G. [1939]: **Tensor Analysis of Networks**, John Wiley and Sons, New York.

LEIBNIZ, G. [1962]: **G.W. Leibniz Mathematische Schriften**, Ed. C. Gerhardt, pp. 290-293, G. Olms Verlagsbuchhandlung.

LEVY, S. [1953]: "Structural Analysis and Influence Coefficients for Delta Wings", *J. Aeronautical Sc.*, Vol. 20.

McHENRY, D. [1943]: "A Lattice Analogy for the Solution of Plane Stress Problems, *J. Inst. Civ. Eng.*, 21, 59-82.

NITSCHE, J.A. [1970]: "Lineare Spline-Funktionen und die Methoden von Ritz für Elliptische Randwertprobleme", *Archive for Rational Mechanics and Analysis*, Vol. 36, pp. 348-355.

NITSCHE, J.A. [1963]: "Ein Kriterium für die Quasi-Optimalität des Ritzschen Verfahrens",

Numerische Math., Vol. II, pp. 346-348.

ODEN, J.T. [1976]: **Finite Elements of Nonlinear Continua**, McGraw Hill Book Co., New York

ODEN, J.T. [1972]: "Some Contibutions to the Mathematical Theory of Mixed Finite Element Approximations", in **Theory and Practice in Finite Element Structural Analysis**, Yamada, Y. et al. Eds., Univ. of Tokyo Press, Tokyo, pp. 3-23.

ODEN, J.T. [1970]: "A Finite Element Analogue of the Navier-Stokes Equations", *J. Eng. Mech. Div.*, ASCE, Vol. 96, No. EM 4.

ODEN, J.T. [1969]: "A General Theory of Finite Elements; II. Applications", *Int. J. Num. Meth's. Eng.*, Vol. 1, No. 3, pp. 247-259.

ODEN, J.T., and SOMOGYI, D. [1968]: "Finite Element Applications in Fluid Dynamics", *J. Eng. Mech. Div.*, ASCE, Vol. 95, No. EM 4, pp. 821-826.

OLIVEIRA, Arantes E. [1969]: "Theoretical Foundation of the Finite Element Method", Int. J. Solids and Structures", Vol. 4, pp. 926-952.

PESTEL, E. [1966]: "Dynamic Stiffness Matrix Formulation by Means of Hermitian Polynomials", Proceedings, *Conference on Matrix Methods in Structural Mechanics*, Przemieniecki et al. (Eds.), pp. 479-502.

PIAN, T.H.H. [1966]: "Element Stiffness Matrices for Boundary Compatibility and for Prescribed Stresses, Proceedings, *Conference on Matrix Methods in Structural Mechanics*, Przemieniecki et al. (Eds.), pp. 455-478.

POLYA.G. [1952]: " Sur une Interprétation de la Méthode des Différences Finies qui Peut Fournir des Bornes Supérieures ou Inférieures", *Compt. Rend.*, 235, 995.

PRZEMIENIECKI, J.S., R.M. BADER, W.F., BOZICH, J.R. JOHNSON, and W.J. MYKYTOW (Eds.) [1966]: Proceedings, *Conference on Matrix Methods in Structural Mechanics*, AFFDL-TR-66-80, (Oct. 26-28, 1965), Wright-Patterson AFB, Ohio.

RAVIART, P.A., and THOMAS, J.M. [1977]: "A Mixed Finite Element Method for 2nd-Order Elliptic Problems", *Proceedings, Symposium on the Mathematical Aspects of the Finite Element Methods.*

RAVIART, P.A. [1975]: "Hybrid Methods for Solving 2nd-Order Elliptic Problems", in **Topics in Numerical Analysis**, Miller, J.H.H. (Ed.), Academic Press, N.Y., pp. 141-155.

SCHATZ, A.H., and L.B. WAHLBIN [1978]: "Maximum Norm Estimates in the Finite Element Method on Polygonal Domains, Part I", *Math. Comput.*, Vol. 32, No. 114, pp. 73-109.

SCHELLBACH, K. [1851]: "Probleme der Variationsrechnung", *J. Reine Angew. Math.*, 41, 293-363.

SYNGE, J.L. [1957]: **The Hypercircle Method in Mathematical Physics**, Cambridge Univ. Press, Cambridge.

STRANG, G. [1972]: "Variational Crimes in the Finite Element Method", in **The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations**, Ed. A.K. Aziz, Academic Press, N.Y.

TAIG, I.C. [1961]: "Structural Analysis by the Matrix Displacement Method", *English Electrical Aviation Ltd. Report*, S-O-17.

TURNER, M.J., CLOUGH, R.W., MARTIN, H.C., TOPP, L.J. [1956]: "Stiffness and Deflection Analysis of Complex Structures", *J. Aer. Sci.*, 805.

WHEELER, M.F. [1973]: "A-Priori L^2-Error Estimates for Galerkin Approximations to Parabolic Partial Differential Equations", SIAM J. Num. Analysis, Vol. II, No. 4, pp. 723-759.

WILLIAMSON, F. [1980]: "A Historical Note on the Finite Element Method", *Int. J. Num. Meth's. Eng.*, 15, 930-934.

ZLÀMAL, M. [1968]: "On the Finite Element Method", *Numerische Math.*, Vol. 12, pp. 394-409.

Shaping the Evolution of Numerical Analysis in the Computer Age—The SIAM Thrust

I. E. Block
SIAM

I intend to trace the development of SIAM, from its inception to 1963 when we introduced SIAM Journal on Numerical Analysis.

No Paper

THE CONTRIBUTION OF J. H. WILKINSON TO NUMERICAL ANALYSIS

B. N. Parlett*

Mathematics Department and
Computer Science Division of EECS Department
University of California
Berkeley, California 94720

1. An Outline of His Career

James Hardy Wilkinson died suddenly at his London home on October 5, 1986, at the age of 67. Here is a very brief account of his professional life.

He won an open competition scholarship in mathematics to Trinity College, Cambridge, when he was 16 years old. He won two coveted prizes (the Pemberton and the Mathieson) while he was an undergraduate at Trinity College and graduated with first class honors before he was 20 years old.

He worked as a mathematician for the Ministry of Supply throughout World War II and it was there that he met and married his wife Heather. In 1947 he joined the recently formed group of numerical analysts at the National Physical Laboratory in Bushy Park on the outskirts of London. He was to stay there until his retirement in 1980. Soon after his arrival he began to work with Alan Turing on the design of a digital computer. That work led to the pilot (prototype) machine ACE which executed its first scientific calculations in 1953. Wilkinson designed the multiplication unit for ACE and its successor DEUCE.

One could say that the decade 1947-1957 was the exciting *learning* period in which Wilkinson, and his colleagues at NPL, discovered how automatic computation differed from human computation assisted by desk top calculating machines. By dint of trying every method that they could think of and watching the progress of their computations on punched cards, paper tape, or even lights on the control console, these pioneers won an invaluable practical understanding of how algorithms behave when implemented on computers.

Some algorithms that are guaranteed to deliver the solution after a fixed number of primitive arithmetic operations IN EXACT ARITHMETIC can produce, on some problems, completely wrong yet plausible output on a digital computer. That is the fundamental challenge of the branch of numerical analysis that Wilkinson helped to develop.

*The author gratefully acknowledges partial support from Office of Naval Research Contract ONR N00014-85-K-0180.

The period 1958-1973 saw the development, articulation, and dissemination of this understanding of dense matrix computations. It was in 1958 that Wilkinson began giving short courses at the University of Michigan Summer College of Engineering. The notes served as the preliminary versions of his first two books. The lectures themselves introduced his work to an audience broader than the small group of specialists who had been brought together in 1957 by Wallace Givens at Wayne State University, Michigan, for the first of a sequence of workshops, that came to be called the Gatlinburg meetings. These conferences are discussed in more detail in the chapter by R.S. Varga. The year 1973 saw the beginning of the NATS project (at Argonne National Laboratory, USA) whose goal was to translate into FORTRAN, and test even further, the ALGOL algorithms collected in the celebrated Handbook of 1971. That book, essentially by Wilkinson and Reinsch, embodied most of what had been learnt about matrix transformations. There is more on this topic below.

By 1973 Wilkinson had received the most illustrious awards of his career. He was elected to the Royal Society of London in 1969. In 1970 he was awarded both the A. M. Turing award of the Association for Computing Machinery and the John von Neumann award of the Society for Industrial and Applied Mathematics. Both these professional groups are in the USA. It was not until 1977 that he was made an honorary fellow of the (British) Institute for Mathematics and its Applications.

The final period, 1974-1986, may be marked by Wilkinson's promotion to the Council of the Royal Society. Indeed he served as secretary for the physical sciences section for two or three years and these duties absorbed much of his energy. When that obligation was discharged he accepted a professorship in the Computer Science department at Stanford University, California, (1977-1984) but he was only in residence for the Winter quarter and not every year was he able to take up his position. His research now focused on more advanced, but less urgent numerical tasks such as computing the Jordan form, Kronecker's form for matrix pencils, and various condition numbers. During the last four years of his life he was absorbed in the atill open problem of how to determine the closest defective matrix to any given square matrix. He also gave much attention to the

task of explaining to the wider mathematical community the nature of the subject with which his name is indissolubly linked: roundoff error analysis. We will say more on this expository problem below.

2. Background

People are awed at the prodigious speed at which the digital computers of the 1980's can execute primitive arithmetic operations; sometimes millions of them per second. Yet this speed is achieved at a price; almost every answer is wrong. When two 14 decimal digit numbers are multiplied together, only the leading 14 digits are retained, the remaining 13 or 14 digits are discarded forever. If such a cavalier attitude to accuracy were to make nonsense of all our calculations then the prodigious speeds would be pointless. Moreover it requires little experience to discover how easily a digital computer can produce meaningless numbers.

Fortunately there are procedures that can survive these arithmetic errors and produce output that has adequate accuracy. Consequently computers can be useful. The difficult task is to discern the robust algorithms. A poor implementation can undermine a sound mathematical procedure and this simple fact has extensive and unpleasant consequences. It suggests that clean, general statements about the properties of numerical methods may not always be possible. Here is an example: will the process of iterative refinement improve the accuracy of the output of a good implementation of Gauss elimination on an ill-conditioned system of linear equations? The answer turns out to depend on whether certain intermediate quantities are computed with extra care. Considerations of this sort make it difficult to present the results of an error analysis and Wilkinson became more and more concerned with this problem.

Before embarking on a list of Wilkinson's contributions, five points must be emphasized.

1) Only a minority of numerical analysts pay attention to roundoff error. For example, in his influential book ***Matrix Iterative Analysis*** (Prentice-Hall, 1962), R. S. Varga mentions during the introduction that he will not be considering the effects of roundoff-error. Virtually every publication concerned with the approximate solution of differential equations invokes exact arithmetic. The tacit assumption is that the approximation errors are so much greater than the effect of roundoff that the latter may be ignored without risk.

Wilkinson's brand of numerical analysis is perhaps best regarded as an extra layer in the analysis of approximate solutions. It slips in just above the arithmetic facilities themselves.

2) The pages that follow give the erroneous impression that Wilkinson single-handedly showed the world how to analyze the effect of roundoff error. Yet this mode of expression is no worse than the familiar statement that William of Normandy won the battle of Hastings in 1066. Wilkinson did receive all the honors and most would agree that he became the leader of the group. Yet he was not working in isolation. Other people, independently, came to understand how roundoff errors can destroy a computation. I would like to intrude my personal opinion that had Wilkinson returned to classical analysis at Cambridge in 1947 our present state of understanding of roundoff would not be significantly changed. F. L. Bauer of Munich could have become the dominant figure, or H. Rutishauser of Zurich.

3) The production of ***The Handbook*** was a remarkable achievement. It testifies to cordial and close cooperation between leading experts in several European countries, the USA, and Australia. In contrast, consider the application of the simplex algorithm to linear programs and the finite element method to analyze structures. There it was the habit for engineers with debugged programs to form companies round those programs. The quest for profit stifled cooperation for improvement. Wilkinson's friendly yet exacting personality played no small part in the success of the Handbook venture. I am aware of no disharmony among the leading researchers on matrix problems.

4) A digital computer works with a finite set of representable numbers which may be combined using operations $\oplus, \ominus, \otimes, \oslash$ that mimic the familiar $+, -, \times, \div$. Unfortunately, some basic properties of the rational number field fail to hold for the computer's system. For example, the associative law fails for both addition and multiplication. Nevertheless, there is some algebraic structure left and it seemed quite likely during the 1950's that rigorous error analysis would have to be carried out in this unattractive setting. Indeed there have appeared a number of ponderous tomes that do manage to abstract the computer's numbers into a formal structure.

Perhaps Wilkinson's greatest achievement was to deflect analysis of algorithms from that morass into a place where insight and simplicity can survive. He makes no use of the pseudo-operators preferring to work with the exact relations satisfied by the computed quantities.

5) In contrast to most mathematicians and despite over 100 published papers, Wilkinson's contribution to numerical analysis is contained in the three books of which he is an author.

3. Roundoff Error Analysis

Wilkinson is honored for achieving a very satisfactory understanding of the effect of rounding errors during the execution of procedures that are used for solving matrix problems and finding zeros of polynomials. He managed to share his grasp of the subject with others by making error analysis intelligible, in particular by systematic use of the "backward" or inverse point of view. This approach asks whether there is a tiny perturbation of the data such that execution of the algorithm in exact arithmetic using the perturbed data would terminate with the actual computed output derived from the original data.

Wilkinson did not invent backward error analysis nor did he refrain from using the natural (or forward) error analysis when convenient. Although his name is not associated with any particular method he performed rigorous analyses of almost every method that was under discussion and trial. This work led him to become one of the leaders of an activity known as mathematical software production. The collection of Algol procedures contained in ***The Handbook*** (see reference list) is a seminal contribution to this branch of computer science.

Most of what follows is amplification of the preceding paragraphs. If the reader is impatient for a theorem or delicate inequality, the following quotation from ***Modern Error Analysis*** (1971) may engender a little forbearance. This is from the published version of his von Neumann lecture.

> "There is still a tendency to attach too much importance to the precise error bounds obtained by an *a priori* error analysis. In my opinion, the bound itself is usually the least important part of it. The main object of such an analysis is to expose the potential instabilities, if any, of an algorithm so that, hopefully, from the insight thus obtained one might be led to improved algorithms. Usually the bound itself is weaker than it might have been because of the necessity of restricting the mass of detail to a reasonable level and because of the limitations imposed by expressing the errors in terms of matrix norms. *A priori* bounds are not, in general, quantities that should be used in practice. Practical error bounds should usually be determined by some form of *a posteriori* error analysis, since this takes full advantage of the statistical distribution of rounding errors and of any special features, such as sparseness, of the matrix."

We would add that there is as yet no satisfactory format for presenting an error analysis so that its message can be summarized succinctly. Could we say that the analysis is the message?

4. The Linear Equations Problem

Given an $n \times n$ real invertible matrix A and $b \in \mathbb{R}^n$ the task is to compute $x = A^{-1}b$. The familiar process known as Gaussian elimination lends itself to implementation on automatic digital computers. It is also well known that Gaussian elimination is one way to factor A into the product LU where L is lower triangular and U is upper triangular. Once L and U are known the solution x is obtained by solving two triangular systems: $Lc = b$, $Ux = c$.

In 1943, Hotelling published an analysis showing that the error in a computed inverse X might well grow like 4^{n-1} where n is the order of A. Alan Turing was making similar analyses informally in England. The fear spread that Gaussian elimination was probably unstable in the face of roundoff error. The search was on for alternative algorithms.

In 1947 Goldstine and von Neumann, in a formidable 80-page paper published in the *Bulletin of the American Mathematical Society*, corrected this false impression to some extent. Some scholars have chosen the appearance of this paper as the birthday of modern numerical analysis. Among other things, this paper showed how the systematic use of vector and matrix norms could enhance error analysis. However it had the unfortunate side effect of suggesting that only people of the calibre of von Neumann and Goldstein were capable of completing error analyses and, even worse, that the production of such work was very boring. Their principal result was that, if A is symmetric, positive definite, then the computed inverse X satisfies

$$\|AX - I\| \leq (14.2)n^2 \varepsilon \mathrm{cond}(A)$$

where $\mathrm{cond}(A) = \|A\| \cdot \|A^{-1}\|$, and $\|\cdot\|$ is the spectral norm and ε denotes the roundoff unit of the computer. Only if A is too close to singular will the algorithm fail and yield no X at all, but that is as it should be. The joy of this result was getting a polynomial in n, the pain was obtaining 14.2, a number that reflects little more than the exigencies of the analysis. Some nice use of "backward" error analysis occurs in the paper but it is incidental. There was good reason for this attitude.

A backward error analysis is not guaranteed to succeed. Indeed no one, to this day, has shown that a properly computed inverse X is guaranteed to be the inverse of some matrix *close* to A, i.e.,

$$X = (A + E)^{-1} \quad \text{and} \quad \|E\|/\|A\| \text{ is small.}$$

Indeed, it is likely that no such result holds in full generality. What is true is that each column of X is the corresponding column of the inverse of a matrix very close to A. Unfortunately, it is a different matrix for each column.

The success of their analysis of the positive definite case prompted von Neumann and Goldstein to recommend the use of the normal equations for solving $Ax = b$ for general A, i.e., $x = (A^TA)^{-1}A^Tb$. However, that was bad advice for several reasons.

The fact is that careful Gaussian elimination, if it does not break down, produces computed solutions z with tiny residuals. It was practical experience in solving systems of equations using desk-top calculators (with n as large as 18)! that persuaded Wilkinson and his colleagues (L.Fox and E. T. Goodwin) that Gaussian elimination does give excellent results even when A is far from being symmetric let alone positive definite. In his first book ***Rounding Errors in Algebraic Processes***, published in 1963, we find for the first time a clear statement of the situation (see p.108). The computed solution z satisfies

$$(A + K)z = b \ .$$

If inner products are accumulated in double precision before the final rounding then

$$\|K\|_\infty \leq g\varepsilon(2.005\, n^2 + n^3 + \tfrac{1}{4}\varepsilon n^4)\|A\|_\infty$$

where g is the element growth factor, namely, the ratio of the largest intermediate value generated in the process to a maximal element of A. The corresponding bound on the residual is

$$\|b - Az\|_\infty \leq g\varepsilon(2.005\, n^2 + n^3)\|z\|_\infty$$

provided that $\varepsilon n \ll 1$. The important quantity g is easily monitored during execution of the algorithm. In his celebrated 1961 paper on matrix inversion, Wilkinson obtains an *a priori* bound on g when A is equilibrated and the "complete" pivoting strategy is employed. This is a clever piece of analysis and yields:

$$g^2 = g(n)^2 < n(2^1 3^{\frac{1}{2}} \cdots n^{\frac{1}{n-1}}) \quad ,$$

a slowly growing function of n. Being a man of intellectual integrity Wilkinson hastens to show that the bound cannot be sharp and indeed is not realistic at all. For certain Hadamard matrices $g(n) = n$, but apart from these cases Wilkinson reports that he has *never* encountered a value of g exceeding 8 despite intensive monitoring of the programs in use at NPL.

At this point we wish to emphasize that all the reaults quoted so far do a disservice both to Wilkinson and the topic of error analysis. Neither the powers of n that appear in the inequalities quoted above nor the coefficients in front of those powers convey genuine information about the process under analysis! It could be argued that the residual bound $\|b-Az\| < g\varepsilon n^3\|z\|$ is very weak indeed. Wilkinson's contribution cannot be conveyed by quoting such theorems. His achievements in regard to Gaussian elimination was to show the following:

- The effect of roundoff errors is not difficult to analyze. Indeed, the analysis is now presented in undergraduate courses.

- If the element growth factor g is small (say, $g < 5$) then the computed solution will have a residual norm scarcely larger than that belonging to the representable vector closest to $A^{-1}b$.

- When A is ill-conditioned, i.e., $\|A\|\,\|A^{-1}\|\sqrt{\varepsilon} > 1$ then g is very likely to be 1 if a reasonable pivoting strategy is used. In fact, for many ill-conditioned matrices the complete pivoting strategy produces factors L and U with elements that diminish rapidly as the algorithm proceeds.

- The technique known as iterative refinement may be employed to obtain an accurate solution provided that the system is not too ill-conditioned for the precision of the arithmetic operations. Moreover if the iteration converges slowly then the coefficient matrix A must be ill-conditioned.

- The partial pivoting strategy cannot guarantee that g will be small. There exist matrices for which $g = 2^{n-2}$.

The following very specific result of the 1961 paper is, to me, more interesting and more informative than all its theorems. The Hilbert matrix was a favorite test example in the 1940's and 1950's,

$$H = (h_{ij}) \quad , \qquad h_{ij} = (i+j-1)^{-1}$$

H_n denotes the leading principal $n \times n$ submatrix of H. Formulae are known for H_n^{-1}. Wilkinson showed that when Gaussian elimination was used to invent H_5 on a binary machine then the act of rounding the fractions 1/3, 1/5, 1/6, 1/7, 1/9 to the closest representable numbers caused *more* deviation in the computed inverse than *all* the rounding errors that occur in the rest of the computation. That computation involves more than 100 multiplications and 100 additions.

Despite several significant insights, the celebrated 1961 paper still does not make clear just how stable Gaussian elimination is for solving Ax=b. The contrast between this paper and the 1963 book is instructive. The paper follows the lead of von Neumann and Goldstine and concentrates exclusively on the problem of matrix inversion. Not only are the error bounds rather large but backward error analysis fails. However the problem of matrix inversion is not very important. The overwhelming demand is for solving systems of equations and here the backward analysis is simple and very satisfactory. The computed solution z satisfies some equation $(A + K)z = b$ and the insight comes in seeing how K depends on L, U and other quantities. The insight vanishes when norms are taken. Too much information is discarded.

5. The Eigenvalue Problem

Nearly three quarters of Wilkinson's publication list is devoted to this subject. No specific method bears his name yet every available method was analyzed by him and most of the published implementations of the better techniques owe something to his careful scrutiny.

The eigenvalue problem comprises many subproblems. The primary distinction is between symmetric matricea and the rest. For both classes the eigenvectors may or may not be needed. It is easy to describe Wilkinson's contribution to this topic. It is his magnum opus, ***The Algebraic Eigenvalue Problem*** (Oxford University Press, 1965). However that gives only half the picture. That book gave the understanding needed to produce the eigenvalue programs that appeared in the Handbook (***Handbook for Automatic Computation***, vol. II, ***Linear Algebra***, edited, Springer-Verlag, 1971). The latter was edited jointly with the gifted but self-effacing Dr. Christian Reinsch. The Handbook gave rise to the collection of FORTRAN programs called EISPACK which first appeared in 1974. The later version of these routines (1977) are available in virtually every scientific computer center in the world. It is pleasant to report that this useful product was achieved by the willing cooperation of many experts. To some extent this happy outcome is due to Wilkinson's generous and agreeable personality for he was certainly the leader of the group.

At a more technical level we now discuss some of his "results". Journal articlea are given in the brief reference list.

- His study of polynomials, and the sensitivity of their zeros to changes in their coefficients,

helped to stop the quest for the characteristic polynomial as a means of computing eigenvalues. See ***Rounding Errors in Algebraic Processes.***

• In 1954, W. Givens explicitly used backward error analysis to demonstrate the extreme accuracy of the Sturm sequence technique for locating specified eigenvalues of symmetric tridiagonal matrices. His analysis was for fixed point arithmetic and was never published. Wilkinson showed that the result still holds for standard floating point arithmetic and, contrary to popular wisdom, that backward error analysis of most algorithms is easier to perform for floating point arithmetic. Even more interesting was his demonstration that Givens Sturm sequence algorithm could be disastrous for computing eigenvectors while simultaneously being superb for locating eigenvalues. The point is worth emphasizing. Given an appropriate eigenvalue that is correct to working precision the eigenvector recurrence can sometimes produce an approximate eigenvector that is orthogonal to the true direction to working accuracy yet the signs of the computed components are correct. The contribution here was a well chosen class of examples.

• Wilkinson showed that the backward error analysis of *any* method employing a sequence of orthogonal similarity transformations can be made clear and simple. In particular the final matrix is similar to a small perturbation of the original matrix. This perturbation is essentially the sum of the local errors at each step: there is no propagated error.

An important consequence of this analysis is the following. Let C denote the equivalence class of matrices orthogonally similar to the original matrix. For the computation of eigenvalues it does not matter if roundoff errors cause the computed sequence to depart violently from the exactly computed sequence ***provided that*** the computed sequence lies close to C. A naive forward analysis can miss vital correlations between computed quantities.

Indeed a number of efficient, stable algorithms do regularly produce intermediate quantities that differ significantly from their exact counterparts. Nevertheless eigenvalues are preserved to within working accuracy. The QR algorithm is an example of this phenomenon.

• Although it was invented in 1959/60, the QR algorithm of J.G.F.Francis did not achieve universal acceptance until about 1965. It provides an ideal way to diagonalize a symmetric tridiagonal matrix since it produces a sequence of symmetric tridiagonal matrices that converge to diagonal form. However the QR algorithm requires a strategy for choosing shifts. Let

$$T_n = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \cdot & \cdot & \\ & & \cdot & \cdot & \beta_{n-1} \\ & & & \beta_{n-1} & \alpha_n \end{bmatrix}$$

be such a matrix with $\beta_i > 0$, $i = 1,\ldots,n-1$. Wilkinson's shift w is defined to be the eigenvalue of $\begin{pmatrix} \alpha_{n-1} & \beta_{n-1} \\ \beta_{n-1} & \alpha_n \end{pmatrix}$ that is closer to α_n. It is the favorite strategy (rather than choosing α_n) but when it was first introduced there was no proof that it would always lead to convergence. Convergence here means that $\beta_{n-1} \to 0$ as the algorithm is continued without limit. The Rayleigh quotient shift α_n causes the β_{n-1} to be monotone decreasing but the limit need not vanish. Wilkinson's shift sacrifices the monotonicity and gains convergence.

In a tour-de-force in 1971 Wilkinson proved that, with his strategy, convergence is assured (in exact arithmetic) and is usually cubic. A tricky argument showed that the product $\beta_{n-2}\beta_{n-1}$ is monotone decreasing to zero though initially the rate could be very slow.

This was not the last word however. In 1979 Parlett and Hoffman discovered an elementary proof that $\beta^2_{n-1}\beta_{n-2}$ decreases geometrically at each step by a factor at most $1/\sqrt{2}$. Convergence of β_{n-1} follows readily.

• In 1976, Wilkinson and Golub published a long article on the Jordan canonical form. They discussed its discontinuous dependence on the matrix elements in the defective case. They showed how to go about computing robust bases for the associated cyclic subspaces (the numerical analyst's Jordan chains of principal vectors), and they also explained the limitations of this form in practical calculations.

A natural extension of this research was to the computation of the Kronecker form of a pair of matrices (A,B). This form arises in the study of systems of differential equations with constant coefficients:

$$B\dot{u} = Au\,, \qquad u(0) \in \mathbb{R}^n \text{ is given}\,.$$

• In the last decade of his life Wilkinson's attention was more and more attracted to the difficult and still open problem of determining, for any given A, the closest defective matrix B.

6. The Zeros of Polynomials

Until near the end of the 1950's the computation of the zeros of polynomials was regarded as an important activity in scientific computation. So it is not surprising that a significant part of Wilkinson's work of this period was devoted to this task. His contribution is consolidated in Chapter 2 of ***Rounding Errors in Algebraic Processes.*** Thanks in part to his discoveries, polynomials no longer attract much attention. It was the advent of digital computers that drove people to think in detail about general polynomials of large degree; 20 or 100 or 1000.

Since isolated zeros are analytic functions of the coefficients one may consider the derivative of any isolated zero with respect to each coefficient. As the degree rises these derivatives can hardly avoid becoming huge. The presence of such an ill-conditioned zero can make it difficult to compute comparatively well conditioned zeros.

By use of well chosen examples Wilkinson brought these facts home to numerical analysts. Of considerable personal interest is the fact that

Wilkinson was led to an explicit appreciation of the importance of backward error analysis when he investigated the reliability of Horner's method (also known as nested multiplication) for *evaluating* a polynomial. He realized that, with floating point arithmetic, the output of Horner's recurrence is, in all cases, the exact value of a polynomial each of whose coefficients is a tiny relative perturbation of the original one. The relative change in the coefficient of x^r is less than

$$(1.01)(r+1)2\varepsilon \quad ,$$

where ε is the roundoff unit. In the majority of cases the inherent uncertainty in each coefficient will exceed the worst case error given above. In this way a fearsome error analysis melts away into classical perturbation theory.

One of Wilkinson's final works, "The Perfidious Polynomial" (Chapter I in *Studies in Numerical Analysis*, G. H. Golub, editor, Math. Assoc. Amer., vol. 24, 1984) sums up his experience with polynomials in a way that is designed for readers outside numerical analysis. This pellucid essay was awarded the Chauvenet prize for mathematical exposition. Unfortunately Wilkinson died before he could receive it.

REFERENCES

Books by JHW

Rounding Errors in Algebraic Processes, Notes on Applied Science No. 32, National Physical Laboratory, England, HMSO, 1963. Also published in 1964 by Prentice-Hall, Inc.

The Algebraic Eigenvalue Problem, Oxford University Press, 1965.

Handbook for Automatic Computation, vol. II: *Linear Algebra*, (edited with C. Reinsch), Springer-Verlag, 1971.

Selected Articles by JHW

"Error Analysis of Direct Methods of Matrix Inversion", *Journal of the Association for Computational Machinery*, vol. 8 (1961), 281-329.

"Error Analysis of Eigenvalue Techniques Based on Orthogonal Transformations", *Journal of SIAM*, vol. 10 (1962), 162-195.

"Global convergence of tridiagonal QR algorithm with origin shifts", *Linear Algebra and Its Applications*, vol. 1 (1968), 409.

(with G. H. Golub), "Ill-conditioned eigensystems and the computation of the Jordan canonical form", *SIAM Review*, vol. 18 (1976), 578.

"Kronecker's canonical form and the QZ algorithm", *Linear Algebra and Its Applications*, vol. 28 (1979), 295-305.

"On neighbouring matrices with quadratic elementary divisors", *Numerical Mathematics*, vol. 44 (1984), 1-21.

Articles Not by JHW

H. H. Goldstine and J. von Neumann, "Numerical Inventing of Matrices of High Order", *Bulletin of the American Mathematical Society*, vol. 53 (1947), 1021-1099.

W. Hoffman and B. N. Parlett, "A New Proof of Global Convergence for the Tridiagonal QL Algorithm", *SIAM Journal of Numerical Analysis*, vol. 15 (1978), 929-937.

H. Hotelling, "Some new methods in matrix calculations", *Annals of Mathematical Statistics*, vol. 14 (1943), 1-34.

The Influence of George Forsythe and his Students

James Varah
Head, Computer Science Department
University of British Columbia

1. Introduction

It is a pleasure to have the opportunity to comment on the influence of George Forsythe, from the point of view of one of his students, and from a perspective 15 years after his death. This article owes much to earlier commentaries on George, published immediately following his death, and also to material made available to me by George's daughter, Diana Forsythe, and by the Stanford Archives.

2. His Life

The facts concerning George's life are easy enough to list: he was born in 1917 in State College, Pa. and spent most of his formative years in Ann Arbor, Michigan. He attended Swarthmore College, graduating with a major in Mathematics in 1937. He attended graduate school at Brown University, and received his Ph.D. in 1941. He worked during the War for the Air Force as a meteorologist, and following the War worked principally for UCLA and the National Bureau of Standards, before going to Stanford University as Professor of Mathematics in 1957.

It was for the period at Stanford, from 1957 to his untimely death in 1972, that George is probably best remembered. His interest in computing (scientific and otherwise), which had begun at NBS, developed and flourished at Stanford - he was instrumental in forming a Computer Science Division within the Math Department in 1961, and was Director of the Stanford Computation Centre from 1961 to 1965. Then in 1965, he became the first Head of the newly formed Department

of Computer Science. During his headship, the Department developed into one of the truly outstanding Departments of Computer Science anywhere, a position it has continued to hold.

George died very suddenly of cancer in the Spring of 1972. His untimely death was a shock to all his many colleagues and friends, and resulted in several memorials and dedications: an article in the SIGNUM Newsletter (Moler [1972]); two articles in the CACM (Herriot [1972], Knuth [1972]); a special Stanford memorial resolution by Harriot et al., available from the Stanford archives; and a special issue of the SIAM Journal on Numerical Analysis (April 1973), with a dedication by Alston Householder. Moreover, when a new building to house the Computation Centre was built in 1980, it was named after him. Two national awards bear his name: the ACM undergraduate paper competition, and the SIGNUM memorial lecturer award for leadership in numerical mathematics.

3. His Research

George's early interest in scientific computation was fostered by the meteorological problems he was involved with during the War. Then while at NBS, he interacted with many of the early pioneers in scientific computation, when this group was coming to grips with the intricacies of basic floating-point computation. His early work on the numerical solution of partial differential equations culminated in his 1960 book with Wasow, Finite Difference Methods for Partial Differential Equations. This book remained a standard in the field for many years.

He also made contributions to the use of orthogonal polynomials in scientific computation and to our understanding of various aspects of the solution of linear systems. Two other textbooks remain in use today: Computer Solution of Linear Algebraic Systems, with Cleve

Moler (1967), and Computer Methods for Mathematical Computations, with Moler and Michael Malcolm (1973).

He was instrumental in pointing out the significance of finite arithmetic in the computational solution of fundamental mathematical problems - his article Pitfalls in Computation, published in the American Math. Monthly in 1970, for example, is still an excellent source of instructional material on the subject.

A full list of publications (4 books, 83 articles) is given in Knuth [1972].

Besides his own research in numerical computation, George was one of the early pioneers in computer science education. He advocated the introduction of computing into mathematics education at an early stage; moreover he was one of the first proponents of Computer Science as a separate discipline. The concept of algorithms as central to Computer Science was clear in his mind: he was Algorithms Editor for the Communications of the ACM from 1964 to 1966, and was President of the ACM for the same period. I can recall as a graduate student helping him with the editing of the Algorithms section - jogging referees and examining new algorithm proposals. He was remarkably diligent and enthusiastic about this work, believing the Algorithms to be an essential part of the CACM.

4. His Students

Besides the Stanford Computer Science Department, George's most enduring legacy is his Ph.D. students. In his 15 years at Stanford, George had 17 students receive their Ph.D. in Mathematics or Computer Science. Here is the complete list:

Ph.D. Students

1. Eldon Hanson (1960): presently with Lockheed Aerospace Corp., Sunnyvale, Cal.
   thesis: On Jacobi methods and block-Jacobi methods for computing matrix eigenvalues.
2. James Ortega (1962): presently Chairman, Dept. of Applied Mathematics, University of Virginia.
   thesis: An error analysis of Householder's method for the symmetric eigenvalue problem.
3. Betty Jane Stone (1962): presently living in Washington, D.C.
   thesis: (a) Best possible ratios of certain matrix norms.
   (b) Lower bounds for the eigenvalues of a fixed membrane.
4. Beresford Parlett (1962): presently Professor of Math and Computer Science, University of California at Berkeley
   thesis: Application of Laguerre's method to the matrix eigenvalue problem
5. Donald Fisher (1962): presently at Oklahoma State University, Norman, OK.
   thesis: Calculation of subsonic cavities with sonic free streamlines.
6. Ramon Moore (1963): presently Professor of Computer Science, Ohio State University.
   thesis: Interval arithmetic and automatic error analysis in digital computing. (joint supervision with McGregor)
7. Robert Causey (1964): presently Professor and Chairman, Dept. of Computer Science, Chris Newport College, Newport News, VA.
   thesis: On closest normal matrices
8. Cleve Moler (1965): presently with Intel Corp., Beaverton, Ore.
   thesis: Finite difference methods for the eigenvalues of Laplace's

operator.

9. James Daniel (1965): presently Professor of Math, University of Texas.

   thesis: The conjugate gradient method for linear and nonlinear operator equations. (joint supervision with Schiffer)

10. Donald Grace (1965): presently at Oklahoma State University, Norman, OK.

    thesis: Computer search for nonisomorphic convex polyhedra (joint supervision with Polya)

11. Roger Hockney (1966): recently retired from University of Reading, England.

    thesis: The computer simulation of anomalous plasma diffusion and the numerical solution of Poisson's equation (joint supervision with Golub and Buneman)

12. James Varah (1967): presently Head, Computer Science Dept., University of British Columbia

    thesis: The computation of bounds for the invariant subspaces of a general matrix operator.

13. Paul Richman (1968): presently with Bell Laboratories, Chicago, IL.

    thesis: (a) ε-calculus

    (b) Transonic fluid flow and approximation of the iterated integrals of a singular function (joint supervision with Herriot)

14. Alan George (1971): presently Distinguished Professor, University of Tenn. and ORNL.

    thesis: Computer implementation of the finite element method (joint supervision with Dorr)

15. Richard Brent (1971): presently Head, Dept. of Computer Science, Australian National University.

    thesis: Algorithms for finding zeros and extrema of functions

without calculating derivatives
(joint supervision with Dorr and Moler)

16. David Stoutemyer (1972): presently Professor of Computer Science, University of Hawaii
thesis: Numerical implementation of the Schwartz alternating procedure for elliptic partial differential equations.

17. Michael Malcolm (1973): presently President, WMI, Waterloo, Ont. and Adjunct Professor, University of Waterloo.
thesis: Nonlinear Splines

Some of these people have pursued careers in industry or government; others have stayed in an academic environment, and produced Ph.D. students of their own. A (incomplete) "tree" of these students is reproduced as Appendix I, and includes 71 names.

One of the striking aspects of George's interactions with his students is the lack of joint authorship. Apart from the two books with Moler, he did not produce joint research papers with his students. This was a conscious decision on his part: he felt that the student's research belonged to the student, who should get full credit for it.

Yet this is not to say that George didn't play a large role in the development of the thesis research. Far from it; George's approach was to be so interested in the problem at hand that the student would naturally be inspired to pursue the topic thoroughly. He had a very good sense of when to go into detail, and when to leave the work for the student, so that in the end the student felt it was his own work, but in fact George had played a large part in the development.

Here is a direct quote I received recently from one of his students. I'm sure the rest of his students would agree with the sentiments expressed.

"Forsythe opened my eyes to the compelling excitement of research and scholarship. Above all, he gave so very generously of his time and

talent. Every week I would send him a written summary of my research ideas, then we would meet for an hour to discuss them. He even had the patience to correct every single spelling and grammatical mistake. He helped steer me clear of my bad ideas, and his numerical instincts were uncanny. I have tried to use him as a role model for my relations to students, but I do not expect to attain his degree of patience and generosity."

The Ph.D. research topics covered by George's students ranged all over the map of scientific computation - from basic numerical linear algebra to optimization and zero-finding to numerical techniques for partial differential equations. One general theme which he liked to pursue was the development of precise, sharp, computable error bounds for various algorithms and problems. In his graduate course in numerical computation, he focussed this theme on a particular problem involving zeros of the first Bessel function, using a Taylor series expansion. He would insist that the students produce an algorithm to find the zeros, and give error bounds for them using roundoff error bounds for each step of the computation. If one was careful about the work, rather sharp bounds could be obtained, and we all found the exercise very illuminating. I still use this example when I teach roundoff analysis.

Another theme which emerges from reviewing the thesis work (and subsequent research) of George's students is the use of the computer as an essential tool in the understanding of mathematical phenomena. He emphasized the development of <u>algorithms</u> more than theorems - and thus made us (his students) feel at home in Computer Science Departments. His famous quotation, given during his IFIP address in 1971, "Numerical analysts have gone over the last 15 years from being queer people in Mathematics Departments to being queer people in Computer Science Departments", remains true today. But I believe that he would still feel that there is a place for Numerical Analysis, or as we prefer to call it today, Scientific Computation, in Computer Science

Departments.

5. The Work of His Students

As mentioned earlier, George passed on to all his students the importance of actual, hands-on computation. Mathematics, and mathematical theorems, were not neglected, but were not the central issue. That was computation, and algorithms for computation - and a deep understanding of the algorithms was at the heart of his brand of Numerical Analysis.

His students' work certainly has continued this theme - for example Ortega's work on understanding algorithms for nonlinear systems of equations; Parlett's work on understanding algorithms for matrix eigenvalue problems; Moore's work on algorithms for interval arithmetic; Moler's work on the algorithms of LINPACK, EISPACK, and MATLAB; Alan George's work on algorithms for sparse linear equations; and Brent's work on algorithms for zero-finding. All of this work is highly regarded by the scientific computation community.

Besides the hundreds of scientific papers contributed by George's students, there have also been highly regarded textbooks. For example:

1. The Numerical Solution of Nonlinear Systems of Equations, by Ortega and Rheinboldt
2. The Symmetric Eigenvalue Problem, by Parlett
3. Interval Analysis, by Moore
4. Computer Solution of Linear Algebraic Systems, by Forsythe and Moler
5. Computer Methods for Mathematical Computations, by Forsythe, Malcolm, and Moler
6. Computer Solution of Large Sparse Positive Definite Systems, by George and Liu.

Moreover, many of his students have held administrative positions

with universities and national organizations. Several have been members of the SIAM Council and Board, for example. Jim Ortega is currently Chairman of Applied Mathematics at the University of Virginia; Beresford Parlett served a term as Chairman of Computer Science at Berkeley; Cleve Moler was Chairman of Computer Science at New Mexico; Jim Daniel was Chairman of Mathematics at Texas; Jim Varah is Head of Computer Science at British Columbia; Alan George was Dean of Mathematics at Waterloo; and Richard Brent is Head of Computer Science at Canberra.

6. His Legacy

It is now 15 years since George died; scientific computation has grown enormously in this period and its applications are felt over a wide range of disciplines. Yet his approach to the subject, and his attitude towards research, remain as relevant as ever. His insistence on a fundamental understanding of the basic mechanics of floating-point arithmetic, his emphasis on algorithm development, his keen interest in any new unsolved problem, and his generous, open manner regarding research problems are just some examples. Those of us who were fortunate enough to work with him are charged with the responsibility of carrying on his spirit of inquiry, and his essential humanity, and conveying them to future generations of scientists.

References

Herriot [1972]: In Memory of George E. Forsythe. Comm. ACM 15 (Aug. 1972), 719-720.

Knuth [1972]: George Forsythe and the Development of Computer Science. Comm. ACM 15 (Aug. 1972), 721-726.

Moler [1972]: A Memory of George Forsythe. SIGNUM Newsletter 7 (Oct. 1972), 8-9.

Appendix I

The Updated Forsythe Student Tree

1. Hansen, Eldon - 1960

2. Ortega, James - 1962

    - 2.1 Elkin, Richard - 1968
    - 2.2 Caspar, Joseph - 1969
    - 2.3 Stepleman, Robert - 1969
        - 2.3.1 Shoosmith, John - 1973
    - 2.4 Voigt, Robert - 1969
    - 2.5 More, Jorge - 1970
        - 2.6.1 Thomas, Steve - 1974
    - 2.6 Lambiotte, Jules - 1975
    - 2.7 Adams, Loyce - 1982
    - 2.8 Romine, Charles - 1986
    - 2.9 Poole, Eugene - 1986

3. Stone, Betty - 1962

4. Parlett, Beresford - 1962

    - 4.1 Johnson, Olin - 1968
    - 4.2 Bunch, James - 1969
    - 4.3 Poole, William - 1970
    - 4.4 Nazareth, Larry - 1973
    - 4.5 Chen, N.F. - 1975
    - 4.6 Wang, Ying - 1975
    - 4.7 Scott, David - 1978
    - 4.8 White, T. - 1980
    - 4.9 McCurdy, A. - 1981
    - 4.10 Greenbaum, A. - 1981
    - 4.11 Nour-Omid, B. - 1982
    - 4.12 Simon, H. - 1982
    - 4.13 Taylor, D.L. - 1983
    - 4.14 Ng, K.C. - 1983

5. Fisher, Donald - 1962

6. Moore, Ramon - 1963

6.1 Talbot, Thomas - 1968
6.2 Wittie, Larry - 1974
6.3 Athavale, M.L. - 1974
6.4 Lee, Y.D. - 1980
6.5 Jones, Sandie - 1978

7. Causey, Robert - 1964

8. Moler, Cleve - 1965

8.1 Schryer, Norman - 1969
8.2 Cline, Alan K. - 1970
8.3 Crawford, Charles - 1970
8.4 Kammler, David - 1971
8.5 Eisenstat, Stanley - 1972
8.6 Kaufman, Linda - 1973
8.7 VanLoan, Charles - 1973
8.8 Burris, Charles - 1974
8.9 Sanderson, James - 1976
8.10 Starner, John - 1976
8.11 Davis, George, 1979
8.12 Dongarra, Jack - 1980
8.13 Jones, Ronal - 1985
8.14 Dubrulle, Augustin - 1986
8.15 Madrid, Humberto - 1986

9. Daniel, James - 1965

10. Grace, Donald - 1965

11. Hockney, Roger - 1966

11.1 Brownigg, David - 1975

12. Varah, James

12.1 Doedel, E.J. - 1976
12.2 Benson, Maurice - 1978
12.3 Foreman, Michael - 1984

13. Richman, Paul - 1968

14. George, Alan - 1971

14.1 Liu, Joseph - 1976
14.2 Gonnet, Gaston - 1977
14.3 McIntyre, David - 1981
14.4 Ng, Esmond - 1983
14.5 Rashwan, Hamza - 1985

15. Brent, Richard - 1971

16. Stoutemyer, David - 1972

17. Malcolm, Michael - 1973

ORIGINS OF THE SIMPLEX METHOD

by George B. Dantzig
Stanford University

In the summer of 1947, when I first began to work on the simplex method for solving linear programs, the first idea that occurred to me is one that would occur to any trained mathematician, namely the idea of step by step descent (with respect to the objective function) along edges of the convex polyhedral set from one vertex to an adjacent one. I rejected this algorithm outright on intuitive grounds — it had to be inefficient because it proposed to solve the problem by wandering along some path of outside edges until the optimal vertex was reached. I therefore began to look for other methods which gave more promise of being efficient, such as those that went directly through the interior, [1].

Today we know that before 1947 that four isolated papers had been published on special cases of the linear programming problem by Fourier (1824) [5], de la Vallée Poussin (1911) [6], Kantorovich (1939) [7] and Hitchcock (1941) [8]. All except Kantorovich's paper proposed as a solution method descent along the outside edges of the polyhedral set which is the way we describe the simplex method today. There is no evidence that these papers had any influence on each other. Evidently they sparked zero interest on the part of other mathematicians and were unknown to me when I first proposed the simplex method. As we shall see the simplex algorithm evolved from a very different geometry, one in which it appeared to be very efficient.

The linear programming problem is to find

$$\min z,\ x \geq 0 \quad \text{such that} \quad Ax = b,\ cx = z(\min), \tag{1}$$

where $x = (x_1, \ldots, x_n)$, A is an m by n matrix, and b and c are column and row vectors.

Curiously enough up to 1947 when I first proposed that the a model based on linear inequalities be used for planning activities of large-scale enterprises, linear inequality theory had produced only forty or so papers in contrast to linear equation theory and the related subjects of linear algebra and approximation which had produced a vast literature, [4]. Perhaps this disproportionate interest in linear equation theory was motivated more than mathematicians care to admit by its practical use as an important tool in engineering and physicis, and by the belief that linear inequality systems would not be practical to solve unless they had three or less variables, [5].

My proposal served as a kind of trigger — ideas that had been brewing all through World War II but had never found expression burst forth like an explosion. Almost two years to the day that I first proposed that L.P. be used for planning, Koopmans organized the 1949 conference (now referred to as *The Zero-th Symposium on Mathematical Programming*) at the University of Chicago. There mathematicians, economists, and statisticians presented their research and produced a remarkable proceedings entitled *Activity Analysis of Production and Allocation*, [2]. L.P. soon became part of the newly developing professional fields of Operations Research and Management Science. Today thousands of linear programs are solved daily throughout the world to schedule industry. These involve many hundreds, thousands and sometimes tens of thousands of equations and variables. Some mathematicians rank L.P. as "the newest yet most potent of mathematical tools" [16].

John von Neumann, Tjalling C. Koopmans, Albert W. Tucker, and others well known today, some just starting their careers back in late 1940's, played important roles in L.P.'s early development. A group of young economists associated with Koopmans (R. Dorfman, K. Arrow, P. Samuelson, H. Simon and others) became active contributors to the field. Their research on L.P. had a profound effect on economic theory leading to Nobel Prizes. Another group led by A.W.Tucker, notably D. Gale and H. Kuhn, began the development of the mathematical theory.

This outpouring between the years of 1947-1950 coincided with the first building of digital computers. The computer became *the* tool that made the application of linear programming possible. Everywhere we looked, we found practical applications that no one earlier could have posed seriously as optimization problems because solving them by hand computation would have been out of the question. By good luck, clever algorithms in conjunction with computer development gave early promise that linear programming would become a practical science. The intense interest by the Defense Department in the linear programming application also had an important impact on the early construction

of computers [17]. The U.S. National Bureau of Standards with Pentagon funding became a focal point for computer development under Sam Alexander; its Mathematics Group under John Curtis began the first experiments on techniques for solving linear programs primarily by Alan Hoffman, Theodore Motzkin, and others [3].

Since everywhere we looked, we could see possible applications of linear programs, it seemed only natural to suppose that there was extensive literature on the subject. To my surprise, I found in my search of the contemporary literature of 1947 only a few references on linear inequality systems and none on solving an optimization problem subject to linear inequality constraints.

T.S. Motzkin in his definitive 1936 Ph.D. thesis on linear inequalities [4] makes no mention of optimizing a function subject to a system of linear inequalities. However, 15 years later at the First Symposium on Linear Programming (June 1951), Motzkin declared: "there have been numerous rediscoveries [of LP] partly because of the confusingly many different geometric interpretations which these problems admit". He went on to say that different geometric interpretations allows one "to better understand and sometimes to better solve cases of these problems as they appeared and developed from a first occurrence in Newton's Methodus Fluxionim to right now".

The "numerous rediscoveries" that Motzkin referred to probably were to two or three papers we have already cited concerned with finding the least sum of absolute deviations, or minimizing the maximum deviation of linear systems, or determining whether there exists a solution to a system of linear inequalities. Fourier pointed out as early as 1824 these were all equivalent problems, [5]. Linear Programs, however, had also appeared in other guises. In 1928, von Neumann [19] formulated the zero-sum matrix game and proved the famous Mini-Max Theorem, a forerunner of the famous Duality Theorem of Linear Programming (also due to him) [11]. In 1936, Neyman-Pearson considered the problem of finding an optimal critical region for testing a statistical hypothesis. Their famous Neyman-Pearson Lemma is a statement about the Lagrange Multipliers associated with an optimal solution to a linear program, [20].

After I had searched the the contemporary literature of 1947 and found nothing, I made a special trip to Chicago in June 1947 to visit T.J. Koopmans to see what economists knew about the problem. As a result of that meeting, Leonid Hurwicz, a young colleague of Koopmans, visited me in the Pentagon in the summer and collaborated with me on my early work on the simplex algorithm, a method which we described at the time as "climbing up the bean pole" — we were maximizing the objective.

Later I made another special trip, this one to Princeton in the fall of 1947, to visit the great mathematician Johnny von Neumann to learn what mathematicians knew about the subject. This was after I had already proposed the simplex method but before I realized how very efficient it was going to be, [1].

The origins of the simplex method go back to one of two famous unsolved problems in mathematical statistics proposed by Jerzy Neyman which I mistakenly solved as a homework problem; it later became part of my Ph.D. thesis at Berkeley, [9]. Today we would describe this problem as proving the existence of optimal Lagrange multipliers for a semi-infinite linear program with bounded variables. Given a sample space Ω whose sample points u have a known probability distribution $dP(u)$ in Ω, the problem I considered was to prove the existence of a critical region ω in Ω that satisfied the conditions of the Neyman-Pearson Lemma. More precisely, the problem concerned finding a region ω in Ω that minimized the Lebesgue-Stieltjes integral defined by (4) below, subject to (2) and (3):

$$\int_\omega dP(u) = \alpha, \tag{2}$$

$$\alpha^{-1} \int_\omega f(u) dP(u) = b, \tag{3}$$

$$\alpha^{-1} \int_\omega g(u) dP(u) = z(\min), \tag{4}$$

where $0 < \alpha < 1$ is the specified "size" of the region; $f(u)$ is a given vector function of u with $m-1$ components whose expected value over ω is specified by the vector b; and $g(u)$ is a given scalar function of u whose unknown expected value z over ω is to be minimized.

Instead of finding a critical region, we can try to find the characteristic function $\phi(u)$ with the property that $\phi(u) = 1$ if $u \in \omega$ and $\phi(u) = 0$ if $u \notin \omega$. The original problem can then be restated as:

Find $\min z$ and a function $\phi(u)$ for $u \in \Omega$ such that:

$$\int_{u \in \Omega} \phi(u) dP(u) = \alpha, \quad 0 \leq \phi(u) \leq 1, \tag{5}$$

$$\alpha^{-1} \int_{u \in \Omega} \phi(u) f(u) dP(u) = b, \tag{6}$$

$$\alpha^{-1} \int_{u \in \Omega} \phi(u) g(u) dP(u) = z(\min). \tag{7}$$

A discrete analog of this semi-infinite linear program can be obtained by selecting n representative sample points $u^1, \ldots, u^j, \ldots, u^n$ in Ω and replacing $dP(u^j)$ by discrete

point probabilities $\Delta_j > 0$ where n may be finite or infinite. Setting

$$x_j = (\Delta_j/\alpha) \cdot \phi(u^j), \quad 0 \leq x_j \leq \Delta_j/\alpha, \tag{8}$$

the approximation problem becomes the bounded variable LP:

Find $\min z$, $0 \leq x_j \leq \Delta_j/\alpha_j$:

$$\sum_1^n x_j = 1 \tag{9}$$

$$\sum_1^n A_{\cdot j} x_j = b \tag{10}$$

$$\sum_1^n c_j x_j = z(\min) \tag{11}$$

where $f(u^j) = A_{\cdot j}$ are $m-1$ component column vectors, and $g(u^j) = c_j$.

Since n the number of descrete j could be infinite, I found it more convenient to analyze the L.P. problem in the geometry of the finite $(m+1)$ dimensional space associated with the coefficients in a column. I did so initially with the convexity constraint (9) but with no explicit upper bound on the non-negative variables x_j, [10], [2], [11]. Since the first coefficient in a column (the one corresponding to (9)) is always 1, my analysis omitted the initial 1 coordinate. Each column $(A_{\cdot j}, c_j)$ becomes a point (y, z) in R^m where $y = (y_1, \ldots, y_{m-1})$ has $m-1$ coordinates.

The problem can now be interpreted geometrically as one of assigning weights $x_j \geq 0$ to the n points $(y^j, z^j) = (A_{\cdot j}, c_j)$ in R^m so that the "center of gravity" of these points, see Figure 1, lies on the vertical "requirement" line (b, z) and such that its z coordinate is as small as possible.

Simplex Algorithm

Step t of the algorithm begins with an $m-1$ simplex, see Figure 1, defined by some m points $(A_{\cdot j_i}, c_{j_i})$ for $i = (1, \ldots, m)$ and m weights $x^0_{j_i} > 0$ (in the non-degenerate case) such that $\sum A_{\cdot j_i} x_{j_i} = b$. In the figure, the vertices of the $m-1=2$ dimensional simplex correspond to $j_1 = 1$, $j_2 = 2$, $j_3 = 3$. The line (b, z) intersects the plane of the simplex (the triangle in the figure) in an *interior point* (b, z_t). A point $(A_{\cdot s}, c_s)$ is then determined whose vertical distance below this "solution" plane of the simplex is maximum.

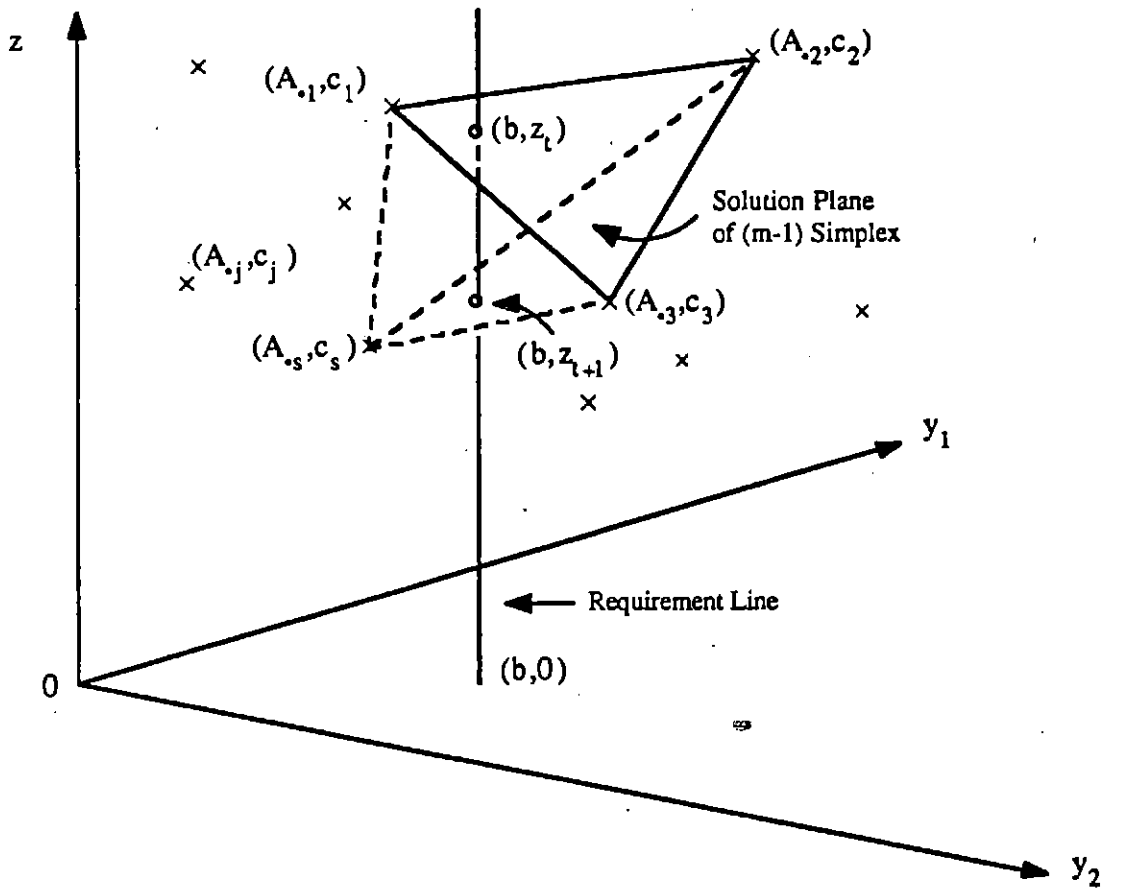


Figure 1. The m Dimensional Simplex

Algebraically the equation $z = \pi y + \pi_0$ of the plane associated with the simplex, is found by solving the system m equations $\pi A_{\cdot j_i} + \pi_0 = c_{j_i}$, $j_i = (j_1, \ldots, j_m)$. Next, let $j = s$ be the index of $(A_{\cdot s}, c_s)$ the point most below this plane, namely

$$s = \arg\min_j [c_j - (\pi A_{\cdot j} + \pi_0)]. \tag{12}$$

If $[c_s - (\pi A_{\cdot s} + \pi_0)]$ turns out to be non-negative, the iterative process stops, otherwise the m simplex, the tetrahedron in Figure 1, is formed as the convex combination of the point $A_{\cdot s}, c_s)$ and the $(m-1)$ simplex. The requirement line (b, z) intersects this m-simplex in a segment $(b, z_{t+1}), (b, z_t)$ where $z_{t+1} < z_t$. The face containing (b, z_{t+1}) is then selected as the new $(m-1)$-simplex. Operationally the point $(A_{\cdot s}, c_s)$ replaces $A_{\cdot j_r}, c_{j_r}$ for some r. The index r is not difficult to determine algebraically.

Geometrical insight as to why the simplex method is efficient can be gained by viewing the algorithm in two dimensions, see Figure 2. Suppose a piecewise linear function $y = f(x)$ is defined by the underbelly of the convex hull of the points $(y^j, z^j) = (A_{\cdot j}, c_j)$. We wish

to solve the equation $y = f(b)$ and to find two points $(y^j, z^j), (y^k, z^k)$ and the weights $(\lambda, \mu) \geq 0$ on these two points such that $\lambda y^j + \mu y^k = b$, $\lambda + \mu = 1$, $\lambda z^j + \mu z^k = f(b)$. In the two dimensional case, the simplex method resembles a kind of secant method in which, given any slope σ, it is cheap to find a point (y^s, z^s) of the underbelly such that the slope (actually the slope of a support) at y^s is σ, but where it is not possible given b to directly compute $y = f(b)$.

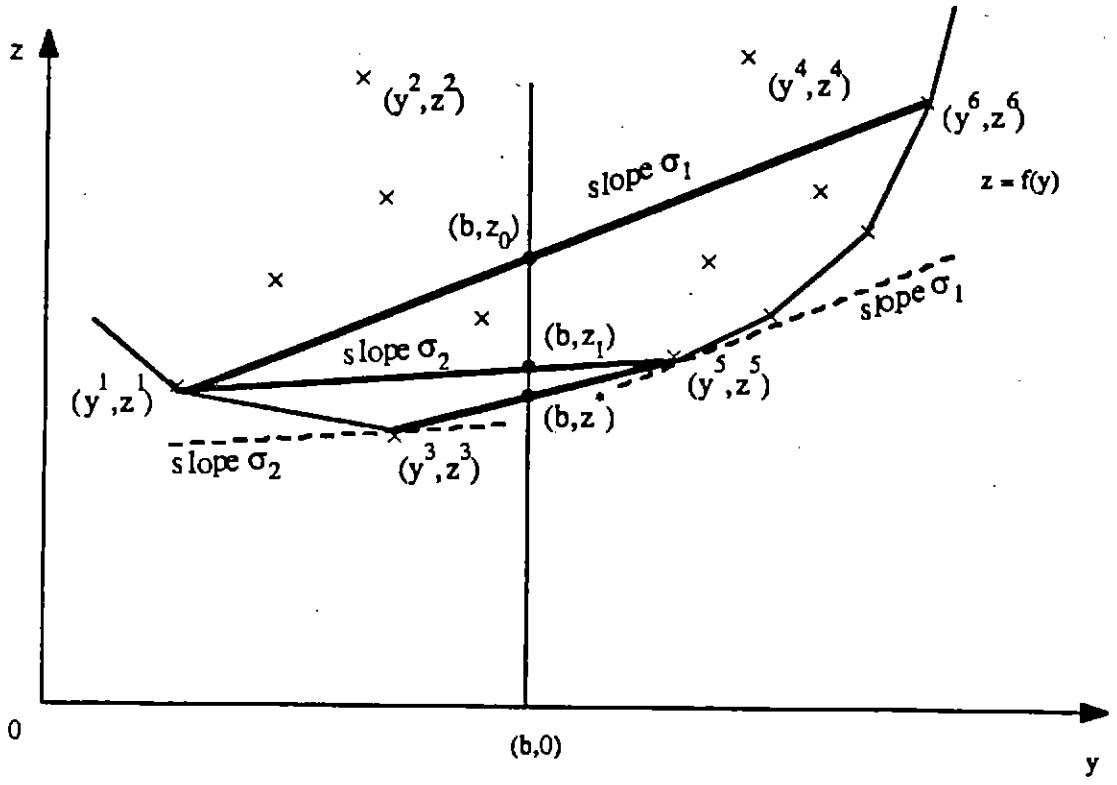


Figure 2. The Under-belly of the Convex Hull

In Figure 2, the algorithm is initiated (in Phase II of the simplex method) by two points, say (y^1, z^1) and (y^6, z^6), on opposite sides of the requirement line. The slope of the "solution" line joining them is σ_1. Next, one determines that the point (y^5, z^5) is the one most below the line joining (y^1, z^1) to (y^6, z^6) with slope σ_1. This is done algebraically by simply substituting the coordinates (y^j, z^j) into the equation of the solution line $z - z^6 = \sigma_1(y - y^6)$ and finding the point $j = s$ such that $\sigma_1(y^j - y^6) - (z^j - z^6)$ is maximum. For the example above, $s = 5$ and thus (y^5, z^5) replaces (y^6, z^6). The steps are then repeated with (y^1, z^1) and (y^5, z^5). The algorithm finds the optimum point (b, z^*) in two iterations with the pair $(y^3, z^3), (y^5, z^5)$.

In practical applications, one would expect that most of the points $(A_{\cdot j}, c_j)$ would lie above the underbelly of their convex hull. We would therefore expect that *very few* j would be extreme points of the underbelly. Since the algorithm only chooses $(A_{\cdot s}, c_s)$ from among the latter and these typically would be rare, I conjectured that the algorithm would have very few choices and would take about m steps in practice.

It is, of course, not difficult to construct cases that take more than m iterations so let me make some remarks about the rate of convergence of z_t to z^*, the minimum value of z, in the event that the method takes more than m interations.

Convergence Rate of the Simplex Method

Assume there exists a constant $\bar{\theta} > 0$ such that for every iteration τ, the values of all basic variables $x^{\tau}_{j_i}$ satisfy

$$x^{\tau}_{j_i} \geq \bar{\theta} > 0 \qquad \text{for all } j_i, \tag{13}$$

At the start of iteration t, by eliminating the basic variables from the objective equation, we obtain

$$z_{t-1} - z = \sum(-\bar{c}^t_j)x_j \tag{14}$$

where $\bar{c}^t_{j_i} = 0$ for all basic $j = j_i$. If $(-\bar{c}^t_s) = \max(-\bar{c}^t_j) \leq 0$, the iterative process stops with the current basic feasible solution optimal. Otherwise, we increase non-basic x_s to $x_s = \theta_t \geq \bar{\theta}$ and adjust basic variables to obtain the basic feasible solution to start iteration $t+1$.

Let $z^* = \min z$ and $x_j = x^*_j \geq 0$ be the corresponding optimal x_j. We define $\Delta_t = z_t - z^*$.

Theorem. *Independent of n the number of variables,*

$$(\Delta_t/\Delta_0) \leq (1-\theta_1)(1-\theta_2)\cdots(1-\theta_t) \leq e^{-\Sigma\theta_\tau} \leq e^{-\bar{\theta}\cdot t}. \tag{15}$$

where $\theta_t \geq \bar{\theta} > 0$ is the value of the incoming basic variable x_s on iteration t.

Proof.

$$\Delta_{t-1} = z_{t-1} - z^* = \sum(-\bar{c}^t_j)x^*_j \leq (-\bar{c}^t_s)\sum x^*_j = (-\bar{c}^t_s). \tag{16}$$

$$\Delta_{t-1} - \Delta_t = z_{t-1} - z_t = (-\bar{c}^t_s)x_s = (-\bar{c}^t_s)\theta_t \geq \Delta_{t-1}\cdot\theta_t\,, \tag{17}$$

where the inequality between the last two terms is obtained by multiplying (16) by θ_t. Rearranging terms,

$$\Delta_t \leq (1-\theta_t)\Delta_{t-1} < e^{-\theta_t}\Delta_{t-1} \leq e^{-\bar{\theta}}\Delta_{t-1} \tag{18}$$

and (15) follows. ∎

Corollary. *Assuming θ_r has "on the average" the same average value as any other $x^r_{j_i}$, namely $(1/m)$, then the expected number of iterations t required to affect an e^{-k} fold decrease in Δ_0 will be less than km iterations, i.e.*

$$(\Delta_t/\Delta_0) < e^{-\Sigma\theta_r} \doteq e^{-t/m} . \tag{19}$$

Thus, under the assumption that the value of the incoming variable is $1/m$ on the average, a thousand-fold decrease in $\Delta_t = z_t - z^*$ could be expected to be obtained in less that $7m$ iterations because $e^{-7} < .001$.

It was considerations such as these that led me back in 1947 to believe that the simplex method would be very efficient.

It is fortunate back in 1947 when algorithms for solving linear programming were first being developed, that the column geometry and not the row geometry was used. As we have seen, the column geometry suggested a very different algorithm, one that promised to be very efficient. Accordingly, I developed a variant of the algorithm without the convexity constraint (9) and arranged in the fall of 1947 to have the Bureau of Standards test it on George Stiegler's nutrition problem [14]. Of course, I soon observed that what appeared in the column geometry to be a new algorithm was, in the row geometry, the vertex descending algorithm that I had rejected earlier.

It is my opinion that any well trained mathematician viewing the linear programming problem in the row geometry of the variables would have immediately come up with the idea of solving it by a vertex descending algorithm as did Fourier, de la Vallée Poussin, and Hitchcock before me — each of us proposing it independently of the other. I believe, however, that if anyone had to consider it as a practical method, as I had to, he would have quickly rejected it on intuitive grounds as a very stupid idea without merit. My own contributions towards the discovery of the simplex method were (1) independently proposing the algorithm, (2) initiating the development of the software necessary for its practical use, and (3) observing by viewing the problem in the geometry of the columns rather than the rows that, contrary to geometric intuition, following a path on the outside of the convex polyhedron, might be a very efficient procedure.

The Role of Sparsity in the Simplex Method

To determine $s = \arg\min_j[c_j - (\pi A_{.j} + \pi_0)]$ requires forming the scalar product of two vectors π and $A_{.j}$ for each j. This "pricing out" operation as it is called is usually very cheap because the vectors $A_{.j}$ are sparse, i.e., they typically have few non-zero coefficients

(perhaps on the average 4 or 5 non-zeros). Nevertheless if the number of columns n is large, say several thousand, pricing can use up a lot of CPU time. (Parallel processors could be used very effectively for pricing by assigning subsets of the columns to different processors, [18].)

In single processors, various *partial pricing* schemes are used. One scheme used in MINOS software system is to partition the columns into subsets of some k columns each, [12]. The choice of s is restricted to columns that price out negative among the first k until there are none and then moving on to the next k, etc. Another scheme used is to price out all the columns and rank them as to how negative they price out. A subset of j, say the fifty most negative in rank, are then used to iteratively select s until this subset no longer has a column that prices out negative. Then a new subset is generated for selecting s and the process is repeated. The use of partial pricing schemes are very effective when n is large especially for matrix structures that contain so called "GUB" (Generalized Upper Bound) rows, [13].

Besides the pricing-out of the columns, the simplex method requires that the current basis B, i.e. the columns $(j_1, \ldots, j_m)$ used to form the simplex in Figure 1 be maintained from iteration t to $t+1$ in a form that makes it easy to compute two vectors v and π where $Bv = A_{\cdot s}$ and $\pi B = (c_{j_1}, \ldots, c_{j_m})$. The matrix B is typically very sparse. In problems where the number of rows $m > 1{,}000$, the percent of non-zeros may be less than $\frac{1}{2}$ of one percent. Even, for such B, it is not practical to maintain B^{-1} explicitly because it could turn out to be 100% dense. Instead B is often represented as the product of a lower and upper triangular matrix where each is maintained as a product of elementary matrices with every effort being made to to keep the single non-unit column of these elementary matrices as sparse as possible. Maintaining this sparsity is important for otherwise the case of $m = 1{,}000$ the algorithm would have to manipulate data sets with millions of non-zero numbers. Solving systems $Bv = A_{\cdot s}$ in order to detrmine which variable leaves the basis would become too costly.

The Role of Near Triangularity of the Basis

The success of the simplex method in solving very large problems encountered in practice depends on two properties found in almost every practical problem. First, the basis is usually very sparse. Second, one can usually rearrange the rows and columns of the various bases encountered in the course of solution so that they are nearly triangular. Near triangularity makes it a relatively inexpensive operation to represent it as a product of a lower and upper triangular matrices and to preserve much of the original sparsity.

Even if the bases were very sparse but not nearly triangular, solving systems $Bv = A_{\cdot s}$ could be too costly to perform.

The success of solving linear programming therefore depends on a number of factors: (1) the power of computers, (2) extremely clever algorithms; but it depends most of all upon (3) a lot of good luck that the matrices of practical problems will be very very sparse and that their bases, after rearrangement, will be nearly triangular.

For forty years the simplex method has reigned supreme as the preferred method for solving linear programs. It is historically the reason for the practical success of the field. As of this writing, however, the algorithm is being challenged by new interior methods proposed by N. Karmarkar [15] and others, and by methods that exploit special structure. If these new methods turn out to be more successful than the simplex method for solving certain practical classes of problems, I predict it will not be because of any theoretical reasons having to do with polynomial time but because they can more effectively exploit the sparsity and near triangularity of practical problems than the simplex method is able to do.

References

[1] G.B. Dantzig, "Reminiscences about the Origins of Linear Programming," *Mathematical Programming* (R.W. Cottle, M.L. Kelmanson, B. Korte, eds.), Proceedings of the International Congress on Mathematical Programing, Rio de Janeiro, 1984, pp. 105-112.

[2] T.C. Koopmans (ed). *Actively Analysis of Production and Allocation*, John Wiley & Sons, Inc., New York, 1951, 404 pages.

[3] A.J. Hoffman, M. Mannos, D. Sokolousky, and N. Wiegmann, "Computational Experience in Solving Linear Programs," *J. Soc. Indus. Appli. Math.*, Vol. 1, No 1, 1953, pp. 17-33.

[4] T.S. Motzkin, "Beitrage zur Theorie der Linearen Ungleichungen," Jerusalem, 1936 (Doctoral Thesis, University of Zurich).

[5] J.B.J. Fourier, "Solution d'une question particuliere du calcul des inegalities," original 1826 paper with an abstract of an 1824 paper reprinted in Oeuvres de Fourier, Tome II Olms, Hildesheim 1970, pp. 317-319.

[6] Ch. de la Vallée Poussin, "Sur la Methode de l'approximation minimum," *Annales de la Societe de Bruxelles*, 35 (2) 1910-1, pp. 1-16.

[7] L.V. Kantorovich, "Mathematical Methods in the Organization and Planning of Production," Publication House of the Leningrad State University, 1939, 68 pp. Translated in *Management Science*, Vol. 6, 1960, pp. 366-422.

[8] F.L. Hitchcock, "The Distribution of a Product from Several-Sources to Numerous Localities," *J. Math. Phys.*, Vol. 20, 1941, pp. 224-230.

[9] G.B. Dantzig and A. Wald, "On the Fundamental Lemma of Neyman and Pearson," *Ann. Math. Statist.*, Vol. 22, 1951, pp. 87-93.

[10] G.B. Dantzig, "Linear Programming," in *Problems for the Numerical Analysis of the Future*, Proceedings of Symposium on Modern Calculating Machinery and Numerical Methods, UCLA, July 29-31, 1948, Appl. Math., Series 15, National Bureau of Standards, June 1951, pp. 18-21.

[11] G.B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ, 1963, 627 pages.

[12] B.A. Murtagh and M.A. Saunders, "MINOS 5.0 User's Guide" Technical Report SOL 83-20, Systems Optimization Laboratory, Department of Operations Research, Stanford University, 1983.

[13] G.B. Dantzig and R.M. Van Slyke, "Generalized Upper Bounding Techniques", *J. Computer and System Sciences*, Vol 1, No. 3, October 1967, pp. 213-226.

[14] G.J. Stigler, "The Cost of Subsistence", *J. of Farm Econ.*, Vol. 27, No. 2, May 1945, pp. 303-314.

[15] N. Karmarkar, "A New Polynomial Algorithm for Linear Programming," *Combinatorica*, Vol. 4, 1984.

[16] R. Coughlin and D.E. Zitarelli, *The Ascent of Mathematics*, McGraw-Hill, 1984, p. 265.

[17] G.B. Dantzig, "Impact of Linear Programming on Computer Development", Technical Report SOL 85-7, Department of Operations Research, Stanford University, Stanford, June 1985.

[18] G.B. Dantzig, "Planning Under Uncertainty Using Parallel Computing," Technical Report SOL-87-1, Department of Operations Research, Stanford University, Stanford, January 1987.

[19] J. von Neumann, "Zur Theorie de Gesellschaftsspiele," *Math. Ann.*, Vol. 100, 1928, pp. 295-320. Translated by Sonya Bargmann in A.W. Tucker and R.D. Luce (eds), *Contributions to the Theory of Games*, Vol IV, *Annals of Mathematics* Study No. 40, Princeton University Press, Princeton, New Jersey, 1959, pp. 13-42.

[20] J. Neyman and E.S. Pearson, "Contributions to the Theory of Testing Statistical Hypotheses," *Statist. Res. Mem.*, Parts I and II, 1936, 1938.

A PERSONAL RETROSPECTION OF RESERVOIR SIMULATION

Donald W. Peaceman
Annultant, Exxon Production Research Company

I plan to talk about some early history of reservoir simulation, from a very personal point of view. I'll give some history, some philosophy, and some numerical analysis. I will try to stress the interrelationship between the type of computing equipment that we had available at any given time, the kinds of calculations we were able to make, and the kinds of problems we were able to solve. So, if you'll indulge me, we'll take a look at what computing and numerical analysis were like twenty to thirty-five years ago, in the not so good old days.

I started in 1951 with Humble Oil and Refining Company, which at that time was a subsidiary of Standard Oil of New Jersey. Standard Oil of New Jersey became Exxon, and the Research Division of Humble Oil evolved into the present day Exxon Production Research Company.

When I came to work in 1951, we didn't have any real computers available to us. Yet there was some reservoir modelling going on. I found some old pictures that illustrate how physical models were used.

Fig. 1 shows the earliest one that I found. It was made in 1933 and it shows a sand-packed model that was used to study water coning. On top is an oil layer, with a water layer underneath it. You can see that wells were drilled just into the upper oil layer. The production of oil causes the pressure around the well to decrease, and that causes the water to cone up and be produced with the oil. Though oil fields have been produced since 1860, it wasn't until the 1930's that people in the oil industry started looking at reservoir mechanics in any kind of a scientific way. So this was one of the first attempts to understand why water starts to be produced with oil and why the produced water-oil ratio increases with time.

Fig. 2 shows another sand-packed model also used to study coning, some 25 years later -- still, before computers took over. This model was somewhat more sophisticated. You can't see it, but it was wedge- shaped, to take into account the radial geometry around a well.

The analogy between electric current flow and Darcy flow through sand had been recognized for quite a while, and electrolytic models were used in the late thirties and the forties to solve Laplace's equation for various geometries. Fig. 3 shows an example of how elaborate an electrolytic model could get. This was a model of the East Texas Field, which is still one of the largest fields in the United States. The model was made of plastic and covered with an electrolyte solution. The plastic was contoured to represent the shape and permeability distribution in the field -- with the depth of the solution above the plastic being proportional to the thickness times permeability that was actually measured in the field. The object was to measure the potential distribution in the field in order to predict the water influx from the aquifer surrounding the field.

But electrolytic models were steady-state models. To get a better representation of unsteady-state flow, the reservoir analyzer shown in Fig. 4 was devised, involving a scaled electrical network of resistors and capacitors. Voltages represented pressure; current flow represented fluid flow; resistors corresponded to permeability times thickness, and the capacitors corresponded to porosity times thickness times compressibility. With this representation, unsteady-state compressible flow could be taken into account. The electrical network corresponded, of course, to a finite-difference equation solved continuously in time.

In addition to these physical analog models, some mathematical methods were available in 1951. In the thirties and forties, three authors made the most significant contributions to applying the methods of mathematical physics to reservoir engineering. Muskat, of Gulf, wrote a book in 1937 [1] that summarized his work -- and that book is still very useful. Hurst [2,3] at Humble, and later Hurst and van Everdingen [4] at Shell also made significant contributions. Their methods were based primarily on infinite series solutions to Laplace's equation and the heat conduction equation. While these methods were very elegant, they suffered from serious limitations in their application to real reservoir problems -- they assumed uniform properties and ideal geometries, and could only be used where the differential equations are linear. And also, these methods

required the tedious evaluation of infinite series, which had to be computed by hand.

So that was the state of reservoir modeling when I came to work at Humble in 1951. We had nothing that you could call a computer. We did

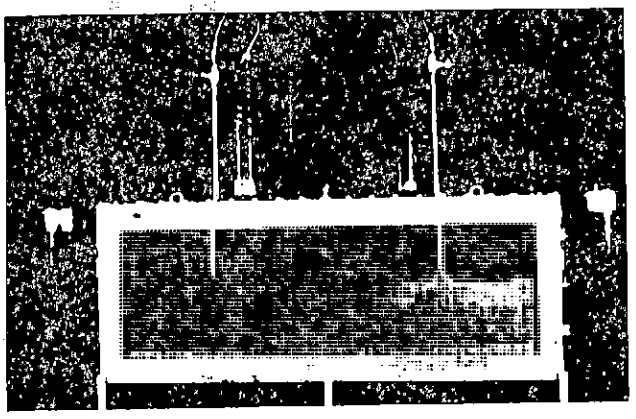


Fig. 1. Early Study of Water Coning (ca 1933)

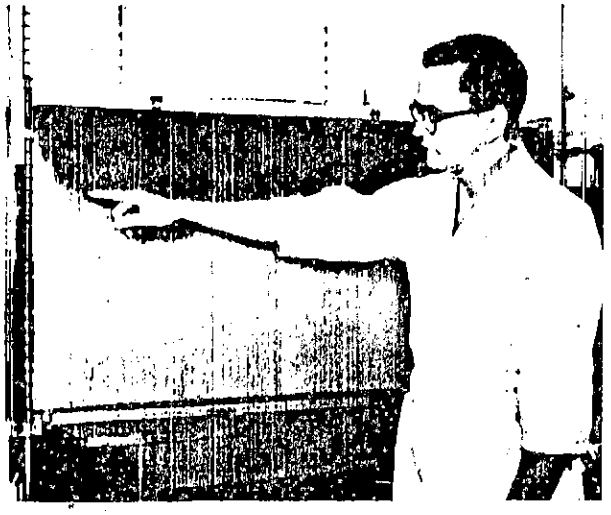


Fig. 2. Study of Water Coning (ca 1958)

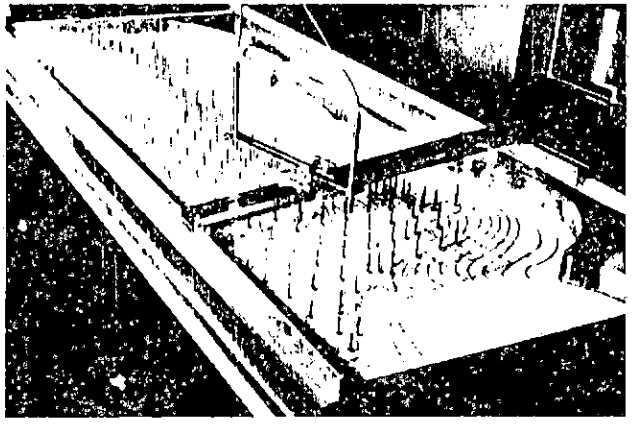


Fig. 3. Electrolytic Model of East Texas Field

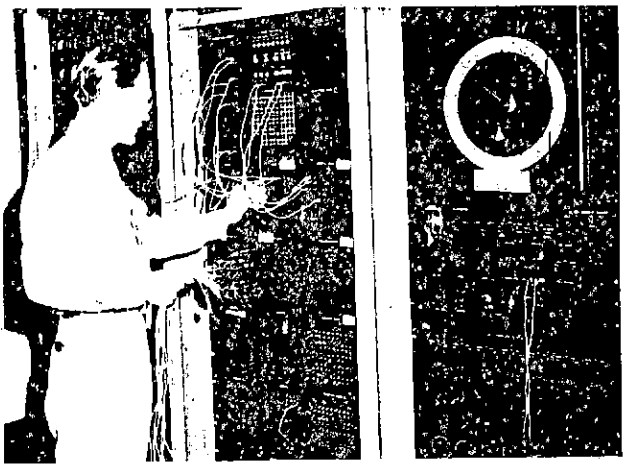


Fig. 4. Electrical Network Reservoir Analyzer

have access to some accounting machines that the accounting department would let us use, but only at night. Henry Rachford had come to work a year before me, and was already playing with an accounting machine called the IBM 604. He, along with the managers of the Production Research Division of Humble, had the vision to see that digital computation was going to be the way to do reservoir modelling and that, by using finite-difference methods to solve partial differential equations, we could overcome the limitations of the analytical methods. We wanted to be able to include nonuniform properties, arbitrary geometry, and nonlinearities in the differential equations.

But that vision was still pretty faint. The first partial differential equation that we tried to solve was one-dimensional gas flow, and the first limitation that we were trying to overcome was that of nonlinearity. If one assumes a perfect gas, then the equation for linear one-dimensional gas flow is

$$\frac{\partial^2 p^2}{\partial x^2} = \frac{2\phi\mu}{K}\frac{\partial p}{\partial t} . \qquad (1)$$

It looks a lot like the linear one-dimensional heat conduction equation, except that the second derivative term has p^2 in it instead of p, making that equation nonlinear. Because of that, there is no known analytical solution. The initial condition is uniform pressure; at one end is a fixed production rate, q, giving the nonlinear boundary condition:

$$q = \frac{KA}{\mu RT} p \frac{\partial p}{\partial x} , \qquad x = 0 . \qquad (2)$$

At the other end, the system is closed, so we have a no-flow boundary condition, with a zero derivative.

Of more practical interest was the radial problem, corresponding to the depletion of a circular gas reservoir with a well at the center.

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r \frac{\partial p_\zeta^2}{\partial r} \right) = \frac{2\phi\mu}{K}\frac{\partial p}{\partial t} . \qquad (3)$$

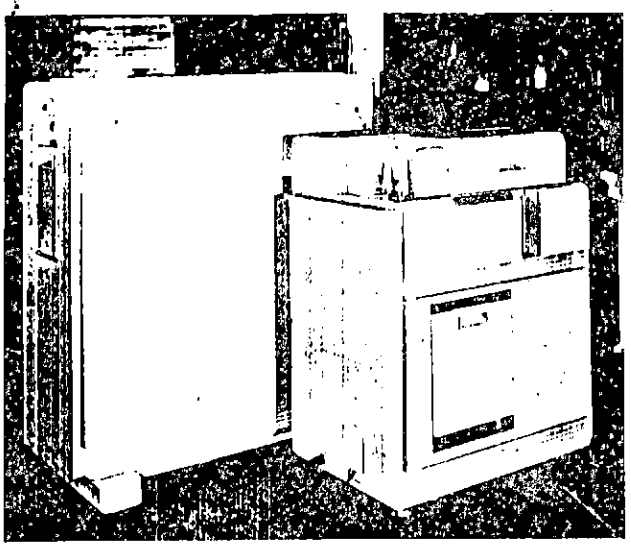


Fig. 5. IBM 604 Multiplying Punch

The initial and boundary conditions are similar.

When I came to work, Henry Rachford and John Rice were already at work on this problem, trying to use the accounting machine, the IBM 604. (See Fig 5.) Let me try to describe this gadget to you. It was called a multiplying punch -- the only input/output that it had was a card reader and card punch. It could only handle fixed-point decimal numbers, with no alphabetic information. The way the accountants used it, say for a payroll application, would be to have some numeric data already punched on each card, such as an employee's identification number and his salary. Each card would be read at the read station, then it would travel to the punch station. On the way, the marvelous electronic multiply unit would calculate the withholding and social security taxes, subtract them, and then punch into the blank space on the card the taxes and the take-home pay. After being punched, the card would then travel to the stacker. To see the results printed, an operator would have to carry the deck of cards to another machine to print the results.

The electronic multiply unit was really quite flexible, but programming it was done in a way that we would now consider very quaint. There was a board with a lot of holes, and this board could be placed into a holder with terminals in the back. Programming was done by plugging wires from one hole to another. Down one side of the board was a series of holes called program steps. On the other side were holes for various functions, such as reading from a card into electronic registers, adding or multiplying the contents of two registers together, and punching the contents of the registers onto the card. Remember that in those days, all the electronics was done with vacuum tubes.

This device was certainly unsuitable for scientific computations, yet it was all we had. John Rice and Henry Rachford programmed it to solve the one-dimensional gas flow problem. They were already familiar with the paper by O'Brien, Hyman and Kaplan [5], published in 1951, which discussed the finite-difference solution to the linear heat conduction problem. That paper introduced us to the von Neumann stability analysis, as well as to the use of implicit equations. From the stability analysis, they knew they could not use an explicit method for the radial problem, so they attempted to solve it by this implicit equation:

$$\frac{p_{I-1}^2 - 2p_I^2 + p_{I+1}^2}{\Delta x^2} = K\,\frac{p_I - p_I^{old}}{\Delta t} \quad . \qquad (4)$$

We were pretty naive in those days, so they attempted to solve this equation sequentially from left to right using

$$p_{I+1}^2 = 2p_I^2 - p_{I-1}^2 + \frac{K\,\Delta x^2}{\Delta t}\,(p_I - p_I^{old}) \quad . \qquad (5)$$

This required guessing the slope at the well, and seeing if the slope comes out to be zero at the closed end. If not, adjust the initial slope and try again (a shooting method). We found out the hard way that this sequence of calculations from one end to the other is unstable, and must blow up. In retrospect, this is obvious from an error analysis. But as I said, we were pretty naive, so that was one of our first experiences with an unstable calculation.

The fix, of course, is to solve for all of the pressures at all of the nodes simultaneously. We saw how to do that, when we came upon an unpublished note by L. H. Thomas, of IBM. In that note, he outlined what we now know as the tridiagonal algorithm. I believe our paper on gas flow was one of the first to present this algorithm in the published literature.

In order to use this algorithm, the nonlinear difference equation (4) had to be linearized. We did that by factoring the second-difference term into the form

$$\frac{(p_{I-1}^k + p_I^k)(p_{I-1}^{k+1} - p_I^{k+1}) - (p_I^k + p_{I+1}^k)(p_I^{k+1} - p_{I+1}^{k+1})}{\Delta x^2} = K\,\frac{p_I^{k+1} - p_I^{old}}{\Delta t} \quad , \qquad (6)$$

and iterating on each time step. While the iteration converged only linearly, it did converge very rapidly -- usually five iterations were sufficient. At that time, we were not aware of the Newton-Raphson method, which would have given quadratic convergence.

By the time we had this new approach worked out, we were onto our next machine. It was our own machine now, and not one that we had to borrow. This was the IBM CPC, or Card-Programmed Calculator, shown in Figs. 6 and 7. That's Henry Rachford. IBM didn't really develop the C.P.C. Several computing groups at various aircraft companies modified and hooked together some existing IBM accounting machines. IBM adopted it and marketed it. It was a real kludge. The 418 (in the foreground of Fig. 6) was an electromechanical accounting machine that could read cards, perform simple additions and subtractions, and print results at 150 lines per minute. It had

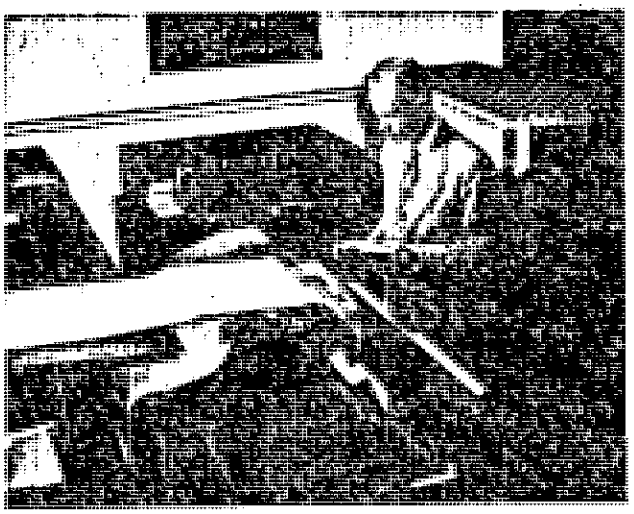


Fig. 6. IBM Card-Programmed Calculator (CPC)

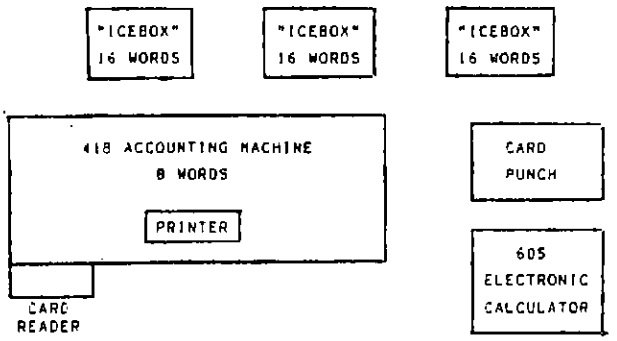


Fig. 7. Components of IBM CPC

the capacity to store eight ten-digit numbers. The box (at the rear) was the card punch. The 605 (at the right rear) was an electronic calculator, an extension of the 604 that I discussed before. All of these required the wiring of large boards. In addition, there were three boxes (not shown in Fig. 6) that we called ice boxes, that could each hold 16 ten-digit numbers in electromechanical counter wheels, like the odometer on a car. We could open the top and actually read out the numbers while debugging. All of this was decimal. It was designed to be fixed-point, but Henry and I wired the machines to do floating-decimal arithmetic, two floating-point operations per card, so that we achieved the magnificent rate of five floating-point operations per second. We could store a total of 56 numbers. This was not a stored-program machine -- instead the program had to be punched onto cards -- so, in effect, we had an unlimited amount of storage available for programs, but only 56 words available for temporary data.

This, then, was the device on which we solved the gas flow problem, for both the linear and radial case [6]. Iteration was carried out by reading the same deck of program cards over and over, and the iteration was monitored by looking at the printed output. When the iteration converged, we then switched to a new deck to start a new time step.

While we were successful in solving the one-dimensional nonlinear problem, we were aware that further progress toward solving realistic field problems would require going to higher dimensions -- at least to two dimensions. At the very least, we knew that we needed to be able to solve the finite-difference analogs of Laplace's equation and the heat conduction equation or, better yet, the variable-coefficient versions of those equations, in an arbitrary geometry. To do this, we had to be able to solve a large system of simultaneous linear equations. Direct solution of these equations by Gaussian elimination was out of the question on the machines that were then available.

We had access in those days to several eminent consultants. One of them was John von Neumann, and he visited with us a couple of times. He was very interested in the work we were doing, but when we asked him how to go about solving two dimensional problems of this sort, he had no more to offer than the so-called extrapolated Liebman method, now known as successive overrelaxation, or SOR. And we already knew about that method.

A breakthrough came, not while we were thinking about solving a problem in x-y coordinates, but, rather, a flow problem in cylindrical coordinates:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial p}{\partial r}\right) + \frac{\partial^2 p}{\partial y^2} = \frac{\partial p}{\partial t} \quad . \tag{7}$$

If we let $x = \ln r$, then the differential equation takes the somewhat simpler form

$$e^{-2x}\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = \frac{\partial p}{\partial t} \quad . \tag{8}$$

Solving this equation implicitly would have had the same difficulty as solving the heat conduction equation implicitly. But the inherent difference between the radial and vertical directions suggested another approach. Suppose we make the difference equation implicit in just one direction, say the radial direction, and explicit in the other, vertical, direction. The half-implicit, half-explicit difference analog would then be

$$e^{-2x_i}\frac{p^{n+1}_{i-1,j} - 2p^{n+1}_{i,j} + p^{n+1}_{i+1,j}}{\Delta x^2} + \frac{p^{n}_{i,j-1} - 2p^{n}_{i,j} + p^{n}_{i,j+1}}{\Delta y^2} = \frac{p^{n+1}_{i,j} - p^{n}_{i,j}}{\Delta t} \quad . \tag{9}$$

The advantage, of course, is that on each line, we just have a tridiagonal system of equations, which is very easy to solve. THe von Neuman stability analysis is very simple to do -- we substitute this Fourier representation,

$$p^{n}_{i,j} = \gamma_n e^{i\alpha x_i} e^{i\beta y_j} \quad , \tag{10}$$

into the difference equation, and examine the growth of γ. The amplification factor for γ is

given by the ratio

$$\frac{\gamma^{n+1}}{\gamma^{n}} = \frac{1 - 4(\Delta t/\Delta y^2)\sin^2(\beta\Delta y/2)}{1 + e^{-2x_1}(\Delta t/\Delta x^2)\sin^2(\alpha\Delta x/2)} \quad . \tag{11}$$

For the difference equation to be stable, the magnitude of this ratio has to be less than one for all α and β. Unless Δt is very small, that won't be true, so we have here a difference equation that is not much better than a fully explicit one.

Suppose we do the opposite. Make it explicit in the radial direction, and implicit in the vertical direction.

$$e^{-2x_1}\frac{p^{n}_{I-1,J} - 2p^{n}_{I,J} + p^{n}_{I+1,J}}{\Delta x^2} + \frac{p^{n+1}_{I,J-1} - 2p^{n+1}_{I,J} - p^{n+1}_{I,J+1}}{\Delta y^2} = \frac{p^{n+1}_{I,J} - p^{n}_{I,J}}{\Delta t} \quad . \tag{12}$$

In that case, we would get the following ratio for the amplification factor,

$$\frac{\gamma^{n+1}}{\gamma^{n}} = \frac{1 - e^{-2x_1}(\Delta t/\Delta x^2)\sin^2(\alpha\Delta x/2)}{1 + 4(\Delta t/\Delta y^2)\sin^2(\beta\Delta y/2)} \quad , \tag{13}$$

and again, unless Δt is sufficiently small, this ratio will be bigger than one in magnitude for some α and β.

Somehow, and we don't remember exactly how, but it seemed natural enough, Henry Rachford and I came up with the idea of doing it one way for one time step, and then the other way for the next time step -- a two-step procedure. Then, repeat the two-step procedure over and over.

It wasn't immediately obvious that this would be stable. But we analyzed it independently overnight, and came to the same conclusion, that it is stable. It is necessary to recognize that you want to look at the amplification factor, not for either step alone, but for the entire process of going from step n to step n+2. Now the second ratio, (13), is really $\gamma^{n+2}/\gamma^{n+1}$, and we can multiply the two ratios (11) and (13) together, and rearrange, to get

$$\frac{\gamma^{n+2}}{\gamma^{n}} = \frac{1 - e^{-2x_1}(\Delta t/\Delta x^2)\sin^2(\alpha\Delta x/2)}{1 + e^{-2x_1}(\Delta t/\Delta x^2)\sin^2(\alpha\Delta x/2)} \cdot \frac{1 - 4(\Delta t/\Delta y^2)\sin^2(\beta\Delta y/2)}{1 + 4(\Delta t/\Delta y^2)\sin^2(\beta\Delta y/2)} \quad . \tag{14}$$

Now, a remarkable thing happens. The first ratio is always less than one in magnitude, no matter what the values of Δt, Δx, or α are. Similarly, the second ratio is always less than one in magnitude, no matter what the values of Δt, Δy, or β are. Hence, the product must be less than one in magnitude, and the two-step procedure must be stable. So that was how alternating direction was born.

Henry and I remember well the date of this discovery, December 30th and 31st, 1953. The reason we remember it so well is that we celebrated New Year's Eve at our house, along with Jim Douglas and his wife. Naturally we were very excited, and could hardly talk about anything else. This shop talk was very distressing to the hostess, my wife. I think she finally forgave us a few years later.

There were several implications to the discovery that were immediately apparent. First, of course, was the fact that the asymmetry of the cylindrical problem had nothing to do with the success of the method, even though that was what triggered the idea. In particular, of course, it could be applied directly to the heat conduction problem in ordinary x-y coordinates.

The second implication, of even greater significance, was the fact that the alternating-direction method can also be used to solve a steady-state problem. The solution to Laplace's equation, is, after all, the solution to the heat conduction equation at infinite time. We can imagine that if we take enough time steps, we will get the solution to Laplace's equation. We can think of accelerating the process by taking some short time steps, and then some longer ones, and then if we're not close enough to the solution, repeat the sequence of short and long time steps. What that amounts to, of course, is nothing more than using alternating direction as an iterative method, with Δt serving as an iteration parameter. Well, Jim Douglas ran with that idea. He carried out an analysis that permits one to calculate an almost optimum sequence of iteration parameters. He also demonstrated convergence of the A.D.I. method to the solution of the heat conduction problem. His results were published in a companion paper [7] to the paper that Rachford and I published in the SIAM Journal early in 1955 [8].

The first tests of A.D.I. were on the Card-Programmed Calculator. For the SIAM paper [8], we solved both an unsteady-state and a steady-state problem on a 14 by 14 square grid. Why 14 by 14? Well, we had 56 words of data storage. As we calculated for each line, we needed to keep four numbers internally for each point. Thus, the longest line we could handle was 14. Most of the temporary data storage was punched out onto cards, and the direction was alternated by using a card sorter. Because the data that was punched onto the cards had to be read back in in reverse order, we punched the cards backwards and upside down, then turned them over, in order to facilitate the sorting process.

Jim Douglas and I wrote a second paper on A.D.I. [9], in which we solved some steady-state problems on geometries other than a square. These were also done on the C.P.C. The first one, shown in Fig. 8, was for heat flow around a corner, with the temperatures zero and one at two boundaries, and with no-flow boundaries elsewhere. Also shown is the grid numbering scheme that we used, where the numbers correspond to the register numbers on the C.P.C.

Fig. 9 shows a problem involving radiation from a square pipe. The inside of the pipe is at temperature T_1; the outside of the pipe has a nonlinear radiation boundary condition. We took advantage of the symmetry to solve the system in one-eighth of the cross-section. The computing grid is also shown.

The third problem, in Fig. 10, was more related to reservoir engineering. We assumed an elliptical reservoir, with no flow at the external

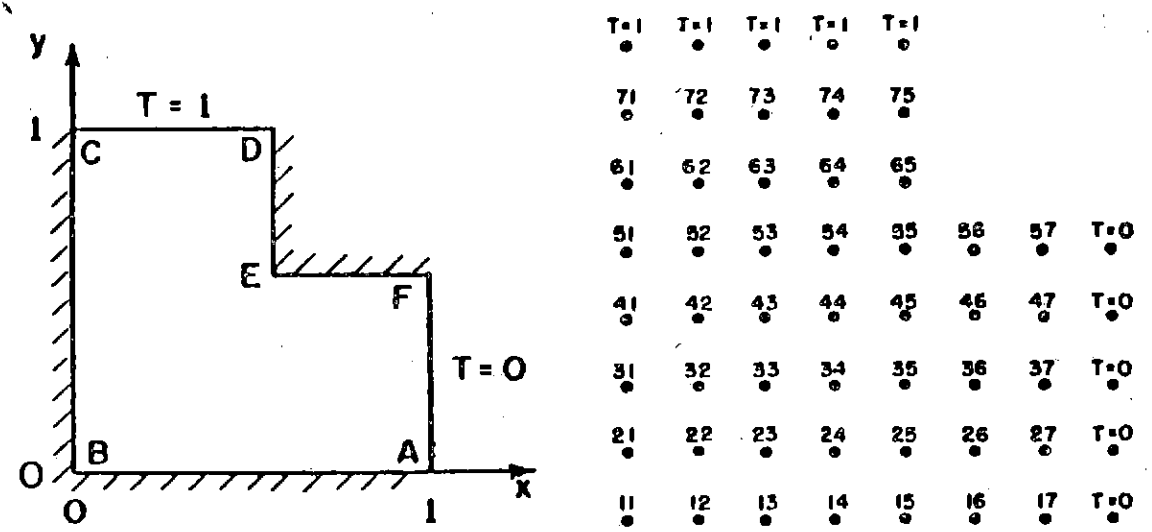


Fig. 8. Heat Flow Around Corner

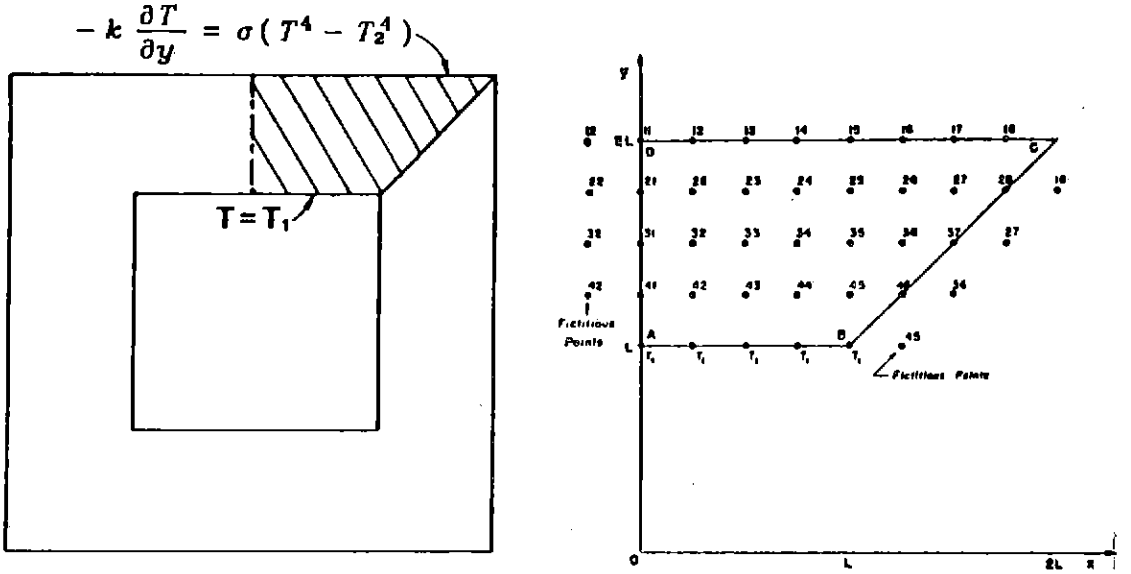


Fig. 9. Square Pipe with Radiation

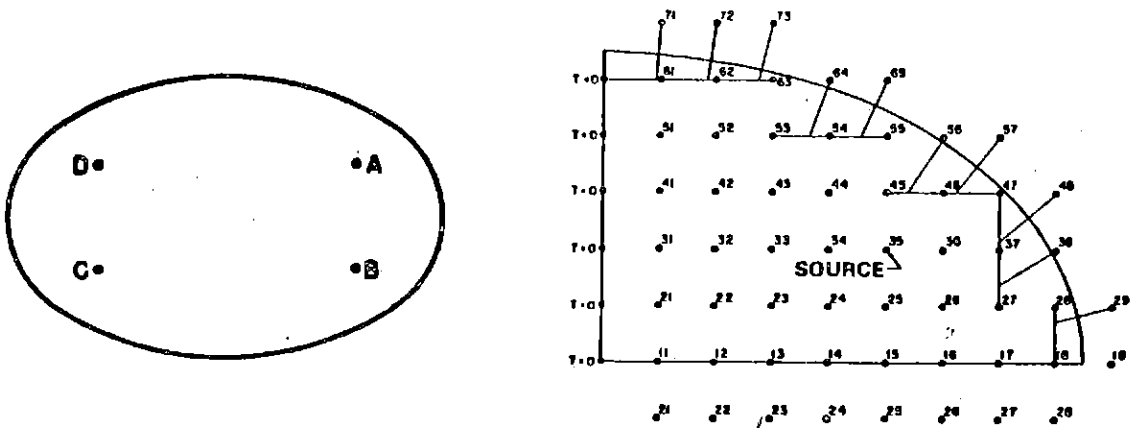


Fig. 10. Flow in Elliptical Reservoir

boundary, with input wells at points A and B, and output wells at points C and D. Again, we took advantage of symmetry, and solved only one-quarter of the system. The computing grid for that problem is also shown. In all of these cases, the longest line was eight points long, which turned out to be very convenient on the C.P.C.

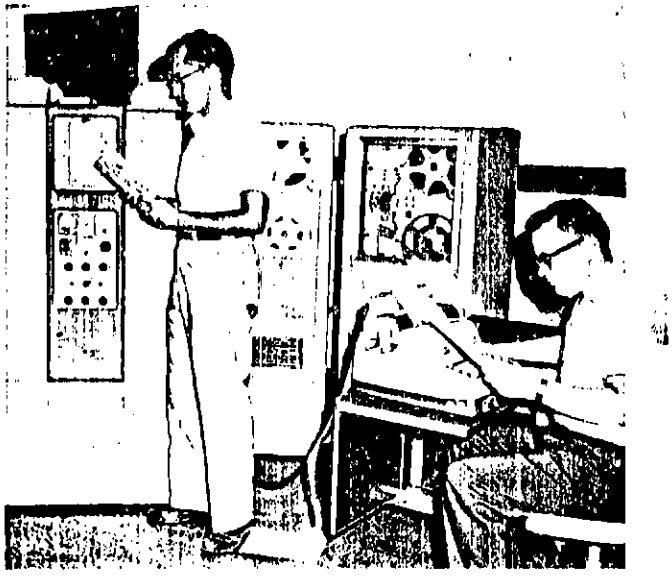


Fig. 11. Bendix G-15 Drum Computer

In 1955 we acquired a Bendix G-15, shown in Fig. 11. This also had vacuum tube electronics, but its storage was almost completely on a magnetic drum. It came with fixed-point binary arithmetic, which was of limited scientific use, so I spent several months programming a floating-point interpreter for it. With that, we had the fantastic capability of doing ten floating-point operations per second. It had 864 words of memory available, for both data and program. You can see that its input/output was paper tape, typewriter, and magnetic tape, none of which were particularly reliable by today's standards.

In addition, within the next few years, we started using IBM's first widely used scientific computer, the 704, shown in Fig. 12. It was a binary machine, with built-in floating-point hardware. Its electronics was based on thousands of vacuum tubes; its central memory was magnetic

Fig. 12. IBM 704

TABLE I. Humble/EPR Computing Equipment

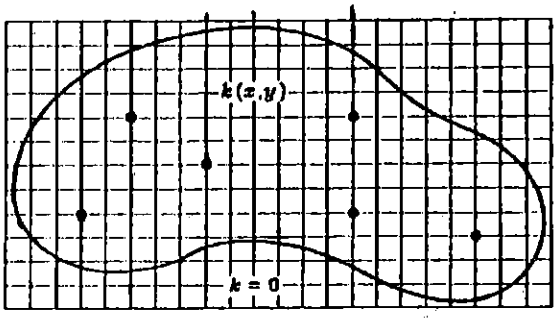
	Date Acquired	Storage(words)	Speed
IBM 604	Before 1950	8(1) + Cards	
IBM CPC	1952	56 + Cards	5 FLOPS
BENDIX G-15	1955	864	10 FLOPS
IBM 704	1956 (away)	8,192	10,000 FLOPS
BENDIX G-20	1961	8,192	20,000 FLOPS
IBM 7040/7044	1962	16,384	40,000 FLOPS
IBM 360/65	1967	256 K	400,000 FLOPS
IBM 370/165	1971	1 M	1 MFLOPS
IBM 370/168	1975	2 M	1.2 MFLOPS
AMDAHL V8	1978	4 M	2 MFLOPS
IBM 3033	1979	4 M	2 MFLOPS
IBM 3081	1982	4 M	6 MFLOPS
CRAY 1S	1982	4 M	20-160 MFLOPS

core; its secondary storage was magnetic tape. We never acquired a 704 of our own. The first one that we used was at the IBM Service Center in New York City, starting about 1956; after several years we started using 704's at various aircraft companies throughout the country that were selling excess time.

As you can see from Table I, the 704 marked a major advance in our computing capability, and with it we were able to solve our first real reservoir problem. That involved solving this variable coefficient version of the steady-state Laplace's equation:

$$\frac{\partial}{\partial x}\left[\, k(x,y)\, \frac{\partial p}{\partial x}\,\right] + \frac{\partial}{\partial y}\left[\, k(x,y)\, \frac{\partial p}{\partial y}\,\right] = q(x,y). \qquad (15)$$

k was the value of permeability times thickness, which was known as a function of x and y. See Fig. 13. As we still do today, the shape of the



1) 1500 POINTS (50 x 30)
2) 8000 POINTS (100 x 80)

Fig. 13. First Field Problem

$$-\nabla\cdot\left[\frac{K\,k_{ro}}{\mu_o} f \frac{dP_c}{dS_w}\nabla S_w\right] - \vec{v}_t \frac{df}{dS_w}\cdot\nabla S_w = \phi\frac{\partial S_w}{\partial t} . \tag{21}$$

On the face of it, (21) looks parabolic, but the first term involves P_c, the capillary pressure, and it usually is small. The second term is the convection term, with velocity times a first order derivative of saturation, and it dominates. So (21) is really almost first-order hyperbolic in nature.

It wasn't until much later that we realized that the appropriate differential equation to analyze for stability is

$$-v_t\frac{\partial f(S)}{\partial x} = \phi\frac{\partial S}{\partial t} \tag{22}$$

where we have assumed one dimension, and zero capillary pressure.

For midpoint weighting of relative permeability, the difference equation simplifies to

$$v_t\frac{f^n_{i-1} - f^n_{i+1}}{2\,\Delta x} = \phi\frac{S^{n+1}_i - S^n_i}{\Delta t} . \tag{23}$$

A von Neumann stability analysis shows that (23) is unstable for any size time step. Indeed, people who started using our method discovered empirically that they were getting oscillatory solutions, which could be avoided by using upstream weighting for relative permeability. In upstream weighting, the relative permeability at the upstream grid point is used for each interval between grid points. Within a few years, it became standard practice in the industry to use upstream weighting. In that case, the appropriate difference equation to look at is

$$v_t\frac{f^n_{i-1} - f^n_i}{\Delta x} = \phi\frac{S^{n+1}_i - S^n_i}{\Delta t} . \tag{24}$$

A stability analysis shows that (24) is stable, provided the time step is small enough, according to the criterion,

$$-v_t\frac{\partial f}{\partial S}\frac{\Delta t}{\Delta x} \leq 1 . \tag{25}$$

With the use of upstream weighting, the methods proposed in our paper, and variations on them, became quite popular for the solution of two and three dimensional problems. General-purpose reservoir simulators were developed by a number of companies over the next ten years. But there was one class of problems that these simulators could not handle. These are coning problems, such as the one shown in Fig. 14. Because of the radial geometry and the converging flow, the velocity is very high near the well. For any reasonable time step, inequality (25) is violated, and the calculated water-oil ratio produced into the well oscillates wildly.

About 1970, three papers were published almost simultaneously [11,12,13], that proposed

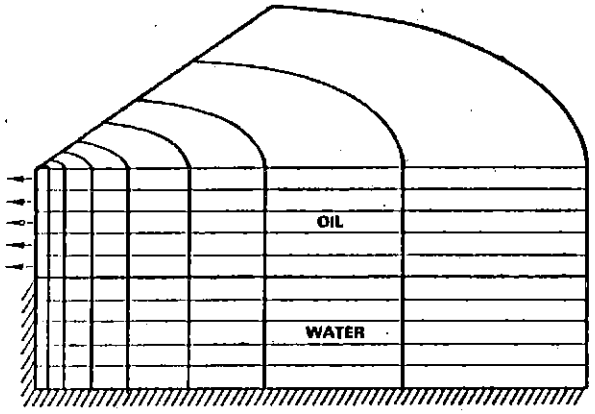


Fig. 14. Coning Problem

essentially the same solution, what we call the semi-implicit approach. Instead of using the old value of relative permeability, an approximation for the new one is used:

$$k_r^{n+1} \cong k_r^n + \frac{dk_r}{dS}(S^{n+1} - S^n) . \tag{26}$$

When this approximation is introduced into the saturation equation, in effect it makes the saturation equation implicit. The equation to analyze now looks like this:

$$v_t\frac{f^{n+1}_{i-1} - f^{n+1}_i}{\Delta x} = \phi\frac{S^{n+1}_i - S^n_i}{\Delta t} . \tag{27}$$

It is stable for any size time step.

It might be useful to look now at how our computing equipment changed over the past twenty five years. Refer again to Table 1. We finally got our own large scale computer in 1961, a Bendix G-20 with 8192 words of core storage. It was twice the speed of the 704, 20,000 floating-point operations per second. Bendix never came through with a Fortran compiler, so we continued to do all our programming in assembly language. We programmed a general-purpose, two-dimensional, two-phase reservoir simulator, with magnetic tape for secondary storage. Fig. 15 shows Henry and me in front of the G-20 tape units, looking at some output.

Bendix finally got out of the computer business. But before they did, we had sent back their G-20, and obtained an IBM 7040, a transistorized version of the 704, again with some increase in speed. Then, in the late sixties, we started with the new IBM 360/370 series, with disc storage and much faster arithmetic speeds. E.P.R. now has several IBM machines, along with the IBM-compatible Amdahl, and you can see that the speed has been increasing significantly into the megaflop range, along with increases in the amount of central memory. E.P.R. now also has a Cray 1-S, with four million words of storage. It is a vector computer, with a theoretical maximum speed of 160 megaflops, although, like most users, we would get a sustained rate in the range of 20 to 40 megaflops. Later versions of the Cray have even higher speeds and larger memories. But now that I'm retired from Exxon, I'm reduced to having

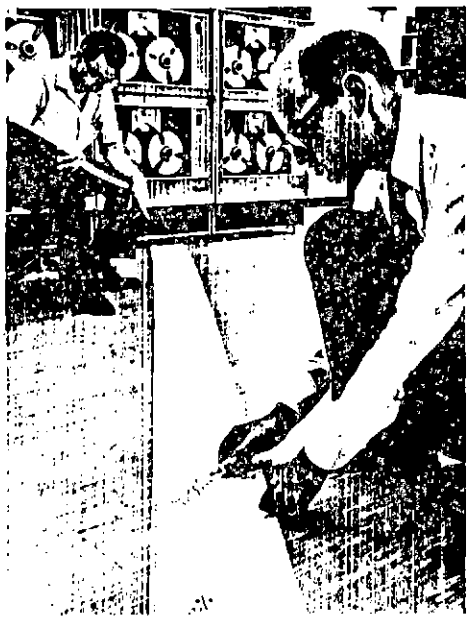


Fig. 15. Bendix G-20 Tape Units

my own personal computer at home. It runs at about 30,000 floating-point operations per second, which puts it where the main-frame computers were 25 years ago. But undoubtedly, we'll see similar increases in speed and memory for personal computers.

One point I would like to make is that in designing a general purpose reservoir simulator, it was never safe to assume that the central random access memory will be big enough. Even though we have seen very large increases in memory size, the computation speed has been going up drastically, while the unit cost of computation has been going down. As a result, the reservoir engineers who use our simulators keep wanting to make their models bigger and bigger, with more and more definition. Or, to put it another way, they tend to run out of central memory before they run out of money. (At least before the price of oil went down.) Consequently, we have found necessary to program our simulators to use secondary storage to supplement the central memory. We used to use magnetic tape for that; now, of course, we use disc storage, which is faster and much more reliable.

But secondary memory is always much slower than central memory, so it's been necessary to learn how to use it in the most efficient way possible. This requires paying attention to the characteristics of the hardware. So you can see, we started out thirty years ago in a very hardware oriented way, and we have never gotten completely away from it. Even with the most modern machines, the effective use of the equipment still requires paying attention to the hardware. And with the advent of vector computers, and other kinds of parallel computers, this has become even more true.

What has happened to the alternating-direction method? Our first paper on two-dimensional immiscible displacement [10] used alternating direction to solve for the two phase pressures on each time step. While this worked fairly well, as we got into more difficult field problems with highly variable permeability distributions, it became more and more difficult to find a sequence of parameters that would make it converge quickly. And frequently it would diverge. In 1968, Herb Stone of Exxon published a new method [14] called SIP, which also requires a sequence of parameters, but it is much more robust, and it is easier to make it work. So, at Exxon, SIP pretty much superceded A.D.I, while other companies tended to go more for successive line overrelaxation. At the present time, the trend is toward preconditioned conjugate gradient methods. However, the search for good iteration methods is far from over, and there is still a lot of research going on in the area of iterative solution of equations.

In earlier times, direct elimination methods were out of the question, but that is no longer true. For two-dimensional problems, they are quite competitive with iterative methods. This also is an active area of research, with people looking particularly at sparse matrix methods, as well as how best to make use of vector computers and other types of parallel machines.

I've just touched on the simplest of the reservoir flow equations that involve the flow of oil, gas, and water. These we now solve routinely, even in three dimensions, with thousands, sometimes tens of thousands, of grid points. But now, the industry is looking more and more at the simulation of enhanced recovery processes, which involve the injection of carbon dioxide, or high pressure nitrogen, or steam, or chemicals, or polymers. The calculations required to simulate these processes are much more demanding, so the needs and the opportunities for research in these areas is tremendous.

I'll say a little bit about numerical methods in general, as applied to reservoir simulation. We started with finite-difference methods twenty five years ago, and they still continue to be used throughout the industry today. There are a number of problems that arise from the use of finite-difference methods. The chief one is probably numerical dispersion, which smears the solution for saturation and concentration. Finite element and other variational methods for solving reservoir problems are being studied by a lot of people, but the results they have obtained have not been impressive enough to cause the industry to stop using finite-difference methods. So the finite-element people continue doing research to try to improve on the finite element methods. However, it seems clear that the finite-difference methods will continue for quite a while to be the mainstay of reservoir simulation. As long as that is the case, there needs to be more research on the finite-difference methods, to understand them better, and to improve on them.

I have tried to do my own little part in the study of finite-difference methods used in reservoir simulation. Three papers illustrate what I've tried to do. The first [15] is a detailed study of the stability of difference equations that use semi-implicit relative permeability. The last two [16,17] discuss how to relate the finite-difference solution for the pressure of a grid block containing a well to the actual pressure at the well itself.

In conclusion, I hope I've conveyed some of the excitement of the early days of reservoir simulation, where we had to fight against the limitations of primitive computing equipment, as well as overcome our naiveness about numerical methods. There are still plenty of challenges left today, and I think they can be just as exciting.

REFERENCES

1. Muskat, M. The Flow of Homogeneous Fluids Through Porous Media. McGraw-Hill Book Co., New York (1937). Reprint edition: International Human Resources Development Corp., Boston (1982).
2. Hurst, W. Unsteady flow of fluids in oil reservoirs. Physics 5, (1934), 71.
3. Hurst, W. Trans. AIME 151, (1943), 57.
4. Van Everdingen, A.F., and Hurst, W. The application of the Laplace transformation to flow problems in reservoirs. Trans. AIME 186, (1949), 305-324.
5. O'Brien, G.G., Hyman, M.A., and Kaplan, S. A study of the numerical solution of partial differential equations. J. Math. and Physics 29, (1951), 223.
6. Bruce, G.H., Peaceman, D.W., Rachford, H.H., Jr., and Rice, J.D. Calculation of unsteady-state gas flow through porous media. Trans. AIME 198, (1953), 79-92.
7. Douglas, Jim, Jr. On the numerical integration of $u_{xx} + u_{yy} = u_t$ by implicit methods. J. SIAM 3, (1955), 42-65.
8. Peaceman, D.W., and Rachford, H.H., Jr. The numerical solution of parabolic and elliptic differential equations. J. SIAM 3, (1955), 28-41.
9. Douglas, Jim, Jr., and Peaceman, D.W. Numerical solution of two-dimensional heat flow problems. A.I.Ch.E. Jour. 1, (1955), 505-512.
10. Douglas, Jim, Jr., Peaceman, D.W., and Rachford, H.H., Jr. A method for calculating multi-dimensional immiscible displacement. Trans. AIME 216, (1959), 297-308.
11. Letkeman, J.P., and Ridings, R.L. A numerical coning model. Soc. Petr. Eng. Jour. 10, (1970), 418-424; Trans. AIME 249, (1970), 418-424.
12. MacDonald, R.C., and Coats, K.H. Methods for the numerical solution of water and gas coning. Soc. Petr. Eng. Jour. 10, (1970), 425-436; Trans. AIME 249, (1970), 425-436.
13. Nolen, J.S., and Berry, D.W. Tests of the stability and time-step sensitivity of semi-implicit reservoir simulation techniques. Soc. Petr. Eng. Jour. 12, (1972), 253-266; Trans. AIME 253, (1972), 253-266.
14. Stone, H.L. Iterative solution of implicit approximations of multidimensional partial differential equations. SIAM J. Numer. Anal. 5, (1968), 530-558.
15. Peaceman, D.W. A nonlinear stability analysis for difference equations using semi-implicit mobility. Soc. Petr. Eng. Jour. 17, (1977), 79-91; Trans. AIME 263, (1977), 79-91.
16. Peaceman, D.W. Interpretation of well-block pressures in numerical reservoir simulation. Soc. Petr. Eng. Jour. 18, (1978), 183-194; Trans. AIME 265, (1978), 183-194.
17. Peaceman, D.W. Interpretation of well-block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability. Soc. Petr. Eng. Jour. 23, (1983), 531-543; Trans. AIME 275, (1983), 531-543.

PARTICLES IN THEIR SELF-CONSISTENT FIELDS: FROM HARTREE' DIFFERENTIAL ANALYZER TO CRAY MACHINES

by Oscar Buneman
Stanford University

After the early success of astronomers in solving rigorously the problem of two gravitationally interacting bodies it became quite a disappointment that the notorious "probleme de trois corps" could never be solved by elegant nineteenth century mathematics. Computations were practical (and respectable) only for the evaluation of series. Finite difference calculus made its way very slowly during the first few decades of this century. Stormer struggled hard to calculate charged particle orbits in the Earth's magnetic field (not even a self-consistent field!) - for which he earned pity, if not ridicule.

Strangely, it was a change in physics which brought the next advance: quantum theory changed particle dynamics from ordinary differential equations to partial differential equations, thus putting field and particle dynamics on the same footing. The combination of Schroedinger's equation for electron density with Poisson's equation for the electric potential results in coupled non-linear PDE's. As a first step, taken in the 1920's, one eliminated the angle variables and reduced the problem to two non-linearly coupled ODE's in the radial variable.

This meant that an efficient integrating machine or procedure was called for and HARTREE built his "differential analyzer" - the first model out of Meccano (American: "erector set"). It uses the principle of a continuously variable gear and its principal element is shown in figure 1. One rotating disc is rolling in contact with another. We note that at constant engine speed one's distance travelled would be the time-integral of the continuously varying gear ratio. The power for this delicate transmission device was provided by a "torque amplifier" which slipped whenever the drive became slower than the load and tightened otherwise.

With initially only four such integrators Hartree solved the problem of self-consistent electronic wave functions and atomic energy levels. Later Metropolitan Vickers built him a well-engineered model with eight integrating tables. Figure 2 shows Hartree bending over that machine in the basement of the Physics building at Manchester University. With him is an assistant whose various roles I shall have occasion to describe later. The little meccano model was sitting by the side of the M-V machine.

Solving quantum mechanical problems as an exercise in coupled PDE's has since become a subject of chemistry with, of course, great strides being made through the availability of more powerful digital computers. However, Hartree's own next

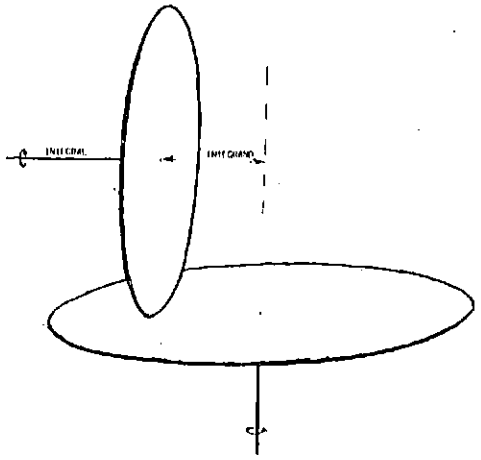


Figure 1. INTEGRATING TABLE IN DIFFERENTIAL ANALYZER

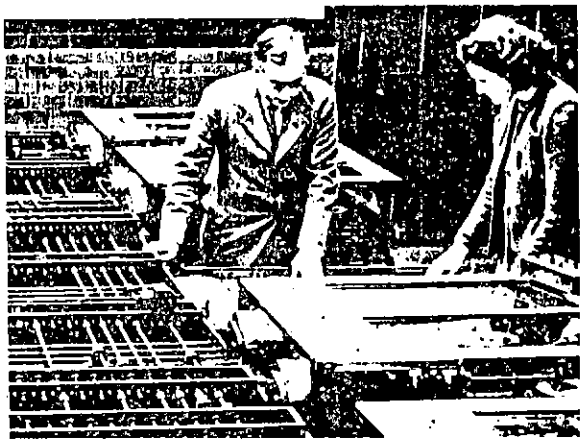


Figure 2. Douglas Hartree with Phyllis Locket leaning over the output plotting table of the differential analyser at Manchester University.

(From M. Wilkes' "Memoirs of a Computer Pioneer", MIT Press 1985)

important contribution to self-consistent field computation came during WW2 and was in classical (meaning non-quantum) dynamics.

The "magnetron", a now very familiar microwave generator, had been invented by Boot,Randall and Self in Birmingham. It was of paramount importance to Britain's defence: the Germans could not jam the magnetron frequencies used for early detection of Luftwaffe take-offs. The magnetron is a fine example of "swords into plough-shares". It is replacing man's tradition of many millennia to cook food with incandescent heat.

Initially it was something of a mystery exactly how and why the magnetron worked and the scientific staff at the British Admiralty realized that in order to unravel the workings of the magnetron one would have to solve a self-consistent field problem, namely that of motion of electrons in the electric field which the electrons themselves produce, in addition to the externally applied electric and magnetic fields.

The Admiralty therefore approached Hartree who promptly initiated classical particle simulation by integrating, numerically, the orbits of large numbers of particles in a field which was either revised in accordance with the instantaneous charge density at each step, or only occasionally, in the hope of reaching a steady field by iteration.

Both one- and two-dimensional simulations were performed by Hartree and the team which he collected for the purpose. The hardware consisted of three Marchant mechanical add-and-shift machines, rather like the mechanical cash registers which have just disappeared. There were three CPU-s: Phyllis Locket who is shown in the picture with Hartree, David Copley, a schoolmaster from Sheffield, and myself. We were about a billion times slower than modern CPU's, but Phyllis was the fastest. Hartree addressed the multi-tasking or parallel CPU problem by sharing out the several hundred orbits between the three of us, at least for the case where the field was only revised occasionally.

He also provided an elegant solution to the difficulty that with time-centered differencing the Lorentz equation of motion in a magnetic field becomes implicit: his algorithm is, to this day, used in particle simulations. The instructions for the three CPU's had been set out by Hartree in the form of a program, with "go-to"-s and loops. We did not called it "looping", though: Phyllis named it "knitting".

The idea of space-time centering in finite difference work was very important to Hartree. When, at another stage in WW2, the Sheffield steel firms wanted to know how long they should cook their ingots he got Phyllis and another assistant to undertake the numerical integration of the heat equation which also becomes implicit under time-centering. Phyllis, by this time, had become Mrs. Nicolson and the other assistant's name was Crank: that was the origin of another famous algorithm!

In the two-dimensional simulations it turned out very beneficial for me to be a human CPU. Unlike electronic CPU-s, which will grind out billions of trivial zeros without objecting or giving us a warning, I observed that my particles shunned certain regions in the field. This made me discover that there exists a new kind of potential, in a rotating frame of reference. It led to the "threshold" criterion for magnetron operation - now an important design tool.

The one-dimensional simulations had yielded a steady state that was approached in transience, but this could not account for magnetron operation, or for the observed currents which flow across the magnetic barrier. I found that this state was two-dimensionally unstable, in a mode similar to the Kelvin-Helmholtz instability. A (linear) differential equation had to be solved to get the growth rates: we programmed that into the differential analyzer. The importance of going into at least two dimensions when there are magnetic fields has dominated charged particle simulation ever since.

We had a major problem over solving Poisson's equation in two dimensions. Hartree introduced us to Southwell's relaxation technique and provided us with hardware in the form of large plastic sheets on which we could record the two-dimensional potential array and on which we could easily rub out to improve our guesses.

We found this far from relaxing and in fact very frustrating trying to chase residuals away to the boundaries. Iterative methods were abandoned at that point (this was in 1944!) and Hartree changed to the direct Fourier method. It turned out that a very modest number of harmonics was adequate: the FFT was not yet known.

Eventually plausible particle-field configurations emerged showing the four- or six-spoke wheel which rotates in the magnetron and which excites the high frequencies in the resonators.

During the late forties and early fifties a small community of electron device engineers maintained self-consistent charged particle simulations while many of us drifted into other areas such as nuclear and fundamental particle physics. However, the quest for fusion brought new impetus to the subject: simulation of plasma electrons and ions in their self-consistent field, and the physics of magnetic barriers.

In the late fifties Dawson at Princeton and I at Stanford began numerical plasma particle simulations. I drew attention to another instability, namely the electron-ion interstreaming instability in high-current plasmas. The non-linear evolution of this had to be calculated by a one-dimensional simulation and the publication of two pages of graphic computer output in Physical Review, showing electron and ion space-time orbits, made quite a stir. It showed how the plasma randomizes directed energy (in the absence of close collisions) and how "anomalous resistivity" comes about. That simulation had been done at Lockheed on an early electronic digital computer, an 1103 AF. There were 256 electrons and 256 ions.

These early simulations were one-dimensional, with no transverse magnetic field, and in view of the importance of magnetic barriers in fusion, plus what one sees in magnetrons, two-dimensional simulations were needed urgently. Fortunately at that point a research student appeared at Stanford who wanted to do plasma physics as well as numerical analysis. He was Roger Hockney and he fitted neatly between Gene Golub and myself as his supervisors.

Hockney embarked on the first serious two-dimensional particle simulation of magnetized plasma. He wrote the program (in Fortran) to advance several thousand particles in the magnetic field by the Hartree algorithm. (Co-incidentally, Hockney had grown up in a house opposite the Hartree's in Manchester, but by this time Hartree was no longer with us.)

When it came to field solving, I drew Hockney's attention to a centerfold in an old text, "Calculus of Observations", by Whittaker and Robinson. It gives a program for the efficient execution of a 24-point discrete Fourier transform. (24 because of the hours in the day and because it includes the numerically convenient angles of 30 or 60 degrees.) The FFT was still unknown to us - or rather, no one had unearthed Gauss' original FFT program, written in Latin.

24-point transforming seemed a bit skimpy for our simulation. The several thousand particles deserved somewhat finer field resolution (particle and field data should balance). This is where Gene Golub stepped in and inspired Hockney with recursive doubling: the first simulation was done on a 48-point grid in the angle direction. The fast direct field solver allowed Hockney to update the field after every particle step. The results, displayed by a movie, showed how just like the electron cloud in the magnetron the plasma develops spokes which allow it to penetrate and conduct across the magnetic field, in what is known as "anomalous diffusion".

Hockney left Stanford for IBM where he started galactic simulations. Many of his ideas and computer practices are documented in his text "Computer Simulation using Particles"

(jointly with J. Eastwood). Having to pick up the threads of his work made me learn Fortran, and to go more deeply into recursive doubling in application to Poisson solving. I found that Gene's principle of recursive doubling could be extended and used for both rows and columns and that one could arrange things so that errors would not build up. I left a few copies of my program (plus sketchy report) at a conference at Los Alamos. They were picked up and later R. Morse told me gleefully: "we've cracked your code". C. Nielssen and B. Buzbee had studied the algorithm and things got back to Gene Golub. The three wrote a profound paper about it.

Particle simulation has since taken a step forward with every advance in computer technology. "Bigger" machines allowed more particles and better resolution. The Illiac, for instance, became popular as a tool for galactic simulations (R.Miller): a large number of stars could be time-stepped in parallel. On the other hand, a code like the Hockney code can now be run on a PC.

Having experienced very early how the inclusion of another dimension can reveal important new physics, there was strong motivation to go from two to three dimensions. This became possible with the advent of the CRAY which combines speed with the benefits of parallelism and pipelining. We have now a TRI-dimensional STANford code, TRISTAN, which traces some five million particles through a field recorded over 160 X 160 X 160 data points. It is fully electromagnetic and relativistic. A time step takes about two minutes. When writing this code, the deep problems did not arise from the physics or numerical analysis, but from the data management. The architecture of the machine heavily affected the choice of simulation methods.

In the code the fields are advanced by Fourier transforming in all three dimensions. At one point there is a distinct bottleneck due to the fact that Fourier methods are global, not local. We have encountered that bottleneck also in simulations on other highly parallel machines of recent design. Looking into the future, one sees an increasing demand for local algorithms, because data path lengths must be minimized. We may want to return to the old "local" method of solving for the field which we discarded in 1944 when we got tired of rubbing out on the plastic sheets. Luckily, it turns out that we can discard the eraser as well, that in a fully electromagnetic simulation each of the local updates is physically significant!