An Interview with M. J. D. Powell

Bulletin of the International Center for Mathematics, n. 14, June 2003.

I am sure that our readers would like to know a bit about your academic education and professional career first. Why did you choose to go to the Atomic Energy Establishment (Harwell) right after college in 1959?

When I studied mathematics at school, nearly all of my efforts were applied to solving problems in text books, instead of reading the texts. Then my teachers marked and discussed my solutions instead of instructing me in a formal way. I enjoyed this kind of work greatly, especially when I was able to find answers to difficult questions myself. Thus I gained a good understanding of some fields of mathematics, but I became unwilling to learn about new subjects at a general introductory level, because I do not have a good memory, and to me it was without fun. I also disliked the breadth of the range of courses that one had to take at Cambridge University as an undergraduate in mathematics. Fortunately, I was able to complete that work adequately in two years, which allowed me to study for the Diploma in Numerical Analysis and Computation during my third year. It was a relief to be able to solve problems again most of the time, and the availability of the Edsac 2 computer was a bonus. I welcomed the use of analysis and the satisfaction of obtaining answers. I wished to continue this kind of work after graduating, but the possibility of remaining in Cambridge for a higher degree was not suggested to me. Contributing to academic research and publishing papers in journals were not suggested either, although I developed a successful algorithm for adaptive quadrature in a third year project. Therefore in 1959 I applied for three jobs at government research establishments, where I would assist scientists with numerical computer calculations. I liked the location of Harwell and the people who interviewed me there, so it was easy for me to accept their offer of employment.

You obtained your doctor of science only in 1979, twenty years after your bachelors degree and three years after being hired as a professor in Cambridge. Why was that the case?

After graduating from Cambridge in 1959 with a BA degree, I had no intention of obtaining a doctorate. All honours graduates from Cambridge are eligible for an MA degree after about 3 further years, without taking any more courses or examinations, but from my point of view that opportunity was not advantageous, partly because one had to pay a fee. When I became the Professor of Applied Numerical Analysis at Cambridge in 1976, I was granted all the privileges of an MA automatically, and my official degree became BA with MA status. Two years later, I was fortunate to be elected as a Professorial Fellow at Pembroke College, and the Master of Pembroke suggested that I should follow the procedure for becoming a Master of Arts. Rather than expressing my reservations about it, I offered to seek an ScD degree instead, which required an examination of much of my published work. Thus I became an academic doctor in 1979.

Was it hard to adapt to the academic life after so many years in Harwell?

After about five years at Harwell, most of my time was spent on research, which included the development of Fortran software for general computer calculations, the theoretical analysis of algorithms, and of course the publication of papers. The purpose of the administrative staff there was to make it easier for scientists to carry out their work. On the other hand, I found at Cambridge that one had to create one's own opportunities for research, which required some stubbornness and lack of cooperation, because of the demands of teaching, examining and admitting students, and also because administrative duties at universities can consume the time that remains, especially during terms. This change was particularly unwelcome, and is very different from the view that most of my relatives and friends have of university life. Indeed, when I was at Harwell they did not doubt that I had a full time job, but they assume that at Cambridge the vacations provide a life of leisure.

In your work in optimization we find several interesting and meaningful examples and counter-examples. Where did you get this training (assuming that not all is natural talent)? From your exposure to approximation theory? From the hand calculations of the old computing times?

The construction of examples and counter-examples is a natural part of my strong interest in problem solving, and of the development of software that I have mentioned. Specifically, numerical results during the testing of an algorithm often suggest the convergence and accuracy properties that are achieved, so conjectures arise that may be true or false. Answers to such questions are either proofs or counter-examples, and often I have tried to discover which of these alternatives applies. Perhaps my training started with my enjoyment of geometry at school, but then the solutions were available. I am pleased that you mention hand calculations, because I still find occasionally that they are very useful.

Was exemplification a relevant tool for you when you taught numerical analysis classes? Did your years as a staff member at Harwell influence your teaching?

My main aim when teaching numerical analysis to students at Cambridge was to try to convey some of the delightful theory that exists in the subject, especially in the approximation of functions. Only 36 lectures are available for numerical analysis during the three undergraduate years, however, except that there are also courses on computer projects in the second and third years, where attention is given to the use of software packages and to the numerical results that they provide. Moreover, in most years I also presented a graduate course of 24 lectures, in order to attract research students. The main contribution to my teaching from my years at Harwell was that I became familiar with much of the relevant theory there, because it was developed after I graduated in 1959, but I hardly ever mentioned numerical examples in my lectures, because of the existence of the Cambridge computer projects, and because the mathematical analysis was more important to my teaching objectives. Therefore my classes were small. Fortunately, some of the strongest mathematicians who attended them became my research students. I am delighted by their achievements.

Could you tell us how computing resources evolved at Harwell in the sixties and seventies and how that impacted on the numerical calculations of those times?

Beginning in 1958, I have always found that the speed of computers and the amount of storage are excellent, because of the huge advances that occur about every three years. On the other hand, the turnaround time for the running of computer programs did not improve steadily while I was at Harwell. Indeed, for about four years after I started to use Fortran in 1962, those programs were run on the IBM Stretch computer at Aldermaston, the punched cards being transported by car. Therefore one could run each numerical calculation only once or twice in 24 hours. Of course it was annoying to have to wait so long to be told that one had written DIMESNION instead of DIMENSION, but ever since I have been grateful for the careful attention to detail that one had to learn in that environment. Moreover, it was easier then to develop new algorithms that extend the range of calculations that can be solved. Conveying such advances to Harwell scientists was not straightforward, however, mainly because they wrote their own computer programs, using techniques that were familiar to them. The Harwell Subroutine Library, which I started, was intended to help them, and to reduce duplication in Fortran software. Often it was highly successful, but many computer users, both then and now, prefer not to learn new tricks, because they are satisfied by the huge gains they receive from increases in the power of computers.

You once wrote: "Usually I produced a Fortran program for the Harwell subroutine library whenever I proposed a new algorithm,..."[1]*. In fact, writing numerical software has always been a concern of yours. Could you have been the same numerical analyst without your numerical experience?*

My principal duty at Harwell was to produce Fortran programs that were useful for general calculations, which justified my salary. My work on the theoretical side of numerical analysis was also encouraged greatly, and its purpose was always to advance the understanding of practical computation. Indeed, without numerical experience, I would be cut off from my main source of ideas. It is unusual for me to make progress in research by studying papers that other people have written. Instead I seek fields that may benefit from a new algorithm that I have in mind. I also try to explain and to take advantage of the information that is provided by both good and bad features of numerical results.

Roger Fletcher wrote once that "your style of programming is not what one might call structured". Some people think that a piece of software should be well structured and documented. Others that it should be primarily efficient and reliable. What are your views on this?

I never study the details of software that is written by other people, and I do not expect them to look at my computer programs. My writing of software always depends on the

[1] *A View of Nonlinear Optimization*, History of Mathematical Programming: A Collection of Personal Reminiscences (J.K. Lenstra, A.H.G. Rinnooy Kan, and A. Schrijver eds), North-Holland (Amsterdam), 119-125 (1991).

discipline of subroutines in Fortran, where the lines of code inside a subroutine can be treated as a black box, provided that the function of each subroutine is specified clearly. Finding bugs in programs becomes very painful, however, if there are any doubts about the correctness of the routines that are used. Therefore I believe that the reliability and accuracy of individual subroutines is of prime importance. If one fulfils this aim, then in my opinion there is no need for programs to be structured in a formal way, and conventional structures are disadvantageous if they do not suit the style of the programmer who must avoid mistakes. Those people who write reliable software usually achieve good efficiency too. Of course it is necessary for the documentation to state what the programs can do, but otherwise I do not favour the inclusion of lots of internal comments.

And by the way, how do you regard the recent advances in software packages for nonlinear optimization?

Most of my knowledge of recent advances in software packages has been gained from talks at conferences. I am a strong supporter of such activities, as they make advances in numerical analysis available for applications. My enthusiasm diminishes, however, when a speaker claims that his or her software has solved successfully about 90% of the test problems that have been tried, because I could not tolerate a failure rate of 10%. Another reservation, which applies to my programs too, is that many computer users prefer software that has not been developed by numerical analysts. I have in mind the popularity of simulated annealing and genetic algorithms for optimization calculations, although they are very extravagant in their use of function evaluations.

Many people working in numerical mathematics undervalue the paramount importance of numerical linear algebra (matrix calculations). Would you like to comment on this issue? How often was research in numerical linear algebra essential to your work in approximation and optimization?

An optimization algorithm is no good if its matrix calculations do not provide enough accuracy, but, whenever I try to invent a new method, I assume initially that the computer arithmetic is exact. This point of view is reasonable for the minimization of general smooth functions, because techniques that prevent serious damage from nonlinear and nonquadratic terms in exact arithmetic can usually cope with the effects of computer rounding errors, as in both cases one has to restrict the effects of perturbations. Therefore I expect my algorithms to include stability properties that allow the details of the matrix operations to be addressed after the principal features of the algorithm have been chosen. Further, I prefer to find ways of performing the matrix calculations myself, instead of studying relevant research by other people.

I read in one of your articles that "a referee suggested rejection because he did not like the bracket notation". What is your view about the importance of refereeing? How do you classify yourself as a referee?...

The story about the bracket notation is remarkable, because the paper that was nearly rejected is the one by Roger Fletcher and myself on the Davidon–Fletcher–Powell (DFP) algorithm. As a referee, I ask whether submitted work makes a substantial contribution to its subject, whether it is correct, and whether the amount of detail is about right. I believe strongly that we can rely on the accuracy of published papers only if someone, different from the author(s), checks every line that is written, and in my opinion that task is the responsibility of referees. When it is done carefully, then refereeing becomes highly important. I try to act in this way myself, but, because my general knowledge of achievements in my fields is not comprehensive, I often consider submissions in isolation, although I should relate them to published work.

Actually, in my previous question I had in mind the difficulty that others might face to meet your high standards. This brings me to your activity as a Ph.D adviser. What difficulties and what rewards do you encounter when advising Ph.D. students?

Of course I take the view that my requirements for the quality of the work of my PhD students are reasonable. I require their mathematics to be correct, I require relevance to numerical computation, and I require some careful investigations of new ideas, instead of a review of a subject with some superficial originality. Further, I prefer my students to work on topics that are not receiving much attention from other researchers, in order that they can become leading experts in their fields. Some of them have succeeded in this way, which is a great reward, but two of them switched to less demanding supervisors, and another one switched to a well paid job instead of completing his studies. I also had a student that I never saw after his first four terms. Eventually he submitted a miserable thesis, that was revised after his first oral examination, and then the new version was passed by the examiners, but the outcome would have been different if university regulations had allowed me to influence the result. On the other hand, all my other students have done excellent work and have thoroughly deserved their PhDs. One difficulty has occurred in several cases, namely that, because each student has to gain experience and to make advances independently, one may have to allow his or her rate of progress to be much slower than one could achieve oneself. Another difficulty is that my knowledge of pure mathematics has been inadequate for easy communication between myself and most of my students who have studied approximation theory. Usually they were very tolerant about my ignorance of distributions and properties of Fourier transforms, for example, but my heart sinks when I am asked to referee papers that depend on these subjects.

Most of your publications are single-authored. Why do you prefer to work on your own?

I believe I have explained already why I enjoy working on my own. Therefore, when I begin some new research, I do not seek a co-author. Moreover, as indicated in the last paragraph, I prefer my students to make their own discoveries, so usually I am not a co-author of their papers.

I have been trying to avoid technical questions but there is one I would like to ask. What *is your view on interior-point methods (a topic where you made only a couple — but as always relevant and significant — contributions)?*

My view of interior point methods for optimization calculations with linear constraints is that it seems silly to introduce nonlinearities and iterative procedures for following central paths, because these complications are not present in the original problem. On the other hand, when the number of constraints is huge, then algorithms that treat constraints individually are also unattractive, especially if the attention to detail causes the number of iterations to be about the number of constraints. It is possible, however, to retain linear constraints explicitly, and to take advantage of the situation where the boundary of the feasible region has so many linear facets that it seems to be smooth. This is done by the TOLMIN software that I developed in 1989, for example, but the number of variables is restricted to a few hundred, because quadratic models with full second derivative matrices are employed. Therefore eventually I expect interior point methods to be best only if the number of variables is large. Another reservation about this field is that it seems to be taking far more than its share of research activity.

You published a book in approximation theory. Have you ever thought about writing a book in nonlinear optimization?

My book on Approximation Theory and Methods was published in 1981. Two years later, my son died in an accident, and then I wished to write a book on Nonlinear Optimization that I would dedicate to him. I have not given up this idea, but other duties, especially the preparation of work for conferences and their proceedings, have caused me to postpone the plan. Of course, because of the circumstances, I would try particularly hard to produce a book of high quality.

Let me end this interview with the very same questions I asked T.R. Rockafellar (who, by the way, shared with you the first Dantzig Prize in 1982). Have you ever felt that a result of yours was unfairly neglected? Which? Why? What would you like to prove or see proven that is still open (both in approximation theory and in nonlinear optimization)? What was the most gratifying paper you ever wrote? Why?

I was taught the FFT (Fast Fourier Transform) method by J.C.P. Miller in 1959, and then it made Cooley and Tukey famous a few years later. Moreover, my 1963 paper with Roger Fletcher on the DFP method is mainly a description of work by Davidon in 1959, and it has helped my career greatly. Therefore, by comparison, none of my results has been unfairly neglected. My main theoretical interest at present is trying to establish the orders of convergence that occur at edges, when values of a smooth function are interpolated by the radial basis function method on a regular grid, which is frustrating, because the orders are shown clearly by numerical experiments. In nonlinear optimization, however, most of my attention is being given to the development of algorithms. Since you ask me to mention a gratifying paper, let me pick "A method for nonlinear constraints in minimization problems", because it is regarded as one of the sources of the "augmented Lagrangian method", which is now of fundamental importance in mathematical programming. I have been very fortunate to have played a part in discoveries of this kind.

Interview by Luís Nunes Vicente (University of Coimbra)

M.J.D. Powell completed his undergraduate studies at the University of Cambridge in 1959. From 1959 to 1976 he worked at the Atomic Energy Establishment, Harwell, where he was Head of the Numerical Analysis Group from 1970. He has been John Humphrey Plummer Professor of Applied Numerical Analysis, University of Cambridge since 1976 and a Fellow of Pembroke College, Cambridge since 1978.

He made many seminal contributions in approximation theory, nonlinear optimization, and other topics in numerical analysis. He has written a book in approximation theory and more than one hundred and fifty papers.