A View of Nonlinear Optimization

M. J. D. Powell

I first had to solve a nonlinear optimization calculation in 1961. It was a least squares fitting problem to determine about five parameters of an atomic structure. I employed the Gauss - Newton method and read some relevant papers. In particular the Partan technique attracted my attention because it does not require any derivatives. I published a paper on a development of this technique in the *Computer Journal* in 1962, and an important consequence was that I was invited to speak on this work at Leeds University. I never presented that talk because between accepting the invitation and travelling to Leeds I had the good fortune to obtain a copy of Bill Davidon's report, which is the original description of the 'DFP method' for unconstrained optimization. I tried that method on some examples and the results were stunning. Therefore at Leeds I spoke on Bill's work instead of on my insignificant research on Partan. Roger Fletcher was in the audience, and after my lecture he told me that he had also been investigating Davidon's method. Of course we were both very keen to publish our findings and we decided to write the joint paper that appeared in the *Computer Journal* in 1963, although a referee suggested rejection because he did not like the bra-ket notation.

I was able to minimize some nonquadratic functions of 100 variables by Davidon's method in 1962. I mentioned this fact at a meeting that was held at Imperial College then, but this remark was impromptu, it was made in a discussion that included the view from a senior person that 10-variable problems were usually too difficult and I was nervous. Roger Sargent was there and I am very grateful to him because he was one of the few people in the audience who believed me. Another incident then was persuading a physicist at Harwell to solve an optimization calculation by running my version of Davidon's method on the Mercury computer instead of using a different procedure on the IBM 709. He was delighted with the savings in execution time and I do not know if he was aware that the Mercury was a much slower machine.

Davidon, Fletcher, Powell

I enjoyed the 1960's. There were so many opportunities for improving existing optimization algorithms and at Harwell I was allowed to spend most of my time on research. Having solved many calculations by hand, I was quite used to thinking of techniques both for avoiding unnecessary work and for checking the accuracy of numerical calculations. This training has been no less useful to me than my knowledge of mathematics, partly because most papers that proposed new algorithms then did not include any serious convergence analysis. Indeed for many years the only intrusion of theory on my research was the point of view that, if an algorithm does not perform well when the objective function is quadratic, then it is unlikely that it will be efficient in general use. This rule of thumb and trying to make good use of available information were the main considerations that helped me to develop several successful algorithms for unconstrained calculations. Further, my contributions to theory have all been retrospective. Occasionally theoretical discoveries have persuaded me to abandon ideas that I had thought were sound. I prefer to regard the conclusions of theoretical analysis as a guide to fine tuning after the structure of a new algorithm has been chosen. Indeed, I tend to deprecate techniques that are proposed in order to facilitate proofs of convergence properties.

The main example that comes to mind is the DFP (or BFGS) algorithm for unconstrained calculations. We still do not know if the infimum of the norms of the gradient vectors is always zero if this method is applied with exact line searches in exact arithmetic to a smooth nonconvex objective function that has bounded level sets. One can guarantee this convergence property, however, by modifying each search direction if necessary so that its angle with the current steepest descent direction is strictly less than $\pi/2$. Some software includes this modification, which I think is crude. I believe that our analytical abilities are very slight indeed in comparison with the observed properties of actual calculations. We should enjoy these properties even when we cannot prove that they exist, and those that are known to be true are delightful. For example, the

DFP method does have quadratic termination and it is invariant under linear transformations of the variables, provided that suitable adjustments are made to the starting conditions too, but the crude modification usually ruins these qualities. I would adopt a suitable modification if it were known to be necessary for convergence, but otherwise I prefer our ignorance to be blissful.

I believe, however, that theoretical studies are vital to the academic standing of our subject. They are the essence of teaching. The courses of lectures that I gave occasionally on optimization in the late 1960's were disjointed descriptions of methods, except for some analysis when the objective function is quadratic. Now, however, one can dazzle mathematics students with neat proofs of superlinear convergence that show adroitness in manipulating Frobenius norms of the errors of approximations to second derivative matrices, while the convergence of descent methods under suitable line search conditions demonstrates the relevance of analysis. Further, one can allude to proofs that require much skill in combining these operations. I teach optimization in this way myself, because I am convinced that the subject contains much good mathematics that is helpful both to people who will solve real mathematical programming problems and to those who will engage in research on algorithms, but I feel uncertain about the value of much more research on the theoretical side of our subject.

Indeed, my personal view is that the following three common ingredients of theoretical papers are all despicable. Analyzing the fine detail of pathological cases of optimality conditions without providing any practical motivation when rounding errors are going to make these details irrelevant in real calculations. Making assumptions in order to prove theorems when the assumptions conflict with our knowledge of typical behaviour of the algorithms that are being considered. Including parameters in algorithms in order that known techniques of analysis can be applied when there is a possibility that the parameters would be unnecessary if the theoretical techniques were improved. Therefore, when I have tried to prove theorems I have often failed. Fortunately I have never been short of other research subjects to investigate, so I believe that tackling difficult theoretical questions should be a part-time activity.

I mention optimality conditions in the previous paragraph because talks on this subject often irritate me at conferences. When they are sandwiched between potentially interesting lectures on algorithms in sessions on nonlinear programming I find myself in the audience. Then I try to be intelligent for a while but am usually confounded by jargon, especially when some brilliant mathematics is being exposed. I am so naive that I wish to grasp the ideas of a subject without understanding the language that is normally used to describe them. In my mind, for example, solving a linear programming problem by the simplex method means moving from vertex to vertex of a vertical convex polyhedron until the bottom is reached. Therefore such terms as shadow prices, complementary slackness, tableaux and nonbasic variables leave me cold. Participants seemed to wallow in such nomenclature at the 1964 Mathematical Programming Symposium in London, which stiffened my preference to study nonlinear optimization.

Here I wish to acknowledge the influence that Martin Beale and Philip Wolfe

have had on my career. From about 1965 onwards they have both been very interested in my research although they were pre-eminent in linear programming. Therefore, from my point of view, they have done most to unite the mathematical programming community. Martin's experience of conjugate gradients in a linearly constrained setting and our differences of opinion on the best ways to handle nonlinear constraints were both of great value to my research, while Phil has inspired me in many ways. In particular he bet me one shilling that I would not prove the convergence of the DFP method with exact line searches when the objective function is smooth and uniformly convex and I won.

My first contribution to constrained optimization was the augmented Lagrangian method but originally I did not see the connection with the Lagrangian function. Then it was usual to apply algorithms for unconstrained minimization to the objective function plus the sum of squares of constraint violations multiplied by a penalty parameter. It occurred to me that, if the unconstrained calculation yielded a constraint residual of 0.3, say, when aiming at zero, then perhaps the residual would have been much smaller if one had aimed at -0.3 using the same penalty parameter. I found that adjusting the offset was more efficient than the standard procedure of increasing the penalty parameter. This simple idea is analogous to the augmented Lagrangian method, because the expansion of each squared penalty term includes minus the relevant constraint function times the current offset times twice the penalty parameter, so the offset is proportional to an estimate of a Lagrange multiplier.

Usually I produced a Fortran program for the Harwell subroutine library whenever I proposed a new algorithm, but I did not write any general software that applies the constrained optimization technique that has just been outlined. The reason was that Roger Fletcher and Shirley Lill were developing their exact differentiable penalty function method then. This method can satisfy constraints by a single unconstrained calculation using the DFP or BFGS algorithm, for example. I took the view that it was so much more powerful than my contribution that a Fortran code for the augmented Lagrangian procedure would have been superfluous. In fact both methods have been used extensively during the last 20 years.

Sequential quadratic programming (SQP) methods for constrained optimization emerged during the 1970's, much of the early work being done by Shih-ping Han. I contributed an implementation and some convergence analysis that I presented in 1977 at conferences that were held in Dundee and Madison respectively. It is now well known that these methods can achieve superlinear convergence in nondegenerate cases by using first derivatives of the objective and constraint functions and a positive definite approximation to the second derivative matrix of the Lagrangian function, but becoming aware of this fact was stunning. Previously there were rather slow barrier and penalty function methods, there were augmented Lagrangian algorithms, there was the exact differentiable penalty function whose gradient depends on second derivatives of the data, while the reduced gradient methods required frequent correction procedures to cope with nonlinear constraints. Further, we knew that second derivatives of both the objective function and the constraint functions are im-

portant to efficiency. We discovered, not only that these second derivatives are combined suitably if one employs the Hessian of the Lagrangian function, but also that active constraints are a positive benefit because linear approximations to them provide useful reductions in the freedom in the variables. These developments were very exciting and the subsequent systematic testing of Willi Hock and Klaus Schittkowski showed that a breakthrough had been achieved.

One can decry this enthusiasm by taking the view that an SQP method is merely an augmented Lagrangian algorithm that replaces each unconstrained calculation by a single iteration that uses suitable estimates of Lagrange multipliers. Such mundaneness, however, cannot diminish the stirring memories of having participated in a revolution that transformed my perception of the inherent difficulties of nonlinear constraints. Unfortunately, the memories fade as our gains in understanding are absorbed into standard textbooks and routine lecture courses.

The impact of this revolution on available computer software, however, seems to have been rather slight. In my opinion the most successful contribution to software for nonlinear optimization in the 1970's was the Minos code of Bruce Murtagh and Mike Saunders. By taking advantage of any sparsity in matrices of constraint coefficients, it is able to solve problems of a size that the linear programming community regards as nontrivial, and of course it allows the objective function to be nonlinear. A later version admits some nonlinear constraints by the augmented Lagrangian method, and much useful further work in this field has been done at Stanford by Philip Gill, Walter Murray, Mike Saunders and Margaret Wright. On the other hand, the SQP method is less suitable when the number of variables is large because of the need to solve a quadratic programming problem on every iteration. Numerical experiments show that its main advantage is substantial savings in the number of function and gradient evaluations that are needed to complete a calculation. Unfortunately, when the work of these evaluations is dominant and significantly time consuming on a modern machine, then usually the objective function is very elaborate. Typically each evaluation may include an iterative procedure that can introduce discontinuities that are much larger than rounding errors, and convergence to a local minimum that is not global can be a strong possibility. Therefore computers are now so fast that, if there are clear advantages in using an SQP method, then several other considerations are significant too, which causes general SQP software to be of little value to nonexpert computer users.

Therefore I believe that recent and current research on the SQP method will not provide breakthroughs in the solution of real optimization calculations, unless experts in mathematical programming are involved. The current topics that in my view are particularly important to applications generally are techniques for sparsity and structure that will allow huge numbers of variables, spin-offs from Karmarkar's algorithm, the automatic generation of derivatives when the functions to be differentiated are specified by computer codes, the development and investigation of empirical methods like simulated annealing for large calculations, and procedures for global optimization. I have not mentioned parallel computation explicitly because it can be an integral part of these important

fields of investigation.

None of these subjects receives serious attention in the 24 lecture course on optimization algorithms that I am teaching now to graduate students in mathematics, and I believe that this comment also applies to many optimization courses elsewhere. When I was a student myself I received excellent tuition in classical analysis, I learned that matrices can be regarded as linear operators, and I became adept at constructing numerical approximations from finite difference operators. This exposure to some of the disciplines of mathematics and to numerical methods helped me to think successfully when I began to study and construct techniques for optimization a few years later. Because my current optimization course provides similar basic training, I do not see an urgent need to change it yet. Further, I believe that coherent mathematics is more suitable material for lecturing than consideration of the list of incomplete research topics that is given in the previous paragraph. Unfortunately, however, graduate lecture courses often determine the research subjects of doctoral students, especially when the supervisors do not investigate the new fields. My career was helped by my move to Harwell where I was exposed to practical calculations before I began any serious research.

There are many demanding research subjects of high academic quality within the domains of traditional lecture courses that address variable metric, conjugate gradient, active set, augmented Lagrangian and SQP algorithms for nonlinear optimization calculations that have relatively few variables. The fascinating theoretical questions and introvert needs for new techniques include better ways of preserving the positive definiteness of variable metric matrices when Lagrangian functions have negative curvature, the use of trust regions, the development of new merit functions, approximations to quadratic programming subproblems in order to save work and many convergence properties. Further, we have created many precedents and have set academic standards that can be used to assess such work when it is offered for publication or is submitted as a dissertation for a higher degree. Thus nonlinear optimization can foster a self-contained and respectable academic community that turns its back on the real world. In my opinion this would be nearly as bad as being slaves to whoever funds our research. Fortunately there are excellent opportunities for new work throughout the continuous spectrum that is bounded by these extremes. Some examples have been mentioned. They are all derived from the needs of computer users and their calculations.