Numerical Analysis Group Internal Report 2003-1

Some reflections on the current state of active-set and interior-point methods for constrained optimization[1]

Nicholas I. M. Gould[2,3]

ABSTRACT
We reflect on the current state of active-set and interior-point methods for convex and non-convex constrained optimization. We voice some concerns about current SQP methods.

[1] This article appeared as part of special edition of the SIAM Activity Group on Optimization Views-and-News newsletter edited by Sven Leyffer and Jorge Nocedal and devoted to “Large Scale Non-Convex Optimization” (Vol. 14(1), April, 2003, pp. 2–7).

[2] Computational Science and Engineering Department, Rutherford Appleton Laboratory, Chilton, Oxfordshire, OX11 0QX, England, EU. Email : n.gould@rl.ac.uk

[3] Current reports available from “http://www.numerical.rl.ac.uk/reports/reports.html”.

Computational Science and Engineering Department
Atlas Centre
Rutherford Appleton Laboratory
Oxfordshire OX11 0QX

April 29, 2003.

1 Introduction

This is an exciting time to be working in constrained nonlinear optimization. New ideas abound. Collaborations and alliances are forged, rivalry is intense, competition fierce. Why should this be? After all, surely the importance of optimization was recognised many decades ago. So why, now, should there be so much activity and why did it take so long?

I believe that the answer is complicated, but certainly one of the main reasons is that, finally, we really are starting to believe that we have the right (theoretical and practical) tools to tackle the problems we have long been asked to solve. What was the stimulus for this? Well, without doubt in part what has been called the "interior-point" revolution. But also the fightback from the traditionalists, those who promote earlier "active-set" approaches. And finally, the recognition from practitioners that, yes indeed, we can now solve sizable nonlinear programming problems, so that there has been a shift away from linear models and the thinking that lead to these.

In this short article, I hope to explain the salient points of both approaches, the symbiosis that has arisen, and how both approaches have impacted on nonlinear optimization. But I also want to look to the future, and to see how things may develop.

2 History

2.1 Active-set methods

In the beginning, there was linear programming: as simple an approximation to the real world as one could possibly make, but nonetheless one of the most important (and most studied) problems in the history of computational mathematics. As we all know, linear programming is concerned with (say) minimizing a linear function of n unknown parameters (variables) over a feasible region described by m linear equations and/or inequalities. A solution will (almost always) occur at a vertex of the feasible region, and the archetypical *active-set* solution algorithm, the Simplex method, aims to find such a solution by moving through a sequence of objective-improving, feasible, adjacent vertices. Thus, the search is to determine which of the constraints "define" the solution (the *active* ones), and which may be safely discarded, and this defining characteristic extends easily to more general constrained optimization problems. Such an algorithm may explore an exponential (in terms of $m-n$) number of active "sets" in the worst case, is known to depend linearly on these parameters on the average, and in practice really seems to behave just as its average-case performance predicts. Thus for a problem involving, say, a million degrees of freedom, it is reasonable to expect a few millions iterations. While this might at first sound impractical, it is vital to recognise that for linear constraints the dominant cost per iteration is usually the solution of a system of linear equations, and that each system is a rank-one modification of its predecessor. Thus the cost per iteration is often very small, and it is this feature that has kept the Simplex method for linear programming competitive over the past 50 years. Most commercial linear programming systems (such as CPLEX, Xpress and OSL) still have Simplex components (albeit with numerous enhancements such as advanced crash and pre-solve procedures, steepest edge exchange rules, and hyper-sparsity exploiting linear solvers, etc.), and the active-set paradigm also extends into the nonlinear world by virtue of successful and widely-used packages like MINOS [25] and SNOPT [14]. Our experience is that to build a successful active-set method requires considerable care, since rounding errors have ample opportunities to build up and cause havoc over the large number of iterations that occur, even in the linear and quadratic programming cases.

2.2 Interior-point methods

Knowing that the Simplex method might take an exponential number of steps started a race to find alternatives whose worst-case complexity was polynomial in $m - n$. The first-reported polynomial algorithm, the ellipsoid method, has alas not turned out to be effective in practice [1]. Fortunately the next competitor, Karmarkar's method [21], proved to be a major advance, and stated a frenetic research feeding-frenzy on interior-point methods which has continued to this day. Karmarkar's genius was to produce a *nonlinear* iteration that attempted to stay well away from the boundary of the feasible region (and thus avoid the influence of the myriad of vertices) until it approached optimality. It was soon recognised that the method (and many of its successors) may be interpreted as the approximate minimization of a sequence of logarithmic barrier functions—or, if we prefer, as following the "central path" defined as the trajectory of such minimizers as a function of the barrier parameter—and these perspectives have obvious and important consequences for its use for nonlinear problems.

The current state for linear (and many convex) problems is primal-dual variants (in which, as the name suggest, duality plays a strong role and primal and dual variables are treated equally) whose worst-case behaviour to achieve a close-to optimal solution varies like $O(\sqrt{m-n})$ in theory and significantly better (perhaps $O(\log(m-n))$?) in practice [26, 38]. All of the major commercial systems contain interior-point solvers (again with a large number of enhancements). It is interesting to note that although such methods require considerably fewer iterations than their active-set rivals, the cost per iteration is significantly higher—there is, in general, no rank-one update for the crucial linear systems—so that there is still fierce and healthy competition between the two competing ideologies. Certainly, active-set interior-point hybrids are now popular and successful. It remains to be seen that, if in the long term as problem sizes grow, the superior complexity bounds for interior-point methods proves decisive, but I believe this will be the case.

3 Where are we now?

Thus far, all seems perfect. But how do these ideas extend into the nonlinear, nonconvex world?

3.1 The trouble with SQP ...

Extending active-set methods would at a first glance appear to be easy, simply replacing the solution over the whole feasible set by that over a sequence of active sets in which the inactive inequalities are discarded. However, the resulting subproblems are still nonlinear, and thus in principle will each require an infinite iteration. Early attempts to "truncate" such subproblems suffered from a nasty phenomenon known as zigzagging in which constraints continually entered and left the active set.

A more successful idea is to replace the general problem by a sequence of "simple", tractable approximations. For instance, one might replace the objective and constraints by linear approximations (the so-called Successive Linear Programming or SLP approach [11]) or perhaps the objective by a quadratic approximation (the Successive Quadratic Programming or SQP approach [2, 19]). The advantage here is that the subproblem (a linear or quadratic program) is significantly easier to solve than the nonlinear approximation of the previous paragraph. Indeed if the quadratic approximation is convex (or a linear approximation used), we have polynomial-time subproblem-solution methods at our disposal. Having solved the subproblem, we can use its solution as the next trial iterate, and we might embed such a scheme within a linesearch, a trust-region or a filter globalization scheme. But caution is needed here, since there are a number of potential pitfalls.

Firstly, it is well known that the globalization scheme may interfere catastrophically with the SLP/SQP step (the Maratos effect) and avoiding action may result in extra computation [2, 19].

Secondly, to obtain fast ultimate convergence, it is usually vitally important to use some 2nd derivative information/approximation (and thus ultimately some form of SQP iteration). If we are "lucky" enough to have (and use) exact 2nd derivatives, the resulting nonconvex QP may have a number of local minimizers, some of which may not be consistent with our overall globalization scheme (the SQP step may be "uphill"). Although many active-set QP solvers can ensure that the step is downhill, I do not currently know how to guarantee this for interior-point QP solvers. If we must resort to "approximate" (say secant-approximation) 2nd derivatives, it is known that requirements of sparsity and positive-definiteness together conflict with stability [28], so we may be restricted to dense updates, and thus problems with few degrees of freedom—it is worth noticing that all of the successful SQP and SQP-like methods we are aware of (such as MINOS, SNOPT and filterSQP[10]) rely on having relatively few degrees of freedom.

Thirdly, if there is one lesson we should have learned from large-scale unconstrained minimization, it is to aim to solve the subproblem as *inaccurately* as possible consistent with overall convergence—the truncated Newton approach [8], along with its practical manifestation via the linear (preconditioned) conjugate-gradient method, is one of the key ideas to have evolved in the unconstrained case during the 20th century. So it is clearly desirable to truncate the LP/QP solution process. But how? We are aware of almost no work in this area (but see [24] for an exception), and it is of vital practical importance. Again, it would seem easier to stop "early" with an active-set QP solver than with an interior-point one.

Finally, we would ultimately expect that the active sets for our LP/QP subproblems will settle down as we approach the solution to the overall problem [27]. This suggests that we should be exploiting *a priori* information about candidate active sets to warm start subsequent subproblem solves. This would seem to be one area in which active-set methods have a clear edge, since the ability to warm start interior-point methods is in its infancy—there has been some work in the LP case [15, 40], but to our knowledge none for QPs. In practice, by contrast, we have observed that it is still sometimes faster (especially in the degenerate case) to "cold-start" an interior-point QP than "warm start" active set QP code, simply because even slightly incorrect active set predictions can have dramatic undesirable consequences for active-set methods [20].

We have currently suspended development of the large-scale SQP method that we had intended including in GALAHAD [18] despite having produced both effective active-set and interior-point QP solvers. Our experience has been that without QP truncation, the cost of the QP solution so dominates that other non-SQP approaches (such as IPOPT [33], KNITRO [4] and LOQO [32]), in which truncation is possible, have made significant progress even before our QP code had solved its first subproblem!—see also [23] for further evidence that interior-point methods appear to scale better than SQP ones. We are more enthusiastic about an SLP-QP approach we are currently developing [3], since LP truncation is in principle easier and since the QP phase is restricted to a problem with equality constraints for which a truncated conjugate-gradient iteration is possible.

3.2 Whither interior-point methods . . . ?

As I mentioned above, we produced two (nonconvex) quadratic programming packages for GALAHAD. Considerable numerical experience has indicated to us that the interior-point version, QPB is almost always vastly superior for large problems [7]. Since we have now all but given up our SQP developments, we have now turned to what we consider to be the other possibility, namely to solve general constrained optimization problems by sequential barrier-function minimization, using the lessons learned when designing and evaluating QPB.

We were warned as children that barrier-function methods are beastly because of the effects the barrier has close to the boundary. It later turned out that these fears were almost groundless, and that actually observed inefficiencies were to a large degree due to using the wrong dual variable updates following a reduction in the barrier parameter [36]. Without doubt, the problem does become very poorly conditioned near the solution, but this itself does not cause failure since even search direction calculations that might result in large numerical errors do not, because all such errors are confined to uninteresting subspaces [16, 37, 39]. But being prematurely close to the boundary certainly is bad in that it can be painfully slow to escape. For example, if we wish to minimize $-x$ for $x \in [0,1]$ and start with x_0 very close to zero, the Newton barrier correction (with modest barrier parameter) results in a new point $x_1 \approx 2x_0$. Thus an initial point $x_0 = 2^{-40} \approx 10^{-12}$ will take roughly 40 iterations to move to the centre of the feasible region. The lesson here is, I believe, to stay away from the boundary unless there are good reasons to get close (such as if a particular constraint is active at optimality). I strongly believe that it pays to stay close to "the" central path since this avoids as best as possible premature contact with the boundary, although since different scalings result in different central paths, it is far from obvious which path is actually the one to follow!

An important question if we are to use an interior-point approach is how we should handle equality constraints. To a certain extent, I suggest this depends on quite what sort of constraints they are. If they (or some of them) are linear, I believe that it often pays to use a "phase-1" procedure to find a "well-centred" feasible point for these constraints, and thereafter to ensure that they remain satisfied on all subsequent iterations. The reasoning is simply that dealing with a nonlinear objective is tricky enough without having to cope with non-convexity in subspaces that the constraints rule out—we certainly have seen the advantages even in the "simple" case of nonconvex quadratic programming of using feasible rather than infeasible interior point methods.

Nonlinear equality constraints are altogether more tricky, and it is in this area that the most significant differences between competing modern interior point methods occur. Some methods (like IPOPT, KNITRO, LOQO and INTOPT [22]) prefer to treat them explicitly by gradually moving towards feasibility but balancing the requirements of optimality using a penalty function or filter. Others like [29] and the method we are developing for GALAHAD replace equalities by one-sided inequalities (which are then handled using interior-point technology) and other-sided penalization. At this stage I do not think we know which of this approaches is best, but it is likely that actually there is very little difference.

There are two important side issues here, though. Firstly, if we really believe we have good methods for handing equations, is it better to treat inequalities by converting them to equations using slack variables and then simply treat the slack variables using interior-point methods? ¿From a linear-algebraic perspective there is little difference, but there seem to be ardent devotees of both schools of thought [6], so I do not really believe we have exhausted or settled this question. Secondly, if we plan to use equality constraints explicitly, it is vital that there is some coherence between the search direction employed and the merit function used to ensure their ultimate satisfaction. Several cautionary examples [5, 34] attest to the pitfalls that may befall the unwary.

The asymptotic behaviour of interior-point methods is relatively well understood even in the non-convex case, at least under non-degeneracy assumptions: the barrier parameter may be reduced at a superlinear rate so that the overall iteration converges superlinearly for primal-dual methods [17] and 2-step superlinearly for primal-only methods [9], although the latter requires some care when reducing the barrier parameter. Some progress has been made in the degenerate case, but we do not currently have as complete an understanding as in the linear programming case where degeneracy does not hinder convergence to a well-defined point in the centre of the face of optimally active constraints. In practice, asymptotic convergence behaviour appears to behave just as one would hope from the linear programming experience.

So what are the outstanding issues? The effects of constraint scaling, and just how one might re-scale to improve convergence are not well understood. Just as importantly, since as we have hinted we really wish to truncate the calculation of the Newton-barrier search direction, we need to discover how to precondition the conjugate-gradient scheme that we will undoubtedly use; it is already clear that any preconditioner has to reflect the dominant barrier terms in the Hessian matrix of the barrier function, but just how much more is needed is unknown. Finally, another area where there is room for improvement is in extrapolation for better points on the central path. This has proved to be most useful in the linear programming case, but things are certainly more complicated for nonlinear problems because of possible bizarre behaviour [13] of the central path (multiplicity, bifurcation, and even non-existence).

4 Conclusions

I hope I have persuaded the reader that these are indeed exciting times in nonlinear optimization. With interior-point and (to some extent) active set approaches we now have a realistic chance of solving very large nonlinear programming problems. Of course there are difficulties, but the ingenuity and vigour with which the research community is currently addressing such challenges makes me extremely optimistic about future progress. Even in the last few months we have heard of a number of new and interesting proposals [12, 29, 35, 30, 31], and we eagerly await to see how these complement the already large corpus of algorithms and software.

References

[1] R. G. Bland, D. Goldfarb, and M.J. Todd. The ellipsoid method: A survey. *Operations Research*, 29:1039–1091, 1981.

[2] P. T. Boggs and J. W. Tolle. Sequential quadratic programming. *Acta Numerica*, 4:1–51, 1995.

[3] R. H. Byrd, N. I. M. Gould, J. Nocedal, and R. A. Waltz. An active set algorithm for nonlinear programming using linear programming and equality constrained subproblems. Technical Report RAL-TR-2002-032, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 2002.

[4] R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 2000.

[5] R. H. Byrd, M. Marazzi, and J. Nocedal. On the convergence of Newton iterations to non-stationary points. Technical Report 2001/7, Optimization Technology Center, Northwestern University, Evanston, Illinois, USA, 2001. To appear, Math. Programming.

[6] R. H. Byrd, J. Nocedal, and R. A. Waltz. Feasible interior methods using slacks for nonlinear optimization. Technical Report 11, Optimization Technology Center, Argonnne National Laboratory, Argonne, Illinois, USA, 2000. To appear, Comp. Optim. & Apps.

[7] A. R. Conn, N. I. M. Gould, D. Orban, and Ph. L. Toint. A primal-dual trust-region algorithm for non-convex nonlinear programming. *Mathematical Programming*, 87(2):215–249, 2000.

[8] R. S. Dembo and T. Steihaug. Truncated-Newton algorithms for large-scale unconstrained optimization. *Mathematical Programming*, 26(2):190–212, 1983.

[9] J.-P. Dussault. Numerical stability and efficiency of penalty algorithms. *SIAM Journal on Numerical Analysis*, 32(1):296–317, 1995.

[10] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, 91(2):239–269, 2002.

[11] R. Fletcher and E. Sainz de la Maza. Nonlinear programming and nonsmooth optimization by successive linear programming. *Mathematical Programming*, 43(3):235–256, 1989.

[12] E. M. Gertz and P. E. Gill. A primal-dual trust region algorithm for nonlinear optimization. Optimization Technical Report 02-09, Computer Science Department, University of Wisconsin, Madison, USA, 2002.

[13] J. Ch. Gilbert, C.C. Gonzaga, and E. Karas. Examples of ill-behaved central paths in convex optimization. Technical Report 4179, INRIA, Rocquencourt, France, 2001. To appear, Math. Programming.

[14] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12(4):979–1006, 2002.

[15] J. Gondzio and A. Grothey. Re-optimization with the primal-dual interior point method. Technical Report MS-01-004, Department of Mathematics and Statistics, University of Edinburgh, Scotland, 2001. To appear, SIAM J. Optimization.

[16] N. I. M. Gould. On the accurate determination of search directions for simple differentiable penalty functions. *IMA Journal of Numerical Analysis*, 6:357–372, 1986.

[17] N. I. M. Gould, D. Orban, A. Sartenaer, and Ph. L. Toint. Superlinear convergence of primal-dual interior point algorithms for nonlinear programming. *SIAM Journal on Optimization*, 11(4):974–1002, 2001.

[18] N. I. M. Gould, D. Orban, and Ph. L. Toint. GALAHAD—a library of thread-safe fortran 90 packages for large-scale nonlinear optimization. Technical Report RAL-TR-2002-014, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 2002.

[19] N. I. M. Gould and Ph. L. Toint. SQP methods for large-scale nonlinear programming. In M. J. D. Powell and S. Scholtes, editors, *System Modelling and Optimization, Methods, Theory and Applications*, pages 149–178, Dordrecht, The Netherlands, 2000. Kluwer Academic Publishers.

[20] N. I. M. Gould and Ph. L. Toint. Numerical methods for large-scale non-convex

quadratic programming. In A. H. Siddiqi and M. Kočvara, editors, *Trends in Industrial and Applied Mathematics*, pages 149–179, Dordrecht, The Netherlands, 2002. Kluwer Academic Publishers.

[21] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[22] L. S. Lasdon, J. Plummer, and G. Yu. Primal-dual and primal interior point algorithms for general nonlinear programs. *ORSA Journal on Computing*, 7(3):321–332, 1995.

[23] J. L. Morales, J. Nocedal, R. Waltz, G. Liu, and J. P. Goux. Assessing the potential of interior methods for nonlinear optimization. In O. Ghattas, editor, *Proceedings of the First Sandia Workshop on Large-Scale PDE-Constrained Optimization*, Heidelberg, Berlin, New York, 2002. Springer Verlag.

[24] W. Murray and F. J. Prieto. A sequential quadratic programming algorithm using an incomplete solution of the subproblem. *SIAM Journal on Optimization*, 5(3):590–640, 1995.

[25] B. A. Murtagh and M. A. Saunders. A projected Lagrangian algorithm and its implementation for sparse non-linear constraints. *Mathematical Programming Studies*, 16:84–117, 1982.

[26] J. Renegar. *A Mathematical View of Interior-Point Methods in Convex Optimization.* SIAM, Philadelphia, 2001.

[27] S. M. Robinson. Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear programming algorithms. *Mathematical Programming*, 7(1):1–16, 1974.

[28] D. C. Sorensen. An example concerning quasi-Newton estimates of a sparse Hessian. *SIGNUM Newsletter*, 16(2):8–10, 1981.

[29] A. L. Tits, A. Wächter, S. Bakhtiari, T. J. Urban, and C. T. Lawrence. A primal-dual interior-point method for nonlinear programming with strong global and local convergence properties. Technical Report TR 2002-29, Institute for Systems Research, University of Maryland, College Park, MD 20742, USA, 2002.

[30] M. Ulbrich and S. Ulbrich. Non-monotone trust region methods for nonlinear equality constrained optimization without a penalty function. *Mathematical Programming*, Published Online, 2002.

[31] M. Ulbrich, S. Ulbrich, and L. Vicente. A globally convergent primal-dual interior point filter method for nonconvex nonlinear programming. Technical Report TR00-11, Department of Mathematics, University of Coimbra, Coimbra, Portugal, 2000.

[32] R. J. Vanderbei and D. F. Shanno. An interior point algorithm for nonconvex nonlinear programming. *Computational Optimization and Applications*, 13:231–252, 1999.

[33] A. Wächter. *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering.* PhD thesis, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, USA, 2002.

[34] A. Wächter and L. T. Biegler. Failure of global convergence for a class of interior point methods for nonlinear programming. *Mathematical Programming*, 88(3):565–574, 2000.

[35] A. Wächter and L. T. Biegler. Global and local convergence of line search filter methods for nonlinear programming. CAPD Technical Report B-01-09, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, Philadelphia, USA, 2001.

[36] M. H. Wright. Why a pure primal Newton barrier step may be infeasible. *SIAM Journal on Optimization*, 5(1):1–12, 1995.

[37] M. H. Wright. Ill-conditioning and computational error in interior methods for nonlinear programming. *SIAM Journal on Optimization*, 9(1):84–111, 1999.

[38] S. J. Wright. *Primal-Dual Interior-Point Methods.* SIAM, Philadelphia, USA, 1997.

[39] S. J. Wright. Effects of finite-precision arithmetic on interior-point methods for nonlinear programming. *SIAM Journal on Optimization*, 12(1):36–78, 2001.

[40] E. A. Yildirim and S. J. Wright. Warm-start strategies in interior-point methods for linear programming. *SIAM Journal on Optimization*, 12(3):782–810, 2002.