Acta Numerica (2005), pp. 299–361
DOI: 10.1017/S0962492904000248
© Cambridge University Press, 2005
Printed in the United Kingdom

Numerical methods for large-scale nonlinear optimization

Nick Gould
Computational Science and Engineering Department,
Rutherford Appleton Laboratory,
Chilton, Oxfordshire, England
E-mail: n.i.m.gould@rl.ac.uk

Dominique Orban
Department of Mathematics and Industrial Engineering,
Ecole Polytechnique de Montréal,
2900, Bd E. Montpetit, H3T 1J4 Montréal, Canada
E-mail: dominique.orban@polymtl.ca

Philippe Toint
Department of Mathematics,
University of Namur,
61, rue de Bruxelles, B-5000 Namur, Belgium
E-mail: philippe.toint@fundp.ac.be

Recent developments in numerical methods for solving large differentiable nonlinear optimization problems are reviewed. State-of-the-art algorithms for solving unconstrained, bound-constrained, linearly constrained and non-linearly constrained problems are discussed. As well as important conceptual advances and theoretical aspects, emphasis is also placed on more practical issues, such as software availability.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

CONTENTS

1 Introduction 300
2 Large-scale unconstrained optimization 301
3 Large-scale bound-constrained optimization 311
4 Large-scale linearly constrained optimization 317
5 Large-scale nonlinearly constrained optimization 329
6 Conclusion 347
References 347

1. Introduction

Large-scale nonlinear optimization is concerned with the numerical solution of continuous problems expressed in the form

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ f(x)\ \text{ subject to }\ c_{\mathcal{E}}(x)=0\ \text{ and }\ c_{\mathcal{I}}(x)\geq 0, \tag{1.1}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$, $c_{\mathcal{E}}\colon \mathbb{R}^n \to \mathbb{R}^{n_{\mathcal{E}}}$ and $c_{\mathcal{I}}\colon \mathbb{R}^n \to \mathbb{R}^{n_{\mathcal{I}}}$ are smooth and n, and possibly $n_{\mathcal{E}}$ and/or $n_{\mathcal{I}}$, are large. Here, the components of the vector x are the *variables*, $f(x)$ is the *objective function* and the components of the vectors $c_{\mathcal{E}}(x)$ and $c_{\mathcal{I}}(x)$ are the *constraint* functions. Such problems arise throughout science, engineering, planning and economics. Fortunately algorithmic development and theoretical understanding generally continue to keep apace with the needs of such applications.

Our purpose in this paper is to review recent developments, with an emphasis on discussing state-of-the-art methods for various problem types fitting within the broad definition (1.1). As the title indicates, we will focus on nonlinear problems, that is, on problems for which at least one of the functions involved is nonlinear – although many of the methods for linear programming are variants of those in the nonlinear case, extra efficiencies are generally possible in this first case, and the general state of the art is to be able to solve linear problems perhaps ten times larger than nonlinear ones (Bixby, Fenlon, Gu, Rothberg and Wunderling 2000). We shall also mostly be concerned with large problems, that is, at the time of writing, those involving of the order of 100,000 variables and perhaps a similar number of constraints. However, we accept that this estimate may be too conservative for some problem classes – for instance, larger quadratic programs can certainly be solved today. Moreover, structure plays an important role in the size of problems that can be tackled: large sparse or partially separable cases are easier to handle than dense ones. Finally, the definition of a large problem may also depend on the hardware used, although this effect is less visible than in the past because of the remarkable evolution of personal computers in terms of memory processing power.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

We will not review the history of the field here, but refer the interested reader to Gould and Toint (2004*a*) for a brief perspective and a discussion of the reasons why this mature research domain remains so active and why this is likely to continue for some time. The field has acquired a vast literature, and there have been numerous attempts to synthesize various aspects of it in books, such as those by Bertsekas (1995), Bonnans, Gilbert, Lemaréchal and Sagastizábal (1997), Dennis and Schnabel (1983), Fletcher (1981), Gill, Murray and Wright (1981), Moré and Wright (1993), Nash and Sofer (1990), Nocedal and Wright (1999), Conn, Gould and Toint (2000*a*), in volumes of conference proceedings, such as those edited by Coleman and Li (1990), Leone, Murli, Pardalos and Toraldo (1998), Di Pillo and Gianessi (1996, 1999), Di Pillo and Murli (2003), Hager, Hearn and Pardalos (1994), Spedicato (1994), Yuan (1998), in survey articles, like those given by Conn, Gould and Toint (1994, 1996), Fletcher (1987*b*), Forsgren, Gill and Wright (2002), Gould (2003), Marazzi and Nocedal (2001), Nash (2000*b*) and, in this series, by Boggs and Tolle (1995), Lewis and Overton (1996), Nocedal (1992), Powell (1998), Todd (2001), and Wright (1992).

The paper is structured as follows. Sections of the paper deal with problem classes: Section 2 covers unconstrained problems, while bound-constrained and linearly constrained problems are reviewed in Sections 3 and 4, respectively, and Section 5 considers general nonlinearly constrained cases. In each of these sections, subsections refer to method classes, allowing the interested reader to focus on these across different problem types. In particular, we discuss linesearch and trust region methods successively. We conclude most sections with a paragraph on practicalities and a paragraph on software. Final comments are made in Section 6.

2. Large-scale unconstrained optimization

2.1. General problems

Although general unconstrained optimization problems (that is, problems where $\mathcal{E}$ and $\mathcal{I}$ are empty in (1.1)) arise relatively infrequently in practice – nonlinear least-squares problems (see Section 2.2) being a notable exception – a brief discussion of methods for unconstrained optimization is useful if only for understanding those for problems involving constraints. For a fuller discussion see Nocedal (1992, 1997). While hybrids are possible, the essential distinction over the past 35 years has been between the linesearch and trust region approaches.

Given an estimate x_k of an unconstrained minimizer of $f(x)$, both paradigms rely on simple (differentiable) models $m_k(d)$ of the objective function $f(x_k + d)$. For *linesearch* methods m_k will normally be convex while this is not required in the trust region case; for both it is usually important that $m_k(0) = f(x_k)$ and $\nabla_x m_k(0) = \nabla_x f(x_k)$. Given a suitable model, a model-

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

improving approximate minimizer d_k is computed. In the trust region case, possible unboundedness of the model is naturally handled by the trust region constraint $\|d\| \leq \Delta_k$ for some $\Delta_k > 0$. Since the model is only a local representation of the objective function, it is possible that predicted improvements in f may not actually be realized. Linesearch methods account for this by retracting the step along d_k so that $x_k + \alpha_k d_k$ gives an improvement in f. In contrast, *trust region* methods reject steps for which there is poor agreement between the decrease in m_k and f, and rely on a reduction of the radius Δ_{k+1}, and thus a re-computation of d_{k+1}, to ensure improvement. The mechanics of finding the step-size α_k for linesearch methods (Hager and Zhang 2003, Moré and Thuente 1994) and adjusting the radius Δ_k in trust region methods (Conn *et al.* 2000*a*, §17.1) has been much studied, and can have a significant effect on the performance of an algorithm. But overall the dominant computational cost of both classes of algorithms is in evaluating the values and required derivatives of f and in computing the step d_k; the cost of evaluating f often dominates in simulation-based applications or industry problems, but quite rarely in problems defined in commonly occurring modelling languages such as AMPL (Fourer, Gay and Kernighan 2003) or GAMS (Brooke, Kendrick and Meeraus 1988).

Computation of derivatives

In the early days, researchers invested much effort in finding methods with modest derivative requirements. Typically function values and, sometimes, gradients were available, but second derivatives frowned upon. The advent of automatic differentiation (Griewank 2000) and (group) partial separability (Griewank and Toint 1982*b*, Conn, Gould and Toint 1990) has somewhat altered this position at least amongst researchers, and now methods that are designed to exploit second derivatives (or good approximations thereof) are commonplace. But it is arguable that such new methods have not been as widely used by practitioners as might have been hoped, often because application codes capable of computing function values are unnameable to automatic differentiation for a variety of reasons, size and unavailability of the source-code being two common complaints. Indeed, there are still many practitioners who prefer methods that avoid derivatives at all (Powell 1998), although such methods are usually only appropriate for small-scale problems (but see Colson and Toint (2003) or Price and Toint (2004) for recent attempts to extend these techniques to large-scale cases).

Automatic differentiation offers the possibility of computing gradients and Hessian-vector products at a few times the cost of a function value (Griewank 2000). Tools for automatic differentiation are available both as stand-alone software or as part of modelling languages (AMPL and GAMS being good examples). Partial separability allows the computation of finite-difference gradients at a similar cost if only function values are available,

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

and the same for Hessians if (structured) gradients can be found (Conn *et al.* 1990). Moreover, accurate structured secant approximations to second derivatives can be computed (Griewank and Toint 1982*b*), and this allows one to approximate gradients (by finite-differences) and Hessians (by secant formulae) just given function values if the problem functions are partially separable and the structure specified (Conn, Gould and Toint 1996).

Note that these comments on evaluating derivatives are of interest not only for unconstrained problems, but also for most of the other problems that are discussed in this paper. In the constrained case, the derivative of the constraint and Lagrangian functions will also be concerned, and the techniques to compute them are similar to what we have just mentioned.

Computation of the step

Even if function and derivative values are available, in general the cost of computing the step d_k may be significant if the problem involves a large number of variables. This computation often follows the following line: if H_k is a symmetric positive definite approximation to $\nabla_{xx} f(x_k)$, if the quasi-Newton (QN) model

$$m_k(d) = f(x_k) + d^T \nabla_x f(x_k) + \tfrac{1}{2} d^T H_k d \tag{2.1}$$

is used, and if the minimizer of this model is sought, the resulting step d_k satisfies the QN equations

$$H_k d_k = -\nabla_x f(x_k). \tag{2.2}$$

Since H_k is positive definite, realistic solution options include a (sparse) Cholesky factorization of H_k or application of the (preconditioned) conjugate gradient (CG) method (Hestenes and Stiefel 1952). The former may not be viable if the factors fill in significantly, but is capable of giving a numerical solution with small relative error. The latter is more flexible – rather than needing H_k, it merely requires a series of products $H_k p$ for given vectors p (and possibly preconditioned residuals $r = P_k^{-1} g$ for some suitable symmetric preconditioner P_k), and thus is better equipped for automatic differentiation or finite-difference gradient approximations $(\nabla_x f(x_k + \epsilon p) - \nabla_x f(x_k))/\epsilon$ for small ϵ – but less likely to be able to compute highly accurate numerical solutions of (2.2). When the approximation H_k is indefinite, it may be modified during factorization (Schlick 1993) or as the CG process proceeds (Nash 1984) to restore definiteness. Alternatively, the CG method may be terminated appropriately as soon as one of the products $H_k p$ in the CG method reveals negative curvature (Dembo and Steihaug 1983) or even continued in the subspace of positive curvature whilst gathering negative curvature information (Gould, Lucidi, Roma and Toint 2000).

A significant breakthrough for large-scale unconstrained optimization occurred in the early 1980s with the advent of truncated-QN methods (Dembo,

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

Eisenstat and Steihaug 1982). Here, rather than requiring that d_k satisfies (2.2), instead d_k is asked to satisfy

$$\|H_k d_k + \nabla_x f(x_k)\| \leq \eta_k \|\nabla_x f(x_k)\|, \tag{2.3}$$

where $0 < \eta_k < 1$ and $\eta_k \to 0$ if $\nabla_x f(x_k) \to 0$. This is helpful for use in conjunction with CG methods, since one could anticipate being able to satisfy (2.3) after few CG iterations for modest values of η_k. But more significantly – and perhaps overlooked by those who view CG as simply a method for solving linear systems – the iterates $\{d_{k,j}\}_{j\geq 0}$ generated by the CG method from x_k have two further fundamental properties. Firstly, by construction each successive CG step further reduces the model, that is, $m_k(d_{k,j+1}) < m_k(d_{k,j})$ for $j \geq 0$. Secondly, an appropriate norm of the CG iterates increases at each step, that is, $\|d_{k,j+1}\| > \|d_{k,j}\|$ for $j \geq 0$ (Steihaug 1983). This enables one to construct globally convergent linesearch (Dembo and Steihaug 1983) and trust region (Steihaug 1983, Toint 1981) truncated Newton methods, *i.e.*, methods that converge to local solutions from arbitrary starting points. In the linesearch case, d_k is chosen as the first $d_{k,j}$ for which (2.3) is satisfied, unless negative curvature is discovered when computing the required product $H_k p$ at CG iteration j, in which case either the steepest descent direction $-\nabla_x f(x_k)$ (when $j = 0$) or the current CG approximation $d_{k,j-1}$ (when $j > 0$) may be used instead (Dembo and Steihaug 1983). For the trust region case, such methods should be stopped on the trust region boundary if $\|d_{k,j}\| > \Delta_k$ or negative curvature is discovered, since once the CG iterates leave the trust region they will not return (Steihaug 1983). By judicious control of η_k in (2.3), such methods may also be shown to be superlinearly convergent under reasonable conditions on the approximation H_k to $\nabla_{xx} f(x_k)$.

In the trust region case, an accurate solution of the model problem needs to account for the trust region constraint $\|d\| \leq \Delta_k$. When H_k is positive semi-definite, the strategy of truncating the CG iteration on the trust region boundary (Steihaug 1983, Toint 1981) ensures a model decrease which is at least half as good as the optimal decrease (Yuan 2000). For indefinite H_k this is not so. Although there are excellent methods for solving the problem in the small-scale case (Moré and Sorensen 1983), these rely on being able to solve a (small) sequence of linear systems with coefficient matrices $H_k + \sigma_{k,l} I$ for given $\sigma_{k,l} \geq 0$, and thus implicitly on being able to factorize each coefficient matrix. Since this may be expensive or even impossible in the large-scale case, an alternative is to note that the CG and Lanczos methods compute different bases for the same Krylov space, and that after j steps of the Lanczos method, $Q_{k,j}^T H_k Q_{k,j} = T_{k,j}$ where the columns of the n by j matrix $Q_{k,j}$ are orthonormal and $T_{k,j}$ is tridiagonal. Thus if we seek an approximation to the solution of the trust region problem

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

in the range of the expanding matrix $Q_{k,j}$, we may compute

$$d_{k,j} = Q_{k,j} h_{k,j}, \quad \text{where} \quad h_{k,j} = \underset{\|h\| \leq \Delta_k}{\arg\min} \; e_1^T Q_{k,j}^T \nabla_x f(x_k) e_1^T h + \tfrac{1}{2} h^T T_{k,j} h,$$

where $e_1 = [1, 0, 0, \ldots, 0]^T$. Since $T_{k,j}$ is tridiagonal, we may reasonably factorize $T_{k,j} + \sigma_{k,j,l} I$, and thus the earlier Moré–Sorensen method is now applicable (Gould, Lucidi, Roma and Toint 1999). The Lanczos iteration may be truncated in a similar way to (2.3), preconditioning may be readily incorporated, and the resulting so-called GLTR method has been used as a subproblem solver in a number of large-scale optimization packages (Byrd, Gould, Nocedal and Waltz 2004*a*, Gould, Orban and Toint 2003*a*). Other iterative methods for the exact minimization of (2.1) within the trust region have been proposed (Hager 2001, Rendl and Wolkowicz 1997, Sorensen 1997), but as far as we are aware they have not been used in truncated form.

Another popular and effective method is the limited-memory secant approach (Gilbert and Lemaréchal 1989, Liu and Nocedal 1989, Nocedal 1980). Secant methods maintain Hessian approximations by sequences of low-rank updates, each using a pair of vectors (d_k, y_k), where $y_k = \nabla_x f(x_{k+1}) - \nabla_x f(x_k)$, to satisfy the secant condition $H_k d_k = y_k$ (Nocedal and Wright 1999, §2.2). Noting the success of (particularly) the BFGS secant method for small-scale computation, and recognizing that such methods are generally inappropriate for large problems because the generated matrices are almost invariably dense, the idea of limited memory methods is simply to use no more than m pairs $\{(d_j, y_j)\}_{j=k-m+1}^{k}$ to generate a secant approximation from a given, easily invertible initial matrix. If m is small, application of the resulting limited-memory approximation H_k or its inverse to a given vector may be performed extremely efficiently (Byrd, Nocedal and Schnabel 1994). Although this approach is perhaps most natural in a linesearch framework – because the QN direction $-H_k^{-1} \nabla_x f(x_k)$ is easy to obtain – it may also be used in a trust region one (Burke and Weigmann 1997, Kaufman 1999).

Since estimating H_k directly by secant methods is likely to be out of the question for large problems, an alternative we have already briefly mentioned is to exploit problem structure, and most especially partial separability, to obtain good Hessian approximations. By definition, a partially separable function has the form $f(x) = \sum_i f^{(i)}(x)$, where each element $f^{(i)}$ has a large invariant subspace. Thus it is reasonable to approximate $\nabla_{xx} f(x)$ by $\sum_i H^{(i)}$, where each $H^{(i)}$ approximates the low-rank element Hessian $\nabla_{xx} f^{(i)}(x)$. So-called partitioned QN methods (Griewank and Toint 1982*c*) use suitable secant formulae to build (often highly accurate) approximations $H^{(i)}$. Although the resulting $H_k = \sum_i H_k^{(i)}$ may not be as easily inverted as, say, that from a limited-memory method, it often gives more accurate approximations, and has been used with great success within a truncated CG framework (Conn *et al.* 1990).

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

The final major class of methods are nonlinear variants of the CG method. Briefly, these methods aim to mimic the linear CG approach, and the step d_k is updated every iteration so that

$$d_{k+1} = -\nabla_x f(x_k) + \beta_k d_k$$

for some appropriate scalar β_k. Such methods have a long pedigree (Fletcher and Reeves 1964, Gilbert and Nocedal 1992, Polak and Ribière 1969, Powell 1977). Early methods chose β_k using formulae derived from the linear CG method, but sometimes subsequent steps tended to be closely dependent. A number of modifications have been proposed to avoid this defect, many of them resorting to steps in, or close to, the steepest-descent direction. The most successful recent methods (Dai and Yuan 2000, Hager and Zhang 2003) achieve this seamlessly, and additionally use linesearches with weak step-size acceptance criteria.

Practicalities

Despite the large number of papers devoted to large-scale unconstrained optimization, it is quite difficult to find comparisons between the various approaches proposed. A 1991 survey by Nash and Nocedal (1991) compares the limited-memory L-BFGS method (Liu and Nocedal 1989) with both the (early) Polak–Ribière nonlinear CG method (Polak and Ribière 1969) and a truncated-Newton method in which Hessian-vector products are obtained by differences. Although the results are mixed, the truncated-Newton approach seems preferable for problems well-approximated by a quadratic while L-BFGS appears best for more nonlinear problems. The nonlinear CG method is often best in terms of time, but requires more function evaluations. A contemporary survey by Gilbert and Nocedal (1992) which compares various nonlinear CG methods indicates there is little to choose between variants on the Polak–Ribière theme. However, while the test problems might have been large by 1990 standards, they are certainly not by today's. The only recent comparison we are aware of is that by Hager and Zhang (2003), in which their modern nonlinear CG method is compared with L-BFGS and Gilbert and Nocedal's (1992) improvement to Polak–Ribière. At least on the basis of these tests, modern nonlinear CG appears to be the method of choice if second derivatives are unavailable. However, we should exercise some caution as again the problems were not really large by today's standard, nor do we know how second-derivative-based truncated-Newton fits into the picture.

Two other issues are vital for good performance of many of the methods we have discussed. The first is preconditioning, where beyond very simple ideas such as diagonal or band scaling using Hessian terms (Conn *et al.* 1990), little has been done except for using standard incomplete factorization ideas from numerical linear algebra – Lin and Moré's (1999*a*)

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

memory-conserving incomplete factorization is widely used in optimization circles. One interesting idea is to use a limited-memory approximation to H_k to precondition the next subproblem H_{k+1} (Morales and Nocedal 2000), although more experience is needed to see if this is generally applicable.

The second important advance is based on the observation that while there should be some overall monotonically reducing trend of function values in algorithms for minimization, this is not necessary for every iteration (Grippo, Lampariello and Lucidi 1986). Non-monotonic methods for unconstrained problems were first proposed in a linesearch framework (Grippo, Lampariello and Lucidi 1989), and have been observed to offer significant gains when compared with their monotone counterparts (Toint 1996). The same is true in a trust region context (Deng, Xiao and Zhou 1993, Toint 1997), and many algorithms now offer non-monotonic variants (Gould *et al.* 2003*a*).

Another technique that exploits the potential benefits of non-monotonicity uses the idea of filters. Inspired by multi-objective optimization and originally intended by Fletcher and Leyffer (2002) for constrained problems (see Section 5.1 below), the aim of a *filter* is to allow conflicting abstract objectives within the design of numerical algorithms. To understand the idea, consider an abstract situation where an algorithm attempts to simultaneously reduce two potentially conflicting objectives $\theta_1(x)$ and $\theta_2(x)$. A point x is then said to dominate another point y if and only if $\theta_i(x) < \theta_i(y)$ for $i = 1$ *and* 2 (this definition can obviously be generalized to more than two conflicting objectives). Remembering a dominated y is of little interest when aiming to reduce both θ_1 and θ_2 since x is at least as good as y for each objective. Obviously, an algorithm using this selection criterion should therefore store some or all pairs (θ_1, θ_2) corresponding to successful previous iterates.

It turns out that this concept allows the design of new non-monotonic techniques for unconstrained minimization. For convex problems, we know that finding the (unique) minimizer is equivalent to finding a zero of the gradient. This in turn may be viewed as the (potentially conflicting) objective of zeroing each of the n gradient components $[\nabla_x f(x)]_i$ $(i = 1, \ldots, n)$. One may therefore decide that a new trial point $x_k + d_k$ is not acceptable as a new iterate only if it is dominated by x_p, one of (a subset of) the previous iterates, in the sense that

$$|[\nabla_x f(x_p)]_i| < |[\nabla_x f(x_k + d_k)]_i| \tag{2.4}$$

for all $i = 1, \ldots, n$, which corresponds to the choice $\theta_i(x) = |[\nabla_x f(x_k)]_i|$ $(i = 1, \ldots, n)$. The subset of previous iterates x_p for which the values of the gradient components are remembered and this comparison conducted is called the 'filter' and is maintained dynamically. If $x_k + d_k$ is not acceptable according to (2.4), it can still be evaluated using the more usual trust region

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

technique, which then guarantees that a step is eventually acceptable and that a new iterate can be found. Unfortunately, this technique might prevent progress away from a saddle point for nonconvex problems, in which case an increase in the gradient components is warranted. The filter mechanism is thus modified to dynamically disregard the filter in these cases. The details of the resulting algorithm are described by Gould, Sainvitu and Toint (2004), where encouraging numerical results are also reported on both small- and large-scale problems.

Software

There is a lot of easily available software for unconstrained minimization. Here, and later, we refer the reader to the on-line software guides

`http://www-fp.mcs.anl.gov/otc/Guide/SoftwareGuide/` and `http://plato.asu.edu/guide.html`,

by Moré and Wright, and Mittelmann and Spellucci, respectively. Of the methods discussed in this section, `TN/TNBC` (Nash 1984) is a truncated CG method, `LBFGS` (Liu and Nocedal 1989) is a limited-memory QN method, `VE08` (Griewank and Toint 1982*c*) is a partitioned QN method, and `CG+` (Gilbert and Nocedal 1992) and `CG_DESCENT` (Hager and Zhang 2003) are nonlinear CG methods. In addition, software designed for more general problems – for example `IPOPT`, `KNITRO`, `LANCELOT`, `LOQO` and `TRON` – is often more than capable when applied in the unconstrained case.

2.2. Least-squares problems

Nonlinear least-squares problems, for which

$$f(x) = \frac{1}{2}\sum_{i=1}^{m} f_i^2(x),$$

are perhaps the major source of really unconstrained problems. In particular, large sets of nonlinear equations, parameter estimation in large dynamical systems and free surface optimization often result in sizeable and difficult instances (see Gould and Toint (2004*a*) for examples). Methods for solving problems of this type follow the general trends of Section 2.1, but specifically exploit the special form of the objective function to select – sometimes adaptively (Dennis, Gay and Welsh 1981) – between the 'full QN' model, where the matrix H_k in (2.1) is chosen to approximate the Hessian

$$\nabla_{xx} f(x_k) = J(x_k)^T J(x_k) + \sum_{i=1}^{m} f_i(x_k)\nabla_{xx} f_i(x_k)$$

(where $J(x)$ is the $m \times n$ matrix whose rows are the gradients $\nabla_x f_i(x)$), and

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

the cheaper 'Gauss–Newton' model, for which $H_k = J(x_k)^T J(x_k)$. Furthermore, algorithmic stopping criteria can be adapted to exploit the special structure of $\nabla_x f(x)$ and the fact that zero provides an obvious lower bound on the value of the objective function.

Apart from the contributions of Al-Baali (2003) on dedicated QN updates, the work of Lukšan (1993, 1994, 1996) on incorporating iterative linear algebra techniques in trust region algorithms for nonlinear least-squares and that of Gulliksson, Söderkvist and Wedin (1997) on handling weights (and constraints), there has been little recent research in this area. Of course, most new ideas applicable to general unconstrained optimization may also be applied in the nonlinear least-squares case.

This is in particular the case for filter methods. In this context, the idea is to associate one filter objective $\theta_i(x)$ with each residual, *i.e.*, $\theta_i(x) = f_i(x)$ $(i = 1, \ldots, m)$, or perhaps with the norm of a block of residuals, *i.e.*, $\theta_i(x) = (\sum_{j \in J_i} f_j^2(x))^{\frac{1}{2}}$ for some $J_i \subset \{1, \ldots, m\}$. Details along with proofs of convergence are given by Gould, Leyffer and Toint (2005). Such ideas may be trivially extended to incorporate inequality constraints, thus providing practical means for solving the nonlinear feasibility problems (that is, to find a solution to a set of nonlinear equality and inequality constraints in the least-squares sense). Numerical efficiency and reliability is considered by Gould and Toint (2003*a*).

Software

The only dedicated large-scale nonlinear least-squares packages we are aware of are the sparsity-exploiting `SPRNLP` (Betts and Frank 1994), `VE10` (Toint 1987), which uses the obvious partially separable structure of such problems, and the filter-based code FILTRANE from the GALAHAD library (Gould *et al.* 2003*a*). Of course much general-purpose software is applicable to nonlinear least-squares problems.

2.3. Discretized problems

In practice, many large-scale finite-dimensional unconstrained optimization problems arise from the discretization of those in infinite dimensions, a primary example being least-squares parameter identification in systems defined in terms of either ordinary or partial differential equations. The direct solution of such problems for a given discretization yielding the desired accuracy is often possible using general packages for large-scale numerical optimization (see Section 2.1). However, such techniques rarely make use of the underlying infinite-dimensional nature of the problem, for which several levels of discretization are possible, and thus such an approach rapidly becomes cumbersome. Multi-scale (sometimes known as multi-level)

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

optimization aims at making explicit use of the problem structure in the hope of improving efficiency and, possibly, enhancing reliability.

Using differing scales of discretization for an infinite-dimensional problem is not a new idea. An obvious simple 'mesh refinement' approach is to use coarser grids in order to compute approximate solutions which can then be used as starting points for the optimization problem on a finer grid (Griewank and Toint 1982*a*, Bank, Gill and Marcia 2003, Betts and Erb 2003, Benson, McInnes, Moré and Sarich 2004). However, potentially more efficient techniques are inspired by the multigrid paradigm in the solution of partial differential equations and associated systems of linear algebraic equations (Brandt 1977, Bramble 1993, Hackbusch 1995, Briggs, Henson and McCormick 2000), and have only been discussed relatively recently in the optimization community. Contributions along this direction include the 'generalized truncated Newton algorithm' presented in Fisher (1998), and those by Moré (2003), Nash (2000*a*) and Lewis and Nash (2002, 2005). The latter three papers present the description of `MG/OPT`, a linesearch-based recursive algorithm, an outline of its convergence properties and impressive numerical results. The generalized truncated Newton algorithm and `MG/OPT` are very similar and, like many linesearch methods, naturally suited to convex problems, but their extension to nonconvex cases is also possible. Very recently, Gratton, Sartenaer and Toint (2004) have proposed a recursive multi-scale trust region algorithm (`RMTR`) which fits nonconvex problems more naturally and is backed by a strong convergence theory. The main idea of all the methods mentioned here is to (recursively) exploit the cheaper optimization on a coarse mesh to produce steps that significantly decrease the objective function on a finer mesh, while of course continuing to benefit from mesh refinement for obtaining good starting points. In principle, low frequency components of the problem solution (after suitable prolongation in the original infinite-dimensional space of interest) are determined by the coarse mesh calculations, and optimizing on the fine mesh then only fixes high frequency components.

While the idea appears to be very powerful and potentially leads to the solution of very large-scale problems, the practical algorithms that implement them are still mostly experimental. Preliminary numerical results are encouraging, but the true potential of these methods will only be confirmed by continued success in the coming years.

A second interesting approach to very large problems arising from continuous applications is to look at other ways to simplify them and make them more amenable to classical optimization techniques. For instance, Arian, Fahl and Sachs (2000) and Fahl and Sachs (2003) investigate the use of reduced-order models (using proper orthogonal decomposition techniques) in the framework of trust region algorithms, and apply this technique to fluid-mechanics problems. Note that model simplification of that type can

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

also be thought of as a recursive process, although not immediately based on discretization. The idea is thus close in spirit to the proposals described above. Again, the practical power of this approach, although promising at this stage, is still the object of ongoing evaluation.

3. Large-scale bound-constrained optimization

In the simplest of constrained optimization problems, we seek the minimizer of $f(x)$ within a feasible box, $\Omega = \{x \mid l \leq x \leq u\}$ for given (possibly infinite) lower and upper bounds l and u. Without loss of generality, we assume that $l_i < u_i$ for all $i = 1, \ldots, n$. It has been argued that all unconstrained problems should actually include simple bounds to prevent bad effects of computer arithmetic such as overflows, and certainly many real problems have simple bounds to prevent unreasonable or physically impossible values.

Active set methods

Early methods for this problem tended to be of the active set variety. The active set at x is $\mathcal{A}(x) = \mathcal{L}(x) \cup \mathcal{U}(x)$, where $\mathcal{L}(x) = \{i \mid x_i = l_i\}$ and $\mathcal{U}(x) = \{i \mid x_i = u_i\}$. Trivially, if x_* is a (local) minimizer of f within Ω, x_* is a (local) minimizer of $f(x)$ subject to $x_i = l_i,\ i \in \mathcal{L}(x_*)$ and $x_i = u_i,\ i \in \mathcal{U}(x_*)$. Active set methods aim to predict $\mathcal{L}(x_*)$ and $\mathcal{U}(x_*)$ using suitably chosen disjoint sets $\mathcal{L},\ \mathcal{U} \subseteq \{1, \ldots, n\}$. Given $\mathcal{L}$ and $\mathcal{U}$, a typical method will aim to (approximately)

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to } x_i = l_i,\ i \in \mathcal{L} \ \text{ and } \ x_i = u_i,\ i \in \mathcal{U}; \end{aligned}$$

such a calculation is effectively an unconstrained minimization over the variables $(x_i),\ i \notin \mathcal{A} = \mathcal{L} \cup \mathcal{U}$, and thus any of the methods mentioned in Section 2 is appropriate. Of course the predictions $\mathcal{L}$ and $\mathcal{U}$ may be incorrect, and the 'art' of active set methods is to adjust the sets as the iteration proceeds either by adding variables which violate one of their bounds or by removing those for which further progress is predicted – the same idea is possible (and indeed used) to deal with more general inequality constraints. See Gill *et al.* (1981, §5.5) or Fletcher (1987*a*, §10.3) for more details. Especially effective methods for the quadratic programming case, for which f is quadratic, have been developed (Coleman and Hulbert 1989).

The main disadvantage of (naive) active set methods for large-scale problems is the potential worst-case complexity in which each of the possible 3^n active sets is visited before discovering the optimal one. Although it is possible to design active set methods for the simple-bound case that are capable of making rapid changes to incorrect predictions (Facchinei, Judice and Soares 1998), it is now more common to use gradient-projection methods.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

Gradient-projection methods

The simplest gradient-projection algorithm (Bertsekas 1976, Dunn 1981, Levitin and Polyak 1966) is the obvious linesearch extension of the steepest-descent method to deal with convex constraints, and is based on the iteration

$$x_{k+1} = P_\Omega[x_k - \alpha_k \nabla_x f(x_k)],$$

where $P_\Omega(v)$ projects v into Ω and α_k is a suitable step-size. In the case of simple bounds, $P_\Omega[v] = \mathrm{mid}(l, v, u)$, the (componentwise) median of v with respect to the bounds l and u, is trivial to compute. The method possesses one extremely helpful feature: for non-degenerate problems (*i.e.*, those for which the removal of one or more active constraints necessarily changes the solution), the optimal 'face' of active constraints will be determined in a finite number of iterations (Bertsekas 1976). Of course, its steepest-descent ancestry hints that this is unlikely to be an effective method as it stands, and some form of acceleration is warranted.

The simplest idea exploits the finite optimal-face identification property: if the active faces visited by consecutive iterates of the gradient-projection algorithm are identical, a higher order (Newton-like) method should be used to investigate this face. This was first suggested for quadratic f (Moré and Toraldo 1991), but is now commonplace for general objectives.

A natural question is whether there are other algorithms which have the finite optimal-face identification property for non-degenerate problems. It turns out that the result is true for any algorithm for convex constraints for which the projected gradient $P_{T(x)}[-\nabla_x f(x)]$ converges to zero (Burke and Moré 1988, Calamai and Moré 1987) – here $T(x)$ is the closure of the cone of all feasible directions (the tangent cone) at x. Although the (discontinuous) projected gradient is often hard to compute, in the simple-bound case it is merely (componentwise)

$$\left(P_{T(x)}[-\nabla_x f(x)]\right)_i = \begin{cases} -\min\{0, (\nabla_x f(x))_i\} & \text{if } x_i = l_i \\ -(\nabla_x f(x))_i & \text{if } l_i < x_i < u_i \text{ and} \\ -\max\{0, (\nabla_x f(x))_i\} & \text{if } x_i = u_i. \end{cases}$$

Its continuous variant $P_\Omega[x_k - \nabla_x f(x_k)] - x_k$ is sometimes preferred, and plays a similar role in theory and algorithms.

A restricted version of the finite identification result also holds in the degenerate case, namely that the set of strongly active constraints (*i.e.*, those whose removal will change the solution) will be identified in a finite number of iterations if the projected gradient converges to zero (Lescrenier 1991, Burke and Moré 1994). These finite-identification results apply to many contemporary methods.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

Trust region methods for the problem typically consider the gradient-projection arc

$$d(\alpha) = P_{\Omega \cap \{y \mid \|y - x_k\| \leq \Delta_k\}}[x_k - \alpha \nabla_x f(x_k)] - x_k,$$

from x_k. Given a QN model $m_k(d)$, a so-called (generalized) Cauchy point $d(\alpha_k^{\rm C})$ is found by approximately minimizing $m_k(d)$ along $d(\alpha)$; either the first local arc minimizer (Conn, Gould and Toint 1988*a*) or a point satisfying sufficient-decrease linesearch conditions (Burke, Moré and Toraldo 1990, Toint 1988) is required – the computation of a suitable Cauchy point may be performed very efficiently when the Hessian is sparse (Conn, Gould and Toint 1988*b*, Lin and Moré 1999*b*). Thereafter a step d_k is computed so that

$$x_k + d_k \in \Omega, \quad \|d_k\| \leq \Delta_k \ \text{ and } \ m_k(d_k) \leq m_k(d_k^{\rm C}), \tag{3.1}$$

and the usual trust region acceptance rules applied (*e.g.*, Conn *et al.*, 2000*a*, §6.1). Since it has been shown that the projected gradient converges to zero for these methods, the flexibility in (3.1) is typically used to accelerate convergence by allowing a truncated CG method to explore the face of active constraints at $x_k + d_k^{\rm C}$. Since the CG iterates may try to leave Ω, early methods simply fixed variables to their bounds and restarted the CG iteration (Conn *et al.* 1988*a*), while more modern ones allow infeasible CG iterates by periodically projecting them back into Ω (Gould *et al.* 2003*a*, Lin and Moré 1999*b*).

If second derivatives are unavailable, they may be estimated by any of the methods discussed in Section 2. A particularly appealing approach is to use a limited-memory secant method to estimate the Hessian. Although this approximation is dense, it is so structured that a generalized Cauchy point may still be calculated. Moreover, one of the advantages of limited memory methods, namely that the QN step may be computed directly, is retained, despite the requirement that the QN step be restricted to the face determined by $d_k^{\rm C}$, by judicious use of the Sherman–Morrison–Woodbury formula (Byrd, Lu, Nocedal and Zhu 1995).

Although we have only considered methods which remain feasible with respect to the bounds, there is no theoretical reason – as long as the objective function is well-defined outside Ω – to do so provided there is some mechanism for ensuring that the iterates are asymptotically feasible (Facchinei, Lucidi and Palagi 2002). It is also unsurprising that, just as in the unconstrained case, there is no need for the objective function to decrease monotonically as long as there is some overall monotonic trend (Facchinei *et al.* 1998, Gould *et al.* 2003*a*). Efforts have also been made to embed nonlinear CG methods within a gradient-projection framework (Pytlak 1998). Filter ideas have also been investigated (Gould and Toint 2003*a*) that use penalty techniques (see Section 5) to handle the bounds. Research is on-

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

going to merge filter methods with the projection methods discussed above or the interior-point techniques discussed below.

Interior-point methods

Interior-point methods provide an alternative means of solving bound-constrained problems. For simplicity, consider the case where $\Omega = \{x \mid x \geq 0\}$, suppose that μ is a positive – so-called *barrier* – parameter, and let

$$\phi(x, \mu) \stackrel{\text{def}}{=} f(x) - \mu \sum_{i=1}^{n} \log(x)_i$$

be the logarithmic barrier function for the problem, where $(x)_i$ denotes the ith component of x. The key idea is to trace approximate minimizers of $\phi(x, \mu)$ as μ decreases to zero. Under reasonable assumptions, and for sufficiently small positive values of μ, (local) minimizers of $\phi(x, \mu)$ exist and describe continuous trajectories – primal central paths – converging to (local) solutions of the required problem (Fiacco and McCormick 1968, Wright 1992). Likewise, if X is the diagonal matrix whose ith diagonal element is the ith component of x and e is a vector of ones, the associated (first-order) dual variables estimates

$$z = \mu X^{-1} e, \tag{3.2}$$

are located on trajectories enjoying similar properties and converge to Lagrange multipliers associated with the bound constraints. The cross-product of each pair of trajectories is known as a *primal–dual central path*, and most barrier methods attempt to follow one with increasing accuracy as μ decreases (Fiacco and McCormick 1968, Wright 1992); for this reason, interior-point methods are sometimes also referred to as *path-following methods*.

The unconstrained minimization of ϕ can be handled using the techniques described in Section 2 as long as care is taken to ensure that the iterates remain within the interior of Ω. A QN model of the form (2.1) might be used, and as such would be

$$m_k(d) = \phi(x_k, \mu) + d^T(\nabla_x f(x_k) - \mu X_k^{-1} e) + \tfrac{1}{2} d^T (H_k + \mu X_k^{-2}) d, \tag{3.3}$$

where H_k is, as before, an approximation to $\nabla_{xx} f(x_k)$. However, considerable numerical experience has shown that it is usually preferable to replace the first-order dual variable estimates $z_k = \mu X_k^{-1} e$ in the Hessian term of (3.3) to obtain instead

$$m_k(d) = \phi(x_k, \mu) + d^T(\nabla_x f(x_k) - \mu X_k^{-1} e) + \tfrac{1}{2} d^T (H_k + X_k^{-1} Z_k) d, \tag{3.4}$$

and to compute the dual variable z_k by other means. In this case, since the optimal Lagrange multipliers for the problem are necessarily positive, it is reasonable to require the same of z_k. Rather than computing z_k explicitly

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

from (3.2), it is better to multiply both sides of (3.2) by X, giving $Xz = \mu e$. Applying Newton's method to this last system then yields the alternative

$$z_{k+1} = \mu X_k^{-1} e - X_k^{-1} Z_k d_k, \tag{3.5}$$

involving the current step d_k from x_k. Additional safeguards need to be employed to enforce convergence of the process (Conn *et al.* 2000*a*, Chapter 13). Methods of this latter type are referred to as *primal–dual methods* because they explicitly consider both primal and dual variables, while methods based on the model (3.3) (with its implicitly computed dual variables) are called *primal methods*.

An approximate minimizer of the model (3.4) may be computed by either a direct (factorization) or iterative (CG) method. If the latter is used, it is normally essential to precondition the iteration to remove the effects of the extreme eigenvalues of $X_k^{-1} Z_k$ (Luenberger 1984, Chapter 12). A preconditioner of the form $P_k = G_k + X_k^{-1} Z_k$ for some suitable approximation G_k of H_k is usually recommended, with G_k varying from naive ($G_k = 0$) to sophisticated ($G_k = H_k$).

Both linesearch or trust region globalization of interior-point methods are possible and essentially identical to that discussed in Section 2. The major difference in both cases is the addition of a so-called fraction-to-the-boundary rule, preventing iterates from prematurely approaching the boundary of the feasible set. A trust region algorithm will accept a step d_k from x_k if

(1) $x_k + d_k \geq \gamma_k x_k$ holds componentwise, for some given $0 < \gamma_k < 1$, and
(2) there is good agreement between the changes in m_k and $\phi(x, \mu)$.

For most practical purposes, the fraction-to-the-boundary parameter γ_k is held constant, a typical value being 0.005. It may however be permitted to converge to zero, allowing for fast asymptotic convergence. Wächter and Biegler (2004) choose (in a more general context) $\gamma_k = \max(\gamma_{\min}, \mu_k)$, where $0 < \gamma_{\min} < 1$ is a prescribed minimal value. The fraction-to-the-boundary rule also applies for linesearch methods, and a (backtracking) linesearch is typically performed until it is satisfied. A corresponding backtracking may then also be applied to the dual step to ensure consistency.

Although there is no difficulty in providing a strictly interior primal–dual starting point, (x_0, z_0), in the bound-constrained case, it is generally sensible to ensure that such a point is well separated from the boundary of the feasible region; failure to do this can (and in practice does seriously) delay convergence. Given suitable primal estimates x_0, traditional choices for dual estimates z_0 include the vector of ones or those given from (3.2) although there is little reason to believe that these are more than heuristics. An initial value for μ is then typically $\mu_0 = x_0^T z_0 / n$, so as to obtain good centrality at the initial point.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

It is well known that computing d_k to be a critical point of (3.4) and recovering z_{k+1} via (3.5) is equivalent to applying Newton's method from (x_k, z_k) to the perturbed optimality conditions

$$\nabla_x f(x) - z = 0 \;\text{ and }\; Xz = \mu e$$

for our problem. While it is tempting to try similar approaches directly with $\mu = 0$ – so-called affine methods – these have both theoretical and practical shortcomings for general problems (see, for example, Conn *et al.*, 2000*a*, §13.11). A more promising approach is to note that equivalent first-order optimality conditions are that

$$W(x)\nabla_x f(x) = 0, \tag{3.6}$$

where

$$W(x) = \operatorname{diag} w(x) \;\text{ and }\; w_i(x) = \begin{cases} x_i & \text{if } (\nabla_x f(x))_i \geq 0 \\ -1 & \text{otherwise.} \end{cases}$$

As long as strictly feasible iterates are generated, $W(x)\nabla_x f(x)$ is differentiable, and Newton's method may be applied to (3.6). To globalize such an iteration, combined linesearch/trust region methods have been proposed (Coleman and Li 1994, 1996) and variants which allow quadratic convergence even in the presence of degeneracy are possible (Heinkenschloss, Ulbrich and Ulbrich 1999).

Practicalities

There has been a number of comparative studies of algorithms for bound-constrained optimization (Facchinei *et al.* 2002, Gould *et al.* 2003*a*, Lin and Moré 1999*b*, Zhu, Byrd, Lu and Nocedal 1997), but we feel that none of these makes a compelling case as to the best approach(es) for the large-scale case. In practice, both gradient projection and interior-point methods appear to require a modest number of iterations.

Software

Once again there is a reasonable choice of reliable software for the bound-constrained case. Both `TRON` (Lin and Moré 1999*b*) and LANCELOT/SBMIN (Conn, Gould and Toint 1992) are trust region gradient-projection algorithms with subspace conjugate gradient acceleration – there is an improved version of the latter within GALAHAD (Gould *et al.* 2003*a*) – while `L-BFGS-B` (Zhu *et al.* 1997) is a linesearch implementation of the limited-memory approach. The MATLAB function fmincon (Branch, Coleman and Li 1999) uses an interior-point subspace method based on (3.6). FILTRANE, another algorithm of the GALAHAD library, uses a filter method combined with penalty techniques for the bounds. As before, more general codes such as `KNITRO` and `LOQO` are also highly appropriate.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

4. Large-scale linearly constrained optimization

As the next level of generality, we now turn to problems involving general linear constraints.

4.1. Equality-constrained quadratic programming

A – some might say *the* – basic subproblem in constrained optimization is to

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ q(x) = g^T x + \tfrac{1}{2} x^T H x \ \text{ subject to } \ Ax = b, \tag{4.1}$$

where H is symmetric (but possibly indefinite) and A is m by n and (without loss of generality) of full rank. Such equality-constrained quadratic programming (EQP) subproblems arise when computing search directions for either general methods for equality-constrained or active set methods for inequality-constrained optimization. It is actually often more convenient to consider the related homogeneous problem

$$\underset{\bar{x}\in\mathbb{R}^n}{\text{minimize}}\ \ \bar{q}(\bar{x}) = \bar{g}^T \bar{x} + \tfrac{1}{2} \bar{x}^T H \bar{x} \ \text{ subject to } \ A\bar{x} = 0; \tag{4.2}$$

as long as there is some easy way to find x_0 satisfying $Ax_0 = b$, the solutions of the two problems satisfy $x = \bar{x} + x_0$ provided that $\bar{g} = g + Hx_0$.

Critical points of (4.1) necessarily satisfy the augmented (KKT) system

$$\begin{pmatrix} H & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -g \\ b \end{pmatrix}, \tag{4.3}$$

and solutions of (4.3) can only be solutions of (4.1) if the coefficient matrix of (4.3) has precisely m negative eigenvalues (Chabrillac and Crouzeix 1984, Gould 1985). Thus direct (factorization) methods for solving (4.3) must be capable of coping with indefinite matrices; fortunately there is a growing number of highly capable symmetric, indefinite linear solvers (for example `BCSEXT`, `MA27/57`, `Oblio` or `PARDISO`; see Scott, Hu and Gould (2004) for a comparison), and in particular if H is (block) diagonal a Schur-complement decomposition involving factorizations of H and $-AH^{-1}A^T$ is often to be recommended. Nevertheless, for very large problems direct methods may be unviable or too expensive, and iterative methods may be the only alternative. Although non-symmetric or indefinite iterative methods may be applied, we only consider CG-type methods here, since these have the desirable property of decreasing $q(x)$ at every iteration.

It should be apparent that CG methods can be applied explicitly to (4.2) by computing a basis N for the null-space of A, and then using the transformation $\bar{x} = Nx_n$ to derive the equivalent (unconstrained) problem of minimizing $q_n(x_n) = x_n^T g_n + \frac{1}{2} x_n^T H_n x_n$, where $g_n = N^T \bar{g}$ and $H_n = N^T H N$ are known as the reduced gradient and Hessian respectively. Perhaps not so obviously, the same may be achieved *implicitly* by using the standard

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

preconditioned CG (PCG) method but using a block (so-called), constraint preconditioner of the form

$$\begin{pmatrix} G & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} r \\ w \end{pmatrix} = -\begin{pmatrix} \bar{g} + Hx \\ 0 \end{pmatrix}, \tag{4.4}$$

to obtain the 'preconditioned' residual r from the 'unpreconditioned' $\bar{g}+Hx$, for some suitable G (Coleman 1994, Gould, Hribar and Nocedal 2001, Lukšan and Vlček 1998, Polyak 1969). Various choices for G, ranging from the identity matrix to H, have been suggested, and all require a suitable (block) factorization of the coefficient matrix K of (4.4); basic requirements are that K should be non-singular and have precisely m negative eigenvalues. A further advantage of the PCG approach is that any additional (properly scaled) trust region constraint may easily be incorporated using the GLTR strategy mentioned in Section 2. Nevertheless, requiring a factorization of K may still be considered a disadvantage, and methods which avoid this are urgently needed.

4.2. General quadratic programming

Another important subproblem in constrained optimization is the general quadratic programming (QP) problem, namely to

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ q(x) \ \text{ subject to } \ A_{\mathcal{E}}x = b_{\mathcal{E}} \ \text{ and } \ A_{\mathcal{I}}x \geq b_{\mathcal{I}}. \tag{4.5}$$

Of particular interest is the non-convex case where the symmetric H may be indefinite, although in these circumstances we must normally be content with local solutions. The main application area we are concerned with is in solving subproblems which arise within sequential quadratic programming (SQP) algorithms for general nonlinear optimization (see Section 5.1, but note the caveat there that expresses our concerns over the SQP approach), although there are actually a large number of other (usually convex) applications of (4.5) (Gould, and Toint 2000*a*), including VLSI design, optimal control, economic dispatch and financial planning, to mention only a few. They also constitute a class apart as their necessary and sufficient optimality conditions coincide (Contesse 1980, Mangasarian 1980, Borwein 1982).

Active set methods for general quadratic programming

As was the case for bound-constrained problems we considered in Section 3, QP methods may broadly be categorized as either active-set-based or interior-point-based. As the name suggests, active set methods aim to predict which of the inequality constraints $A_{\mathcal{I}}x \geq b_{\mathcal{I}}$ are active at a solution to (4.5). At each iteration, a working set $\mathcal{W}_{\mathcal{I}} \subseteq \mathcal{I}$ is selected so that the gradients of the constraints $A_{\mathcal{W}}$, $\mathcal{W} = \mathcal{E} \cup \mathcal{W}_{\mathcal{I}}$, are linearly independent.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

For this working set, the EQP

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ q(x)\ \text{ subject to }\ A_{\mathcal{W}}x = b_{\mathcal{W}} \tag{4.6}$$

is solved (if possible) using one of the methods described in Section 4.1. There are a number of possibilities. If (4.6) is unbounded from below or if the solution to (4.6) violates one of the inequality constraints indexed by $\mathcal{I} \setminus \mathcal{W}_{\mathcal{I}}$, one (or more) constraints should be added to the working set. If (4.6) is infeasible or if the solution to (4.6) is not that of (4.5) – the latter is true if any of the Lagrange multipliers y from (4.3) are negative – one (or more) constraints should be removed from the working set. In the convex case, only when the solution to (4.6) satisfies all of the constraints indexed by $\mathcal{I} \setminus \mathcal{W}_{\mathcal{I}}$ and all of the Lagrange multipliers are positive can we be certain that we have solved (4.5). Unfortunately, for non-convex problems, even checking if such a critical point is a local minimizer may be (NP) hard (Murty and Kabadi 1987). While such a strategy is simple – it may be reduced to the solution of a sequence of EQPs – potentially a large number of iterations may be required. Fortunately, just as with the simplex method for linear programming (LP), the working set usually changes very gradually, and the potentially dominant cost of matrix factorization is lessened through updates to existing factors. Thus active set methods for QP usually comprise a large number of very cheap iterations – in contrast, interior-point methods require a few, more expensive ones.

There have been relatively few active set methods for large-scale QP, especially in the nonconvex case. The majority of these are based on the idea of inertia control (Fletcher 1971). Suppose that the coefficient matrix K for the optimality conditions (4.3) corresponding to the current EQP (4.6) has the 'correct' inertia, *i.e.*, K has $|\mathcal{W}|$ negative eigenvalues. If a constraint is added to the working set, the new subproblem will inherit the correct inertia. However, if a constraint is removed from the working set, it is possible that the resulting K may have $|\mathcal{W}| + 1$ rather than $|\mathcal{W}|$ negative eigenvalues. If this happens, there must be a feasible direction of negative curvature, and an inertia-controlling method will follow this direction until it encounters a currently inactive constraint (or perhaps q is unbounded from below on the feasible set). This new constraint will be added to the working set, and once again the resulting K will have either $|\mathcal{W}|$ or $|\mathcal{W}| + 1$ negative eigenvalues. In the former case, the correct inertia has been restored, while in the latter there is again a direction of feasible negative curvature. This process of following negative curvature and adding currently inactive constraints must ultimately terminate (unless the problem is unbounded below) at a vertex of the feasible region, at which point the correct inertia will have been restored. The principal differences between the inertia-controlling methods that have been proposed are the algebraic means

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

by which the factors are maintained and updated. These include using Schur-complement (Gill, Murray, Saunders and Wright 1990, Gill, Murray, Saunders and Wright 1991) or linear-programming basis-type (Gould 1991) updates to a factorization of an initial K, or Cholesky-factor updates of the (dense) reduced Hessian (Fletcher 2000), the latter only really being appropriate for problems with few degrees of freedom.

For problems for which a direct solution of the sequence of generated EQPs is unviable or too expensive, it is also possible to use the PCG method described in Section 4.1. Now rather than controlling the inertia of the KKT matrix, inertia control is only required for the preconditioner (4.4). Once again, factors of the preconditioner must adapt to changes in the working set, but the ability to choose G gives considerable flexibility (Gould and Toint 2002*a*).

Interior-point methods for general quadratic programming

It is easy to generalize the interior-point methods discussed in Section 3 to cope with the quadratic program (4.5). Denoting the ith row of $A_{\mathcal{I}}$ by a_i and the ith component of $b_{\mathcal{I}}$ by b_i, typical barrier methods for such problems aim to

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & \phi(x,\mu) \stackrel{\text{def}}{=} q(x) - \mu\sum_{i\in\mathcal{I}} \log(a_i^T x - b_i) \\ \text{subject to} \quad & A_{\mathcal{E}}x = b_{\mathcal{E}} \end{aligned} \tag{4.7}$$

as μ decreases to 0, while ensuring that x remains interior to $\Omega_{\mathcal{I}} = \{x \,|\, A_{\mathcal{I}}x \geq b_{\mathcal{I}}\}$. Just as in the bound-constrained case and under reasonable conditions, the minimizers of (4.7) and their dual variable (Lagrange multiplier) estimates

$$y_{\mathcal{I}} = \mu C_{\mathcal{I}}^{-1}(x)e, \quad \text{where} \quad C_{\mathcal{I}}(x) = \operatorname{diag} c_{\mathcal{I}}(x) \quad \text{and} \quad c_{\mathcal{I}}(x) = A_{\mathcal{I}}x - b_{\mathcal{I}},$$

define continuous trajectories – primal–dual central paths – leading to (local) solutions of (4.5).

For fixed μ and feasible x_k, basic iterative methods might compute a suitable step d_k by building a primal QN model

$$m_k(d) = \phi(x_k,\mu) + d^T(g_k - \mu A_{\mathcal{I}}^T C_{\mathcal{I}k}^{-1} e) + \tfrac{1}{2} d^T(H + \mu A_{\mathcal{I}}^T C_{\mathcal{I}k}^{-2} A_{\mathcal{I}})d, \tag{4.8}$$

where $g_k \stackrel{\text{def}}{=} Hx_k + g$ and $C_{\mathcal{I}k} \stackrel{\text{def}}{=} C_{\mathcal{I}}(x_k)$, and then trying to (approximately)

$$\underset{s\in\mathbb{R}^n}{\text{minimize}} \quad m_k(d) \quad \text{subject to} \quad A_{\mathcal{E}}s = 0 \quad \text{and (possibly)} \quad \|s\| \leq \Delta_k, \tag{4.9}$$

involving an additional trust region constraint. To ensure feasibility of the next iterate, a step-size $0 < \alpha_k \leq \alpha_k^{\max}$ should be imposed along an approximate solution d_k to (4.9) – a fraction-to-the-boundary rule such as

$$\alpha_k^{\max} = \max\bigl\{0 < \alpha \leq 1 \mid c_{\mathcal{I}}(x_k + \alpha d_k) \geq \gamma_k c_{\mathcal{I}}(x_k)\bigr\}$$

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

is appropriate – and to guarantee convergence it may also be necessary to linesearch along d_k or adjust Δ_k in the usual manner. But as in Section 3, a primal–dual model

$$m_k(d) = \phi(x_k, \mu) + d^T(g_k - \mu A_{\mathcal{I}}^T C_{\mathcal{I}k}^{-1} e) + \tfrac{1}{2} d^T (H + A_{\mathcal{I}}^T C_{\mathcal{I}k}^{-1} Y_{\mathcal{I}k} A_{\mathcal{I}}) d \tag{4.10}$$

involving explicit (positive) dual variables $y_{\mathcal{I}k}$ is generally preferable to (4.8) both in theory and in practice (Conn, Gould, Orban and Toint 2000*b*). In particular, the analogue of the Newton update (3.5),

$$y_{\mathcal{I}k+1} = \mu C_{\mathcal{I}k}^{-1} e - C_{\mathcal{I}k}^{-1} Y_{\mathcal{I}k} d_k, \tag{4.11}$$

is appropriate, so long as appropriate precautions are taken to modify (4.11) to ensure that $y_{\mathcal{I}k+1}$ remains sufficiently positive (Conn *et al.* 2000*a*, Chapter 13).

The key subproblem here is (4.9), but this is precisely of the form discussed in Section 4.1. The only significant extra issue when the objective function has the form (4.10) is that any preconditioner should respect the potentially ill-conditioned Hessian term (Luenberger 1984, Chapter 12), and thus that the leading block in (4.4) should be $G_k + A_{\mathcal{I}}^T C_{\mathcal{I}k}^{-1} Y_{\mathcal{I}k} A_{\mathcal{I}}$ for some suitable approximation G_k to H. Although it might first appear that such a leading block may be unacceptably dense, the preconditioning step (4.4) for the model (4.10) may be trivially rearranged to give the potentially sparser

$$\begin{pmatrix} G_k & A_{\mathcal{E}}^T & A_{\mathcal{I}}^T \\ A_{\mathcal{E}} & 0 & 0 \\ A_{\mathcal{I}} & 0 & -Y_{\mathcal{I}k}^{-1} C_{\mathcal{I}k} \end{pmatrix} \begin{pmatrix} r \\ w_{\mathcal{E}} \\ w_{\mathcal{I}} \end{pmatrix} = - \begin{pmatrix} g_k + Hs \\ 0 \\ -\mu Y_{\mathcal{I}k}^{-1} e \end{pmatrix} \tag{4.12}$$

for auxiliary variables $w_{\mathcal{I}} = C_{\mathcal{I}k}^{-1} Y_{\mathcal{I}k} A_{\mathcal{I}} s - \mu C_{\mathcal{I}k}^{-1} e$ (Gould 1986).

The method sketched above presupposes that an initial point x_0 is known within the intersection of $\Omega_{\mathcal{E}} = \{x \mid A_{\mathcal{E}} x = b_{\mathcal{E}}\}$ and the interior of $\Omega_{\mathcal{I}}$. A suitable point may be found by solving an auxiliary *phase-I* problem such as

$$\underset{x \in \mathbb{R}^n,\ s_{\mathcal{I}} \in \mathbb{R}^{n_{\mathcal{I}}}}{\text{minimize}} \ -\sum_{i \in \mathcal{I}} \log(s_i) \ \text{ subject to } \ A_{\mathcal{E}} x = b_{\mathcal{E}} \ \text{ and } \ A_{\mathcal{I}} x - s_{\mathcal{I}} = b_{\mathcal{I}}, \tag{4.13}$$

where the $s_{\mathcal{I}}$ are being treated as auxiliary, positive *slack* variables. The intention here is to find a point which is significantly interior, and in the above case will give the analytic centre of the feasible region. Fortunately, although finding feasible points for (4.13) may not be obvious, the problem is convex and may be solved using an infeasible interior-point method, such as those discussed in the next section.

With this in mind, an equivalent formulation of (4.5) is to

$$\underset{x \in \mathbb{R}^n,\ s_{\mathcal{I}} \in \mathbb{R}^{n_{\mathcal{I}}}}{\text{minimize}} \ q(x) \ \text{ subject to } \ A_{\mathcal{E}} x = b_{\mathcal{E}}, \ \ A_{\mathcal{I}} x - s_{\mathcal{I}} = b_{\mathcal{I}} \ \text{ and } \ s_{\mathcal{I}} \geq 0, \tag{4.14}$$

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

and an alternative barrier method for (4.5) might aim to

$$\begin{aligned} \underset{x\in\mathbb{R}^n,\, s_{\mathcal{I}}\in\mathbb{R}^{n_{\mathcal{I}}}}{\text{minimize}} \quad & \phi(x, s_{\mathcal{I}}, \mu) \stackrel{\text{def}}{=} q(x) - \mu \sum_{i\in\mathcal{I}} \log(s_i) \\ \text{subject to} \quad & A_{\mathcal{E}} x = b_{\mathcal{E}} \ \text{ and } \ A_{\mathcal{I}} x - s_{\mathcal{I}} = b_{\mathcal{I}} \end{aligned} \tag{4.15}$$

as μ decreases to 0. Although the distinction between this 'slack variable' formulation and (4.7) is actually very slight if the constraints $A_{\mathcal{I}} x - s_{\mathcal{I}} = b_{\mathcal{I}}$ are enforced throughout, as we shall see later the distinction is more pronounced for nonlinear constraints.

Interior-point methods for convex quadratic programming

Quadratic programs are traditionally classified as convex or nonconvex, depending on whether the Hessian matrix H is positive semidefinite or not. Simply finding a local minimizer of a nonconvex QP is an NP-hard problem (Vavasis 1990), as is proving that a first-order critical point is in fact a minimizer (Murty and Kabadi 1987) – most algorithms for general QP are consequently only designed to locate first-order critical points.

Convex QPs are provably solvable by algorithms having polynomial complexity (Nesterov and Nemirovskii 1994, Vavasis 1991). The use of a barrier function to force convergence in the convex case is usually inefficient, and the best methods are more closely allied to those for LP (Wright 1997). The basis of these primal–dual path-following methods for convex QP is to solve the perturbed optimality conditions

$$\begin{aligned} g + Hx - A_{\mathcal{E}}^T y_{\mathcal{E}} - A_{\mathcal{I}}^T y_{\mathcal{I}} &= 0, \\ A_{\mathcal{E}} x - b_{\mathcal{E}} &= 0, \\ \text{and } \ Y_{\mathcal{I}}(A_{\mathcal{I}} x - b_{\mathcal{I}}) - \mu e &= 0 \end{aligned} \tag{4.16}$$

for (4.5) or, more commonly,

$$\begin{aligned} g + Hx - A_{\mathcal{E}}^T y_{\mathcal{E}} - A_{\mathcal{I}}^T y_{\mathcal{I}} &= 0, \\ y_{\mathcal{I}} - z_{\mathcal{I}} &= 0, \\ A_{\mathcal{E}} x - b_{\mathcal{E}} &= 0, \\ A_{\mathcal{I}} x - s_{\mathcal{I}} - b_{\mathcal{I}} &= 0, \\ \text{and } \ Z_{\mathcal{I}} s_{\mathcal{I}} - \mu e &= 0 \end{aligned} \tag{4.17}$$

for (4.14), using Newton's method or a variant thereof, while maintaining strict feasibility for $s_{\mathcal{I}} \geq 0$ and $z_{\mathcal{I}} \geq 0$ (or $A_{\mathcal{I}} x \geq b_{\mathcal{I}}$ and $y_{\mathcal{I}}$ for (4.5)), and letting μ gradually decrease to zero. The most popular are based on the linesearch-based predictor–corrector algorithm of Mehrotra (1992), originally developed for LP.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

A typical *predictor–corrector* iteration for (4.17) involves the solution of a pair of (symmetrized) linear systems of the form

$$\begin{pmatrix} G & 0 & A_{\mathcal{E}}^T & A_{\mathcal{I}}^T & 0 \\ 0 & 0 & 0 & -I & -I \\ A_{\mathcal{E}} & 0 & 0 & 0 & 0 \\ A_{\mathcal{I}} & -I & 0 & 0 & 0 \\ 0 & -I & 0 & 0 & -Z_{\mathcal{I}}^{-1}S_{\mathcal{I}} \end{pmatrix} \begin{pmatrix} d_x \\ d_{s_{\mathcal{I}}} \\ -d_{y_{\mathcal{E}}} \\ -d_{y_{\mathcal{I}}} \\ d_{z_{\mathcal{I}}} \end{pmatrix} = - \begin{pmatrix} g(x) - A_{\mathcal{E}}^T y_{\mathcal{E}} - A_{\mathcal{I}}^T y_{\mathcal{I}} \\ y_{\mathcal{I}} - z_{\mathcal{I}} \\ c_{\mathcal{E}}(x) \\ c_{\mathcal{I}}(x) - s_{\mathcal{I}} \\ -Z_{\mathcal{I}}^{-1} r_{\mathrm{C}}(\mu) \end{pmatrix} \tag{4.18}$$

for different values of $r_{\mathrm{C}}(\mu)$, where (as before) $c_{\mathcal{I}}(x) = A_{\mathcal{I}}x - b_{\mathcal{I}}$, $c_{\mathcal{E}}(x) = A_{\mathcal{E}}x - b_{\mathcal{E}}$, $g(x) = g + Hx$ and $G \approx H$. Note that Newton's method for (4.17) results when $G = H$ and $r_{\mathrm{C}}(\mu) = S_{\mathcal{I}}z_{\mathcal{I}} - \mu e$ with the current value of μ. The first of the two systems uses $r_{\mathrm{C}}(\mu) = S_{\mathcal{I}}z_{\mathcal{I}}$ and defines a *predictor* step intended to reduce primal and dual feasibility. This step is often referred to as an *affine scaling* step and is denoted d^{AFF}. A steplength α^{AFF} is determined to preserve positivity of $z_{\mathcal{I}}$ and $s_{\mathcal{I}}$. On defining the duality gap after the predictor step $\mu^{\mathrm{AFF}} = (z_{\mathcal{I}} + \alpha^{\mathrm{AFF}} d_{z_{\mathcal{I}}}^{\mathrm{AFF}})^T (s_{\mathcal{I}} + \alpha^{\mathrm{AFF}} d_{s_{\mathcal{I}}}^{\mathrm{AFF}})/n_{\mathcal{I}}$ and the centering parameter $\sigma = (\mu^{\mathrm{AFF}}/\mu)^\tau$ with $2 \leq \tau \leq 4$, the second system uses $r_{\mathrm{C}}(\mu) = S_{\mathcal{I}}z_{\mathcal{I}} - \sigma\mu e + D_{s_{\mathcal{I}}}^{\mathrm{AFF}} d_{z_{\mathcal{I}}}^{\mathrm{AFF}}$ and defines a *corrector* step aiming to improve centrality. The final primal and dual common steplength (Mehrotra 1992) is determined by

$$\alpha = \min(1, \eta\alpha_{\mathrm{MAX}}^{\mathrm{P}}, \eta\alpha_{\mathrm{MAX}}^{\mathrm{D}}),$$

where $\eta \in [0.9, 1.0)$ converges to 1 as a solution is approached, and $\alpha_{\mathrm{MAX}}^{\mathrm{P}}$ and $\alpha_{\mathrm{MAX}}^{\mathrm{D}}$ are primal and dual steplengths enforcing a fraction-to-the-boundary rule.

The higher-order corrections scheme of Gondzio (1996), again originally developed for LP, generalizes to convex quadratic programming. Several corrector-like steps are taken, as long as substantial steplengths are acceptable and individual complementarity pairs cluster around their average value. These steps aim for dynamically computed targets (Jansen, Roos, Terlaky and Vial 1996) located in a loose neighbourhood of the central path. The number of corrector steps is computed at the first iteration by balancing the cost of the linear algebra and the expected progress towards optimality.

Just as in Section 4.1, (block) direct or iterative methods may be used to solve the indefinite system (4.18). Further savings often result from the block elimination

$$\begin{pmatrix} G & A_{\mathcal{E}}^T & A_{\mathcal{I}}^T \\ A_{\mathcal{E}} & 0 & 0 \\ A_{\mathcal{I}} & 0 & -Z_{\mathcal{I}}^{-1}S_{\mathcal{I}} \end{pmatrix} \begin{pmatrix} d_x \\ -y_{\mathcal{E}} - d_{y_{\mathcal{E}}} \\ -y_{\mathcal{I}} - d_{y_{\mathcal{I}}} \end{pmatrix} = - \begin{pmatrix} g(x) \\ c_{\mathcal{E}}(x) \\ c_{\mathcal{I}}(x) - s_{\mathcal{I}} - Z_{\mathcal{I}}^{-1} r_{\mathrm{C}}(\mu) \end{pmatrix} \tag{4.19}$$

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

of (4.18), or possibly even from

$$\begin{pmatrix} G + A_{\mathcal{I}}^T S_{\mathcal{I}}^{-1} Z_{\mathcal{I}} A_{\mathcal{I}} & A_{\mathcal{E}}^T \\ A_{\mathcal{E}} & 0 \end{pmatrix} \begin{pmatrix} d_x \\ -y_{\mathcal{E}} - d_{y_{\mathcal{E}}} \end{pmatrix} = - \begin{pmatrix} g(x) + A_{\mathcal{I}}^T S_{\mathcal{I}}^{-1} Z_{\mathcal{I}} [c_{\mathcal{I}}(x) - s_{\mathcal{I}} - Z_{\mathcal{I}}^{-1} r_{\mathrm{C}}(\mu)] \\ c_{\mathcal{E}}(x) \end{pmatrix}, \tag{4.20}$$

which arises by further eliminating $y_{\mathcal{I}} + d_{y_{\mathcal{I}}}$ from (4.19) – of course (4.20) may be inappropriate if the term $A_{\mathcal{I}}^T S_{\mathcal{I}}^{-1} Z_{\mathcal{I}} A_{\mathcal{I}}$ is significantly denser than G, but has the virtue of being considerably smaller if there are many inequality constraints. It is also worth noting that the corresponding predictor–corrector steps for (4.16) satisfy

$$\begin{pmatrix} G & A_{\mathcal{E}}^T & A_{\mathcal{I}}^T \\ A_{\mathcal{E}} & 0 & 0 \\ A_{\mathcal{I}} & 0 & -Y_{\mathcal{I}}^{-1} C_{\mathcal{I}} \end{pmatrix} \begin{pmatrix} d_x \\ -y_{\mathcal{E}} - d_{y_{\mathcal{E}}} \\ -y_{\mathcal{I}} - d_{y_{\mathcal{I}}} \end{pmatrix} = - \begin{pmatrix} g(x) \\ c_{\mathcal{E}}(x) \\ Y_{\mathcal{I}}^{-1} r_{\mathrm{C}}(\mu) \end{pmatrix} \tag{4.21}$$

which is simply (4.19) in the special case $s_{\mathcal{I}} = c_{\mathcal{I}}(x)$ and $z_{\mathcal{I}} = y_{\mathcal{I}}$ – also *cf.* (4.12). The coefficient matrices from (4.18), (4.19)/(4.21) and (4.20) are appropriate preconditioners for PCG as long as they have, respectively, $\mathrm{rank}(A_{\mathcal{E}}) + 2|\mathcal{I}|$, $\mathrm{rank}(A_{\mathcal{E}}) + |\mathcal{I}|$ and $\mathrm{rank}(A_{\mathcal{E}})$ negative eigenvalues (Conn *et al.* 2000*b*, with Sylvester's law of inertia); equivalently $G + A_{\mathcal{I}}^T S_{\mathcal{I}}^{-1} Z_{\mathcal{I}} A_{\mathcal{I}}$ should be positive definite on the null-space of $A_{\mathcal{E}}$, and this will always be the case if G is positive definite. Any of the factorizations mentioned in Section 4.1 are appropriate.

For large problems, it is vital to be able to exploit commonly occurring sub-structure when solving (4.19). Applications from multi-stage stochastic programming, network communications or asset liability management give rise to matrices H and A having one of a number of predefined block structure – examples include H and A being block diagonal, primal or dual block angular or bordered block diagonal. Moreover, this block structure appears recursively in the sense that the structure of the blocks is similar to that of the matrix containing them. This nestedness is fully exploitable if matrices H and A have compatible structures – *i.e.*, have the same number of diagonal blocks with matching numbers of columns – the coefficient matrix K of (4.19) can be reordered to have similarly exploitable block structure (Gondzio and Grothey 2003*a*).

Frequently in practice K may be very ill-conditioned or even singular, and it is common to *regularize* K to avoid such deficiencies. Typically, diagonal blocks R will be added to K so that the resulting matrix $K + R$ is quasi-definite. *Quasi-definite* matrices (Vanderbei 1995) are strongly factorizable in the sense that, for any symmetric permutation P, there exist a unit lower triangular matrix L and a diagonal matrix D such that $P(K + R)P^T = LDL^T$ without recourse to 2×2 pivoting, as is common with other

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

popular factorizations of indefinite matrices (see Section 4.1). If the system is block-structured as above, the quasi-definite factorization may easily be parallelized, since block structure in $P(K+R)P^T$ induces block structure in L and D (Gondzio and Grothey 2003*a*).

The requirement that slack variables introduced in (4.14) remain strictly feasible suggests that a steplength $0 < \alpha_k \leq \alpha_k^{\max}$ be chosen, where

$$\alpha_k^{\max} = \max\{0 < \alpha \leq 1 \mid s_{\mathcal{I},k} + \alpha d_{s_\mathcal{I}} \geq \gamma_k s_{\mathcal{I},k}\},$$

to enforce a fraction-to-the-boundary rule. A similar rule applies to primal variables that are subject to bounds and to dual variables. A strictly feasible initial point is any $s_{\mathcal{I},0} > 0$, but in practice it may be prudent to initialize $s_\mathcal{I}$ to a significantly positive value. Since the inequality constraints also need to be satisfied, a common choice is to pick $s_{\mathcal{I},0} = \max(A_\mathcal{I} x_0 - b_\mathcal{I}, \sigma e)$ componentwise, where x_0 is supplied by the user or the model, $\sigma > 0$ is a given constant, *e.g.*, $\sigma = 1$ and e is the vector of all ones. Often, explicit bound constraints will be honoured by first moving x_0 to satisfy them, and computing $s_{\mathcal{I},0}$ from this perturbed initial point. Another possibility is to compute an affine-scaling step d^{AFF}, *i.e.*, using $\mu = 0$, for the primal–dual system associated with (4.5). On defining $s_\mathcal{I}^{\text{AFF}} = s_{\mathcal{I},0} + d_{s_\mathcal{I}}^{\text{AFF}}$, an initial $s_{\mathcal{I},1}$ is computed based on the feasibility of $s_\mathcal{I}^{\text{AFF}}$, using a rule such as

$$s_{\mathcal{I},1} = \max(\beta e, |s_\mathcal{I}^{\text{AFF}}|) \quad \text{or} \quad s_{\mathcal{I},1} = s_\mathcal{I}^{\text{AFF}} + \gamma \max(0, -s_\mathcal{I}^{\text{AFF}}) + \beta e, \tag{4.22}$$

where the absolute values and maxima are understood elementwise and $\beta, \gamma > 0$ (Gertz, Nocedal and Sartenaer 2003).

Good general-purpose initial values for the Lagrange multipliers y in primal–dual interior methods are hard to find, and poor guesses may introduce unwarranted nonconvexity into the model if the problem is nonconvex. Nonetheless, they are often initialized to approximate least-squares solutions for dual feasibility, *i.e.*, values of y for which the gradient lies on the null-space of the constraints at the starting point, and adjusted to ensure that those corresponding to inequality constraints are strictly positive.

Not all path-following interior-point methods for convex QP are of the predictor–corrector type. The simplest alternative is to solve (4.18) or (4.19) for a pre-assigned μ but to ensure ensure strict feasibility by means of a fraction-to-the-boundary rule, in which the step-size α is chosen as

$$\alpha = \min\left\{1, (1-\epsilon) \max_{[d_s]_i < 0} \frac{s_i}{-[d_s]_i}, (1-\epsilon) \max_{[d_z]_i < 0} \frac{z_i}{-[d_z]_i}\right\},$$

for a small $\epsilon > 0$. A merit function such as

$$\phi(x, s_\mathcal{I}, y, z_\mathcal{I}) \equiv s_\mathcal{I}^T z_\mathcal{I} + \|\nabla \mathcal{L}(x, s_\mathcal{I}, y_\mathcal{E}, y_\mathcal{I}, z_\mathcal{I})\|_2, \tag{4.23}$$

where $\mathcal{L}(x, s_\mathcal{I}, y_\mathcal{E}, y_\mathcal{I}, z_\mathcal{I})$ is the Lagrangian associated with (4.14), is used to assess suitability of such a steplength. Such a fraction-to-the-boundary

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

condition may be implicitly ensured by Zhang's (1994) step-size rule, and in this case yields a global linear convergence rate and a polynomial algorithm.

One further interesting idea in both convex and non-convex cases is to solve (4.5) by a sequence of minimizations over the intersection of the interior of the feasible region with iteratively generated low-dimensional (typically 2- or 3-dimensional) subspaces. The advantage here is that the resulting subproblems are small, so that global optimization is possible. Clearly the choice of subspaces is crucial, and should include at least one 'descent' direction for whatever globalization mechanism is to be used, and others which are geared towards fast asymptotic convergence – solutions of (4.18) for different $r_C(\mu)$ may be used (Boggs, Domich, Rogers and Witzgall 1996).

Practicalities

The only comparison of the competing QP ideologies we are aware of is that of Gould and Toint (2002*b*). As perhaps one might expect, the interior-point approach seems generally to be preferable to the active set, especially for very large problems where the number of active set iterations can be enormous. For 'warm-start' problems, where a solution to a small perturbation of an existing already solved problem is required, there is some virtue in using the active set approach as it seems better able to use good estimates of the optimal active set. Whether this trend will continue is debatable, especially as current research for LP indicates promise for warm-started interior-point methods (Gondzio and Grothey 2003*b*, Yildirim and Wright 2002).

When carefully implemented, interior methods for QP scale almost perfectly with the number of variables, and rarely do they need more than, say, 30–35 iterations. Moreover, unlike active-set-type methods, the linear systems which arise at each iteration have identical block structure. Nonetheless, the solution of such systems may still be costly, and implementations must pay particular attention to exploiting structure – an example of a disastrous situation caused by the lack of exploitation of low-rank-corrector structure is given by Ferris and Munson (2000).

A final important idea is to simplify QPs before solution. Such 'presolve' methods have proved to be very effective for LP (Gondzio 1997), and similar gains are also possible for QP (Gould and Toint 2004*b*).

Software

Currently available active set non-convex QP codes include `VE09` (Gould 1991), `bqpd` (Fletcher 2000) and QPA (Gould and Toint 2002*a*). The PRESOLVE package (Gould and Toint 2004*b*) is, as its name suggests, intended for presolving QPs.

Highly efficient commercial interior-point-based software such as CPLEX 6.0 (1998), MOSEK (Andersen and Andersen 2000) and XPRESS-MP (Guéret, Prins and Seveaux 2002) is available for convex QP. These packages

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

implement path-following algorithms in a primal–dual setting, and are available for parallel machines as well as for personal computers. Significantly, they may be tested online on the NEOS Server for Optimization (Czyzyk, Mesnier and Moré 1998, Gropp and Moré 1997, Dolan 2001).

The object-oriented QP package OOQP (Gertz and Wright 2003) implements generalizations of both Mehrotra's (1992) predictor–corrector and Gondzio's (1996) higher-order correction methods. OOQP has the advantage of being customizable to various application domains, and has been tailored to solve problems arising from support vector machines and Huber regression. Similar features are implemented in COPLQP (Ye 1997).

Specialized structure is exploited automatically by the object-oriented parallel solver OOPS (Gondzio and Grothey 2003*a*). Currently OOPS has been able to solve nontrivial problems involving 52 million variables and 20 million constraints.

Although now a code for general nonlinear programming, a set of default parameter values for convex QP and a careful implementation of a tailored LDL^T factorization for the quasi-definite systems at the heart of the algorithm make LOQO (Vanderbei 1999) one of the most robust predictor–corrector, primal–dual path-following convex QP solvers. Much of this is due to the care with which the factorization is obtained. An LDL^T factorization of the regularized matrix $K + R$ from (4.19) is computed using a two-stage ordering scheme assigning priorities to pivots based on estimates of the fill-in in both AA^T and A^TA. Pivots corresponding to the current priority are treated using a minimum-degree ordering heuristic.

QPB from the GALAHAD library of Gould *et al.* (2003*a*) implements a primal–dual interior method for general QP – for non-convex problems, QPB is only capable of identifying a weak second-order critical point. The Phase-I relies on the package LSQP, itself a primal–dual infeasible method for convex separable QP (Zhang 1994) which is also part of GALAHAD. Numerical tests on a monoprocessor machine on small, $n + m \lesssim 10^4$, medium, $10^4 \lesssim n + m \lesssim 10^5$ and large-scale, $10^5 \lesssim n + m \lesssim 10^6$, problems illustrate how well the method scales with the dimension, and the superiority of interior-point approaches over active-set-type methods when a reliable estimate of the optimal working set is not available and when the number of variables and constraints are large (Conn *et al.* 2000*b*, Gould *et al.* 2003*a*, Gould and Toint 2002*b*).

4.3. General linearly constrained optimization

When the constraints are linear but the objective neither linear nor quadratic, most algorithms try to emulate the QP methods described above, by ensuring feasibility with respect to constraints and requiring a reduction in the objective function (or perhaps barrier function) at each iteration – if an

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

interior-point method is used, the iterates will remain interior to all inequality constraints. The only significant differences occur because the Hessian of the objective function changes at each iteration and must be periodically evaluated or estimated by some means. If the objective is close to linear, solutions (and intermediate iterates) often have a high proportion of active constraints ($|W| \approx n$) and some methods (Murtagh and Saunders 1982, Gill, Murray and Saunders 2002, Friedlander and Saunders 2005) exploit this by maintaining (dense) secant approximations of the reduced Hessian.

Interior-point methods for convex problems have received extensive attention since the existence of self-concordant barriers leads to polynomial algorithms (Nesterov and Nemirovskii 1994, Renegar 2001), and specialized methods have been devised for important applications. A good example is the minimization of a nonlinear but convex, (and preferably, but not necessarily) separable objective subject to linear equalities and bounds which arise in transportation planning, knowledge management or world-wide web traffic modelling (Saunders and Tomlin 1996). The problem is stated as

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ f(x)\ \text{ subject to }\ Ax = b\ \text{ and }\ l \le x \le u,$$

and is regularized and reformulated as

$$\underset{x,r}{\text{minimize}}\ f(x)+\tfrac{1}{2}\|D_1x\|^2+\tfrac{1}{2}\|r\|^2\ \text{ subject to }\ Ax+D_2r = b\ \text{ and }\ l \le x \le u,$$

for some diagonal positive definite regularization matrices D_1 and D_2. A primal–dual path-following method is then applied. The bulk of the computation involves solving systems of the form

$$\begin{pmatrix} H & A^T \\ A & -D_2^2 \end{pmatrix} \begin{pmatrix} d_x \\ -d_y \end{pmatrix} = -\begin{pmatrix} \nabla_x f(x) - A^T y - \mu[(X-L)^{-1} + (U-X)^{-1}]e \\ Ax + D_2^2 y - b \end{pmatrix}$$

where $H = \nabla_{xx} f(x) + D_1^2 + (X-L)^{-1}Z_l + (U-X)^{-1}Z_u$ and y, z_l and z_u are suitable Lagrange multiplier estimates. As the coefficient matrix here is quasi-definite, it admits an LDL^T factorization. Alternatively, eliminating d_y to obtain normal equations and treating them as a least-squares problem, a trial step (d_x, d_y) is computed using a least-squares method, *e.g.*, LSQR of Paige and Saunders (1982).

Not all proposed interior-point methods are of the path-following variety. For example, it is possible to generalize the affine-scaling approach of Coleman and Li (1996) to handle linear inequality constraints (Coleman and Li 2000).

Software

Although it is capable of handling general constraints, the venerable active set NLP solver `MINOS` (Murtagh and Saunders 1982) is perhaps best regarded for its ability to deal with linear constraints. Likewise its successors `SNOPT`

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

(Gill *et al.* 2002) and KNOSSOS (Friedlander and Saunders 2005) are both highly effective for such problems, particularly if there are relatively few degrees of freedom. As usual, other general nonlinear programming packages, such as LOQO and KNITRO may be applied and are comfortable with such problems, although we would not recommend LANCELOT in this case.

5. Large-scale nonlinearly constrained optimization

Finally, we turn our attention to our most general nonlinear programming problem (1.1) and the attendant difficulties of coping with constraint curvature.

5.1. Sequential linear and quadratic programming methods

The phrase 'sequential quadratic programming' (SQP) seems to mean different things to different people, but the central theme is undoubtedly to apply an iteration for which a new iterate is generated by trying to minimize a quadratic approximation of the appropriate Lagrangian function $\ell(x,y) \stackrel{\text{def}}{=} f(x) - y_{\mathcal{E}}^T c_{\mathcal{E}}(x) - y_{\mathcal{I}}^T c_{\mathcal{I}}(x)$ subject to linearizations of some or all of the constraints. Here we will examine several aspects of this approach. There has been a number of surveys of SQP methods over the past 10 years (Boggs and Tolle 1995, 2000, Conn, Gould and Toint 1997, Gould and Toint 2000*b*) and we urge readers to consult these for details since we do not have room to give them all here.

We start by considering problems only involving equality constraints – for some people, such as those who work on PDE-constrained optimization (*e.g.*, Biros and Ghattas (2000)), this *is* SQP – for which the central ideas are best understood. But it is in the context of the general problem (1.1) that we believe most people understand the term SQP, and which we consider next. There is a strong distinction between linearizing a subset of the constraints at each iteration – the EQP subproblem approach, which is strongly influenced by methods for equality constraints – and linearizing all constraints at every iteration – the IQP subproblem approach.

5.2. SQP methods for equality-constrained problems

We first consider SQP methods for the equality-constrained (EC) problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ f(x)\ \text{ subject to }\ c_{\mathcal{E}}(x) = 0. \tag{5.1}$$

SQP methods for EC problems (SQPE) aim to find a correction d_k to the current solution estimate x_k so as to (approximately)

$$\begin{aligned} &\underset{d}{\text{minimize}}\ q_k(d) = d^T g(x_k) + \tfrac{1}{2} d^T H_k d \\ &\text{subject to }\ c_{\mathcal{E}}(x_k) + J_{\mathcal{E}}(x_k) d = 0. \end{aligned} \tag{5.2}$$

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

Here $g(x) = \nabla_x f(x)$ is the gradient of the objective, $J(x) = \nabla_x c_{\mathcal{E}}(x)$ is the Jacobian of the constraints, and H_k is (an approximation to) the Hessian of the Lagrangian function $\ell_{\mathcal{E}}(x, y_{\mathcal{E}}) = f(x) - y_{\mathcal{E}}^T c_{\mathcal{E}}(x)$ for given estimates $y_{\mathcal{E}k}$ of the Lagrange multipliers $y_{\mathcal{E}}$ at x_k. If $H_k = \nabla_{xx}\ell_{\mathcal{E}}(x_k, y_{\mathcal{E}k})$ and

$$d^T H_k d > 0 \;\text{ for all }\; d \;\text{ for which }\; J_{\mathcal{E}}(x_k)d = 0, \tag{5.3}$$

the solution to (5.2) is identical to that obtained by applying Newton's method to the criticality conditions $\nabla_{(x,y_{\mathcal{E}})}\ell_{\mathcal{E}}(x, y_{\mathcal{E}}) = 0$ at $(x_k, y_{\mathcal{E}k})$. Aside from the fundamental issues of how to choose H_k and $y_{\mathcal{E}k}$, SQPE methods have a number of obvious possible shortcomings. In particular (i) the linearized constraints may be inconsistent, (ii) (5.3) may be violated, and (iii) the iteration may diverge.

Possible shortcoming (i) is best dealt with in one of two, related, ways. The first is to re-pose (1.1) as the related penalty problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\;\; \phi(x,\sigma) \stackrel{\text{def}}{=} f(x) + \sigma \sum_{i\in\mathcal{E}} |c_i(x)| + \sigma \sum_{i\in\mathcal{I}} \min(-c_i(x), 0) \tag{5.4}$$

for some sufficiently large $\sigma > 0$ and given norm $\|\cdot\|$, and instead to minimize some model of the (non-smooth) penalty function $\phi(x, \sigma)$. A typical model problem then might be to approximately

$$\underset{d}{\text{minimize}}\;\; q_k(d) + \sigma\|c_{\mathcal{E}}(x_k) + J_{\mathcal{E}}(x_k)d\|; \tag{5.5}$$

if $\|\cdot\|$ is polyhedral (*e.g.*, the ℓ_1- or ℓ_∞-norm), (5.5) may be reformulated as a (consistent) inequality-constrained QP, while if $\|\cdot\|$ is elliptical (*e.g.*, the ℓ_2-norm) a quadratic conic-programming reformulation is possible.

Notice that the intention here is implicitly to allow inconsistent linearized constraints by merely reducing their infeasibility as much as is possible. A second, more direct way of dealing with inconsistency is to aim for reduction in infeasibility rather than full satisfaction of the constraints. A composite step $d_k = n_k + t_k$ may be used to achieve this. The idea is simply that the '(quasi-)normal' step n_k tries to reduce $\|c_{\mathcal{E}}(x) + J_{\mathcal{E}}(x_k)n\|$ while the 'tangential' step t_k aims to reduce $q_k(d)$ while maintaining the infeasibility at the level achieved by n_k; if n_k reduces the infeasibility to zero and t_k solves

$$\underset{t\in\mathbb{R}^n}{\text{minimize}}\;\; q_k(n_k + t) \;\text{ subject to }\; J_{\mathcal{E}}(x_k)t = 0, \tag{5.6}$$

d_k will be the solution to (5.2). Although there is a number of composite-step methods (Conn *et al.* 2000*a*, §15.4), the most appealing is the so-called Byrd–Omojokun approach (Byrd, Hribar and Nocedal 1999, Lalee, Nocedal and Plantenga 1998, Omojokun 1989), in which the CG method is used both to reduce $\|c_{\mathcal{E}}(x_k) + J_{\mathcal{E}}(x_k)n\|_2^2$ and subsequently to approximately solve the EQP (5.6) (see Section 4.1).

If shortcoming (ii) occurs, $q_k(d)$ will be unbounded from below on the

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

feasible region. Suitable remedies are just as in the unconstrained case (see Section 2). Linesearch-based methods cope with such an eventuality either by obtaining a direction of feasible negative curvature or by modifying H_k, although good methods for achieving the latter during matrix factorization are still in their infancy (Forsgren 2002, Forsgren and Murray 1993, Gould 1999). Trust region-based methods impose a constraint to stop steps to infinity, but there is the added complication that the trust region constraint $\|s\| \leq \Delta_k$ may be incompatible with the linearizations $J_{\mathcal{E}}(x_k)d = -c_{\mathcal{E}}(x_k)$ if Δ_k is too small. In this case, one of the remedies proposed for shortcoming (i) may be required.

Shortcoming (iii) may be overcome in the usual way, namely by requiring descent (monotonic or otherwise) with respect to a suitable merit function such as (5.4). A good choice of σ is vital if such a method is to be efficient, and we will return to this later. An unfortunate consequence – the Maratos (1978) 'effect' – is that the SQP step may not be acceptable to merit functions like (5.4), and that an auxiliary calculation (a 'second-order correction') may be required to modify the step to allow fast convergence. Other merit functions, such as the augmented Lagrangian function, avoid this defect and have been used with much success (Boggs, Kearsley and Tolle 1999*a*, Gill *et al.* 2002).

A modern alternative to merit functions, which avoids the need to compute a penalty parameter, is to use the filter idea introduced in Section 2.1. For EC problems, we consider the conflicting objectives $\theta_1(x)$ and $\theta_2(x)$ to be the objective function and the constraints violation $\|c_{\mathcal{E}}(x)\|$, respectively. A step d is thus accepted if either the objective function decreases or if the constraints violation is reduced, while it is rejected if no decrease is obtained in either. But of course many further refinements are necessary in order to devise a workable algorithm. One is the way that filter methods deal with incompatible model constraints. Rather than resorting to the remedies for shortcoming (i) given above, filter trust region algorithms switch to a 'restoration phase', *i.e.*, to the minimization of the constraint violation alone (the objective function is momentarily forgotten) until a model with compatible constraints is found. Since this will be true for any feasible point for the original problem, this restoration phase must terminate at a suitable point as long as it is capable of finding one – or indeed if it is even possible to find one at all. This restoration phase may use any suitable algorithm, including the filter method for nonlinear least-squares mentioned in Section 2.2. It may also be triggered more frequently – the method of Gonzaga, Karas and Vanti (2003) performs the equivalent of a restoration phase at every iteration.

A drawback that is common to SQPE approaches is that they all potentially suffer from the Maratos effect and therefore may need a second-order correction step to guarantee fast convergence. In theory this may be avoided

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

by the filter remembering Lagrangian rather than objective function values (Ulbrich 2004*b*), but, to our knowledge, numerical experience is not yet available to support this idea in practice.

Rival trust region SQPE filter methods impose different requirements on the step computation – Fletcher, Leyffer and Toint (2002*b*) require the global solution of the trust region constrained SQPE, while others (Fletcher and Leyffer 2002, Fletcher, Gould, Leyffer, Toint and Wächter 2002*a*, Gonzaga *et al.* 2003, Gould and Toint 2005, 2003*b*) permit approximate local minimizers – and on the precise technique for maintaining the filter. This technical decision is often based on the distinction between iterations whose main effect is to reduce the objective function (f-iterations), and iterations whose main effect is to reduce constraint violation (θ-iterations).

Linesearch variants of the filter idea are also possible. Despite using a different globalization technique, the proposal of Wächter and Biegler (2003*a*, 2003*b*) remains similar in structure to the trust region variants, in that it also involves restorations, second-order correction steps and similarly uses the distinction between f- and θ-iterations to manage the filter.

For all SQPE algorithms, two other issues which are of great practical importance are the choice of Hessian approximation H_k and Lagrange multiplier estimates $y_{\mathcal{E}k}$. Although exact second derivatives of the Hessian of the Lagrangian are often available, the use of approximations still persists especially for problems where $|\mathcal{E}| \approx n$. In particular, as we noted in Section 4, solving (5.6) may be reduced to minimizing $t_n^T g_n + \frac{1}{2} t_n^T H_n t_n$ and recovering $t_k = N^T t_n$, where $g_n = N^T(g(x_k) + H_k n_k)$ and $H_n = N^T H_k N$, and the columns of N form a basis for the null-space of $J_{\mathcal{E}}(x_k)$. Thus, as long as $|\mathcal{E}| \approx n$, H_n will be small and it will be feasible to maintain H_n as a dense secant approximation to $N^T \nabla_{xx} \ell_{\mathcal{E}}(x_k, y_{\mathcal{E}k}) N$ (see the survey articles mentioned at the start of this section). If $|\mathcal{E}| \not\approx n$, it may still be possible to maintain a useful limited-memory secant approximation to the same matrix (Gill *et al.* 2002). Lagrange multipliers $y_{\mathcal{E}\,k+1}$ are often taken as those from the approximate solution to (5.4), although some form of interpolation between these values and $y_{\mathcal{E}k}$ may be necessary if the merit function, the trust region or constraint inconsistency intervene; little work seems to have been performed to discover the influence of such distractions which is somewhat surprising given the influence $y_{\mathcal{E}k}$ may have on H_k. As an alternative, a direct or CG least-squares solution to $J_{\mathcal{E}}^T(x_k) y = g(x_k)$ may be appropriate (Lalee *et al.* 1998).

5.3. SQP methods for the general problem

Suffice it to say, as the name suggests, an SQP method aims to solve the general problem (1.1) by solving a sequence of (cleverly) chosen QP problems. There are essentially two classes of SQP methods.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

Sequential equality-constrained quadratic programming (SEQP) methods The first, which we call sequential equality-constrained QP (SEQP) methods are essentially SQPE methods for which the set $\mathcal{E}$ is replaced by a (changing) estimate $\mathcal{A}_k \subseteq \mathcal{E} \cup \mathcal{I}$ of (1.1)'s optimal active set. All of the salient points we made about SQPE methods apply equally here, but now the dynamic data structures necessary to accommodate changes in $\mathcal{A}_k$ and, more importantly, the choice of $\mathcal{A}_k$ itself introduce extra complications. Of paramount importance is the globalization strategy, since otherwise there will be little control over constraints not in $\mathcal{A}_k$. In particular, it is vital that all constraints are represented in whatever merit function or filter is used.

A common strategy is to use the non-differentiable penalty function

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ \phi(x,\sigma) \stackrel{\text{def}}{=} f(x) + \sigma\|c_{\mathcal{E}}(x)\| + \sigma\|\min(-c_{\mathcal{I}}(x),0)\| \tag{5.7}$$

as a merit function, and to use an EQP model in which a second-order approximation to the (locally) differentiable part of $\phi(x,\sigma)$ is minimized subject to linearized approximations to the (locally) non-differentiable part remaining fixed (Coleman and Conn 1982); $\mathcal{A}_k$ is thus defined by those constraints with (almost) zero values.

An alternative is to use the active set at a minimizer of a 'simpler' model of (1.1) or (5.7) to predict the active set of (1.1). The most obvious models are linear, and lead to linear programming subproblems which aim to

$$\begin{aligned}\underset{d}{\text{minimize}}\ l_k(d) = d^T g(x_k)\ \text{ subject to }\ & c_{\mathcal{E}}(x_k) + J_{\mathcal{E}}(x_k)d = 0 \\ \text{and }\ & c_{\mathcal{I}}(x_k) + J_{\mathcal{I}}(x_k)d \geq 0.\end{aligned} \tag{5.8}$$

or

$$\underset{d}{\text{minimize}}\ l_k(d) + \sigma\|c_{\mathcal{E}}(x_k) + J_{\mathcal{E}}(x_k)d\| + \sigma\|\min(-c_{\mathcal{I}}(x_k) - J_{\mathcal{I}}(x_k)d,0)\|; \tag{5.9}$$

the advantage here is that there are excellent (simplex and interior-point) methods for large-scale linear programming. However, since the solutions to these subproblems almost inevitably lie at vertices of their feasible regions, and as there is no reason to expect that the solution to (1.1) has n active constraints, (5.8) or (5.9) alone are not sufficient to determine $\mathcal{A}_k$.

One way of remedying this is to impose artificial constraints whose role is simply to cut off those problem constraints which are likely to be inactive at the solution to (1.1); if an artificial constraint is active at the solution of (5.8) or (5.9) it will not be included in $\mathcal{A}_k$. Care must be taken, however, to ensure that the artificial constraints do not exclude optimally active problem constraints, and the balance between these aims is quite delicate. Early sequential linear programming (SLP) methods (Griffith and Stewart 1961) imposed artificial constraints of the form $\|s\|_\infty \leq \Delta$ in which Δ was dynamically adjusted, but it was Fletcher and Sainz de la Maza

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

(1989) who first interpreted this as a trust region constraint. Crucially they showed that the usual trust region acceptance and adjustment rules are sufficient to correctly identify the optimal active set in a finite number of iterations. Both filter-based (Chin and Fletcher 2003) and merit-function-based (Fletcher and Sainz de la Maza 1989, Byrd *et al.* 2004*a*) SLP variants are possible.

If the non-differentiable penalty function $\phi(x, \sigma)$ in (5.7) is used, it is important that the penalty parameter σ be adjusted to ensure that ultimately feasible critical points of the latter correspond to critical points of (1.1). Although in principle one could simply adjust σ once an approximate solution of (5.7) has been found (Mayne and Polak 1976), this is wasteful. It is preferable to adjust σ as soon as there is model-based evidence that the current value is not reducing the constraints, and means for doing this while ensuring convergence to critical points of (1.1) (or perhaps finding a critical point of infeasibility) are known (Byrd, Gould, Nocedal and Waltz 2004*b*).

Sequential inequality-constrained quadratic programming (SIQP) methods

The second class of SQP methods are those we refer to as sequential inequality-constrained QP (SIQP) methods. In these, no *a priori* prediction is made about the active set, but instead a correction d_k is chosen to (approximately)

$$\begin{aligned} \underset{d}{\text{minimize}} \; q_k(d) \; \text{subject to} \; & c_{\mathcal{E}}(x_k) + J_{\mathcal{E}}(x_k)d = 0 \\ & \text{and} \; c_{\mathcal{I}}(x_k) + J_{\mathcal{I}}(x_k)d \geq 0. \end{aligned} \tag{5.10}$$

Now H_k is an approximation to the Hessian of the full Lagrangian, $\ell(x, y) = f(x) - y^T c(x)$, for the problem, and linearizations of all constraints are included. Although we no longer need to specify $\mathcal{A}_k$, constraint inconsistency and iterate divergence are still serious concerns, and now we have the added complication that (iv) (5.10) may have (many) local minimizers.

Given all of these potential pitfalls, why are SIQP methods so popular? One reason is obviously their potential for fast local convergence; under reasonable assumption, the iteration based on (5.10) will correctly identify the active set and thereafter converge rapidly (Robinson 1974). Another is favourable empirical evidence accumulated on small-scale problems (Hock and Schittkowski 1981). But, on this basis and given the growing number of successful codes for large-scale QP, it might be thought surprising that there are so few large-scale SIQP algorithms. We now believe that this is not a coincidence and most likely an indication of the unsuitability of the SIQP paradigm for large-scale optimization. Why do we believe this?

Our first objection to SIQP is simply that, given even the most efficient QP method, the cost of solving a large-scale inequality-constrained

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

QP (IQP) is usually far greater than, say, an equivalently sized EQP or interior-point subproblem. Thus a method that uses IQPs either needs to ensure that relatively few overall iterations are required, or have some mechanism for stopping short of QP optimality. Although there is anecdotal evidence that SIQP methods require few iterations for small-scale problems, we are unaware of any proof that this will always be the case. Likewise, the methods suggested in the tiny body of work on IQP truncation (Goldsmith 1999, Murray and Prieto 1995) may, in the worst case, require the solution of n (related) EQPs per IQP.

Of more serious concern are the dangers posed by allowing indefinite H_k. This possibility rarely surfaced in the small-scale case, since almost always positive definite secant Hessian approximations were used. But for large problems traditional secant approximations are rarely viable on sparsity grounds – limited-memory secant methods are possible (Gill *et al.* 2002) but may give inaccurate approximations, while the alternatives of using partitioned secant approximations or exact second derivatives often generate indefinite Hessians. Indefinite H_k may cause difficulties for a number of reasons. Firstly, the possibility of moving to an unwelcome (possibly higher) local minimizer, d_k, cannot be discounted, particularly when using an interior-point QP solver. Such a d_k may well be unsuitable for use with a globalization strategy. While this appears to be a defect specifically for interior-point QP solvers, active set methods may also be fooled. Consider the simple-bound QP

$$\underset{x\in\mathbb{R}^2}{\text{minimize}}\ \tfrac{1}{2}(x_1^2+x_2^2)-3x_1x_2-\tfrac{5}{4}x_1+\tfrac{7}{4}x_2\ \text{ subject to }\ 0\le x_1,x_2\le 1,$$

whose contours are illustrated in Figure 5.1 – this is a simplified version of that given by Goldsmith (1999). Starting from $x=(0,0)$, many active set QP solvers would move downhill via the corner $(1,0)$ to the (global) minimizer at $(1,1)$ – both steps are along directions of positive curvature. Unfortunately the overall step $(1,1)$ is an initially uphill direction of negative curvature, and thus again unlikely to be suitable for use with a globalization strategy. Of course this does not mean that the method will fail, merely that the approach may be inefficient as extra precautions (such as reducing a trust region radius or modifying curvature) may have to be applied.

Whatever our reservations, SIQP methods remain popular. Linesearch, trust region and filter variants have been proposed. Some avoid difficulty (iv) above by insisting on positive definite (sometimes limited-memory) secant approximations to second derivatives (Gill *et al.* 2002). Others modify true second derivatives to ensure that the reduced Hessian is positive definite (Boggs, Kearsley and Tolle 1999*b*, Boggs *et al.* 1999*a*), while some use the restoration-phase of the filter approach to recover from bad steps (Fletcher and Leyffer 2002).

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

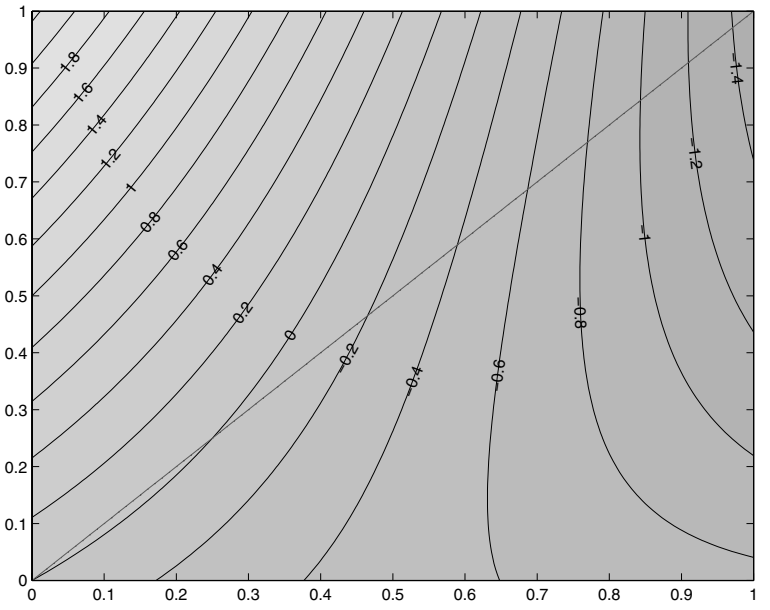


Figure 5.1. Active set QP method gives uphill step

5.4. Interior-point methods for nonlinear programs

As the reader might anticipate, the last decade has seen an explosion in interest in path-following methods for the general nonlinear program (1.1). Amongst the large number of papers devoted to the topic, two related approaches have emerged.

The first places all inequality constraints directly into a logarithmic barrier, leaving only explicit equality constraints. A sequence of barrier subproblems of the form

$$\begin{aligned} &\underset{x\in\mathbb{R}^n}{\text{minimize}} \ \phi(x;\mu) \stackrel{\text{def}}{=} f(x) - \mu\sum_{i\in\mathcal{I}} \log(c_i(x)) \\ &\text{subject to } \ c_{\mathcal{E}}(x) = 0, \end{aligned} \tag{5.11}$$

parametrized by the scalar $\mu > 0$, is solved for positive values of μ which eventually decrease to zero. This approach is particularly appealing when $c_{\mathcal{E}}(x) = 0$ are sufficiently simple to be handled directly, *e.g.*, when they are linear (Conn *et al.* 2000*b*) – indeed, this is simply a generalization of (4.7) – but does require that the inequality constraints are strictly satisfied throughout. Since this may be difficult to achieve – even finding an initial point for which this is true may be far from trivial – the second approach allows inequality constraints to be violated at intermediate stages, but for each introduces a slack variable which is treated by a barrier function. The

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

resulting problem is thus of the form

$$\begin{aligned} \underset{x\in\mathbb{R}^n,\ s_{\mathcal{I}}\in\mathbb{R}^{n_{\mathcal{I}}}}{\text{minimize}}\ \phi(x,s;\mu) &\stackrel{\text{def}}{=} f(x) - \mu\sum_{i\in\mathcal{I}}\log(s_i) \\ \text{subject to}\ \ c_{\mathcal{E}}(x) = 0\ \ &\text{and}\ \ c_{\mathcal{I}}(x) - s_{\mathcal{I}} = 0. \end{aligned} \tag{5.12}$$

Clearly, the introduction of slacks $s_{\mathcal{I}}$ is reminiscent of (4.15). Although it is vital that the slacks remain strictly feasible throughout, not all methods of this type remain infeasible right up to the solution (Byrd, Nocedal and Waltz 2003).

For both approaches, the barrier subproblems are equality-constrained, and the SQPE methods described in Section 5.2 are appropriate (Byrd, Gilbert and Nocedal 2000, Vanderbei and Shanno 1999, Wächter and Biegler 2004). Note, however, that extra precautions to ensure that the barrier terms remain finite must be taken, and it is here that (5.12) has some advantage, since in this case the barrier terms only involve (trivial) linear expressions.

Just as in the linearly constrained case, locally convergent methods may be devised by applying (variants of) Newton's method to the perturbed optimality conditions

$$\begin{aligned} \nabla f(x) - J_{\mathcal{E}}(x)^T y_{\mathcal{E}} - J_{\mathcal{I}}(x)^T y_{\mathcal{I}} &= 0, \\ c_{\mathcal{E}}(x) &= 0, \\ \text{and}\ \ c_{\mathcal{I}}(x)y_{\mathcal{I}} - \mu e &= 0, \end{aligned} \tag{5.13}$$

of (1.1), or

$$\begin{aligned} \nabla f(x) - J_{\mathcal{E}}(x)^T y_{\mathcal{E}} - J_{\mathcal{I}}(x)^T y_{\mathcal{I}} &= 0, \\ y_{\mathcal{I}} - z_{\mathcal{I}} &= 0, \\ c_{\mathcal{E}}(x) &= 0, \\ c_{\mathcal{I}}(x) - s_{\mathcal{I}} &= 0, \\ \text{and}\ \ Z_{\mathcal{I}}s_{\mathcal{I}} - \mu e &= 0, \end{aligned} \tag{5.14}$$

of

$$\underset{x\in\mathbb{R}^n,\ s_{\mathcal{I}}\in\mathbb{R}^{n_{\mathcal{I}}}}{\text{minimize}}\ f(x)\ \ \text{subject to}\ \ c_{\mathcal{E}}(x) = 0,\ c_{\mathcal{I}}(x) - s_{\mathcal{I}} = 0\ \ \text{and}\ \ s_{\mathcal{I}} \geq 0, \tag{5.15}$$

(*cf.* (4.16) and (4.17)). The only differences between the variants on Newton's method described in Section 4.2 and those applicable here are that the Jacobians $A_{\mathcal{E}}$ and $A_{\mathcal{I}}$ in (4.18)–(4.21) are now $J_{\mathcal{E}}(x)$ and $J_{\mathcal{I}}(x)$, respectively, and that G should now be an approximation to the Hessian of the Lagrangian, $\nabla_{xx}\ell(x,y)$; (direct or iterative) methods for solving these systems are identical to those in Section 4.2.

In (5.12), when the constraint Jacobian

$$J(x) = \begin{pmatrix} J_{\mathcal{E}}(x) & 0 \\ J_{\mathcal{I}}(x) & I \end{pmatrix}$$

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

has full rank, composite step (reduced space) variants of (4.18)–(4.21) are also possible. Just as in Section 5.2, the step may be decomposed using the Byrd–Omojokun scheme (Omojokun 1989), and least-squares estimates of the Lagrange multipliers for the equality constraints obtained. If similar multiplier estimates for inequality constraints are found, care needs to be taken to ensure that these remain positive (Wright 1997) or that the quadratic model remains convex in the slacks (Byrd *et al.* 1999). For problems arising from, *e.g.*, dynamical systems where multiplier estimates are not available, it is remarkable that schemes to update the penalty parameter may still be derived (Wächter 2002).

As always, it is necessary to globalize Newton's method in some way, and both (smooth and non-smooth) merit function- and filter-based possibilities have been proposed. Issues that arise with the linesearch globalization of the Newton direction d_x

$$\begin{pmatrix} G & J_{\mathcal{E}}^T(x) & J_{\mathcal{I}}^T(x) \\ J_{\mathcal{E}}(x) & 0 & 0 \\ J_{\mathcal{I}}(x) & 0 & -Z_{\mathcal{I}}^{-1}S_{\mathcal{I}} \end{pmatrix} \begin{pmatrix} d_x \\ -y_{\mathcal{E}} - d_{y_{\mathcal{E}}} \\ -y_{\mathcal{I}} - d_{y_{\mathcal{I}}} \end{pmatrix} = - \begin{pmatrix} g(x) \\ c_{\mathcal{E}}(x) \\ c_{\mathcal{I}}(x) - s_{\mathcal{I}} + \mu Z_{\mathcal{I}}^{-1}e \end{pmatrix} \tag{5.16}$$

(or its Section 4.2 equivalents (4.18)–(4.21)) include the choices of step-size and other (penalty and barrier) parameters and strategies to ensure that G is chosen to guarantee that d_x gives descent for whatever merit function is used – to date, the simple expedient of adding a diagonal matrix λI to G for suitably large λ seems to be the most sophisticated strategy used in the large-scale case (Vanderbei and Shanno 1999, Wächter and Biegler 2004), although, just as in Section 4.2, all that is actually required is that $G + J_{\mathcal{I}}^T(x)S_{\mathcal{I}}^{-1}Z_{\mathcal{I}}J_{\mathcal{I}}(x)$ should be positive definite on the null-space of $J_{\mathcal{E}}(x)$. Murray and Wright (1992) devised a linesearch procedure tailored to the logarithmic barrier function, given a search direction d, by identifying the closest constraint for which d is a descent direction at the current iterate. A step-size is computed by identifying a root of an approximation to the gradient of the barrier along d, by linearizing f and the constraint in question and ignoring all other constraints. Several other interpolating functions are used as approximations of the logarithmic barrier by the same authors, so as to devise specialized linesearches. To the best of our knowledge, these have not been incorporated into large-scale interior-point codes.

Typical merit functions for (5.12) might be the non-smooth penalty-barrier function (Yamashita, Yabe and Tanabe 2004)

$$\phi(x, s; \mu, \nu) = f(x) - \mu \sum_{i \in \mathcal{I}} \log(s_i) + \nu \|c_{\mathcal{E}}(x)\| + \nu \|c_{\mathcal{I}}(x) - s_{\mathcal{I}}\|, \tag{5.17}$$

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

the smooth variant (Gay, Overton and Wright 1998, Vanderbei and Shanno 1999)

$$\psi(x, s; \mu, \nu) = f(x) - \mu \sum_{i \in \mathcal{I}} \log(s_i) + \tfrac{\nu}{2} \|c_{\mathcal{E}}(x)\|_2^2 + \tfrac{\nu}{2} \|c_{\mathcal{I}}(x) - s_{\mathcal{I}}\|_2^2, \quad (5.18)$$

or some scaled equivalents, perhaps even involving a different penalty parameter ν_i per constraint to account for poor scaling. Although these parameters should be handled globally as described in Section 5.3, care must be taken to ensure that the direction computed from (5.16) or its variants is a descent direction for the merit function. This may be guaranteed for ϕ by iteratively increasing the penalty parameter until its directional derivative is negative; only a finite number of increases are required under standard assumptions (Byrd *et al.* 2000).

A disadvantage of (5.17) and (5.18) is that they really only measure suitability of the primal step d_x; other means are used to compute steps in the dual variables. One function which does not suffer from this drawback is the augmented penalty-barrier merit function (Forsgren and Gill 1998),

$$\begin{aligned} \theta(x, y; \mu, \nu) = f(x) &+ \frac{1}{2\mu} \sum_{i \in \mathcal{E}} \left\{ c_i(x)^2 + \nu (c_i(x) + \mu y_i)^2 \right\} \\ &- \mu \sum_{i \in \mathcal{I}} \left\{ \log(c_i(x)) + \nu \left(\log\left(\frac{c_i(x) y_i}{\mu} \right) + 1 - \frac{c_i(x) y_i}{\mu} \right) \right\}, \end{aligned} \quad (5.19)$$

which allows simultaneous minimization in both the primal and dual variables. So long as $G + J_{\mathcal{I}}^T(x) S_{\mathcal{I}}^{-1} Z_{\mathcal{I}} J_{\mathcal{I}}(x)$ is positive definite on the null-space of $J_{\mathcal{E}}(x)$, the primal–dual Newton step (5.16) for (5.13) is a descent direction for θ. If not, negative curvature descent directions are easy to obtain.

For the most part, theoretical analyses of these techniques make relatively strong assumptions – a linear independence qualification condition (LICQ) is often required to establish global convergence, while fast local convergence analyses rely on strict complementarity. Because the objective function and the barrier objective function both decrease monotonically with μ along the exact central path (Fiacco and McCormick 1968, Wright 1992), path-following algorithms for nonlinear programming have a monotone flavour. This is at variance with other methods discussed earlier.

A most disturbing aspect of linesearch-based interior-point methods which use (5.13) to compute the search direction d_x has recently been discovered (Wächter and Biegler 2000). The issue is that if there is a mixture of equality and inequality constraints, if the former are approximated by linearizations and if feasibility of the latter are controlled by restricting the step along the search direction, the resulting iteration may converge to a worthless infeasible point. This surprising result has caused a reassessment of linesearch methods, and in some cases consideration of filter methods

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

with appropriate acceptance measures as a replacement (Benson, Vanderbei and Shanno 2002, Wächter and Biegler 2004). The natural alternative, though, is to consider trust region-based methods, which fortunately do not suffer from this convergence failure.

In trust region interior methods for general nonlinear programming, any of the SQPE approaches discussed in Section 5.2 are appropriate, but now extra care needs to be taken to cope with the required feasibility of the inequality constraints in (5.11) or slacks in (5.12). In particular, in the latter case, it is important that the slack variables do not approach their bounds either prematurely or too rapidly. The obvious SQPE trust region subproblem would minimize a quadratic approximation to the Lagrangian of (5.12) – in which as usual a primal–dual approximation $Z_{\mathcal{I}}S_{\mathcal{I}}^{-1}$ to the Hessian of the barrier terms rather than the primal one $\mu S_{\mathcal{I}}^{-2}$ is used – subject to linearized approximations to the constraints within an appropriately scaled trust region and perhaps a suitable fraction-to-the-boundary constraint. For example, the step (d_x, d_s) may be constrained so that

$$\|(d_x, S^{-1}d_s)\|_2 \leq \Delta \quad \text{and} \quad s + d_s \geq (1-\tau)s,$$

with $0 < \tau \lesssim 1$ (Byrd, Gilbert and Nocedal 2000, Byrd, Hribar and Nocedal 1999), or the fraction-to-the-boundary rule may be imposed after the event (Conn *et al.* 2000*b*). In general, it is especially important that the shape of the trust region mirrors that of the ill-conditioned barrier terms (Conn *et al.* 2000*b*). As before, the issue of linearized constraint incompatibility – particularly when there is a trust region – is present, and a composite-step strategy as outlined in Section 5.3 is appropriate. As in the linesearch case, the penalty parameter ν must be adjusted as the algorithm proceeds to try to ensure asymptotic feasibility of the constraints, and rules to achieve this within a trust region framework are known (Byrd *et al.* 2000). It is also possible to use the augmented penalty-barrier merit function (5.19) within such a framework (Gertz and Gill 2004).

Although primal–dual multiplier estimates $z_{\mathcal{I}}$ are usually preferred to primal ones $\mu S_{\mathcal{I}}^{-1}e$, it is important for global convergence that the former do not differ arbitrarily from the latter. To ensure this property, and to encourage fast asymptotic convergence, generated primal–dual estimates are typically projected into a box containing the primal values. This in turn guarantees proximity of the primal–dual Hessian to the pure primal Hessian, which is also required for fast convergence. An alternative is always to compute least-squares multipliers from an estimate of the optimal active set (Dussault 1995).

Some problems may not be defined when the constraints are violated, and methods based on (5.11) directly respect this requirement. Methods based on (5.12) may be modified to address it by resetting slacks to ensure that all iterates are strictly feasible. This is sometimes referred to as 'feasible mode'

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

(Byrd *et al.* 2003) and is often used in practice (Byrd *et al.* 2000). In a linesearch framework, as soon as an iterate x_k strictly satisfies the constraint c_i, $i \in \mathcal{I}$, *i.e.*,

$$c_i(x_k) \geq \epsilon > 0, \tag{5.20}$$

the ith component of the trial slack variables $s_i^{\mathrm{T}} = s_{ki} + d_{si}$ is reset to $c_i(x^{\mathrm{T}}) = c_i(x_k + d_x)$. Should the resulting step be rejected by the merit function, a shorter step (d_x, d_s) is attempted and the process is repeated. In trust region frameworks, the situation is more complicated since possible successive increases in the merit function caused by this reset might dominate decreases attempted by the step. It should also be kept in mind that (5.20) might very well never happen.

In practice it is common to encounter degenerate problems, that is, those for which the set of Lagrange multipliers is unbounded or, worse, does not exist – often such problems result from 'over-modelling'. For instance, it is easily seen that

$$\underset{x\in\mathbb{R}}{\text{minimize}} \;\; f(x) \;\; \text{subject to} \;\; x^2 = 0, \tag{5.21}$$

where $f : \mathbb{R} \to \mathbb{R}$ is such that $f'(0) \neq 0$, admits no Lagrange multiplier. To deal with this possibility (1.1) may be transformed so as to

$$\begin{aligned} \underset{x,s}{\text{minimize}} \quad & \phi^{\mathrm{s}}(x, s; \nu) \stackrel{\text{def}}{=} f(x) + \nu \sum_{i\in\mathcal{E}} [c_i(x) + 2s_i] + \nu \sum_{i\in\mathcal{I}} s_i \\ \text{subject to} \;\; & c_i(x) + s_i \geq 0 \;\; \text{and} \;\; s_i \geq 0, \;\; \text{for all} \;\; i \in \mathcal{E} \cup \mathcal{I}, \end{aligned} \tag{5.22}$$

in terms of so-called *elastic variables* $s_{\mathcal{E}} \in \mathbb{R}^{n_{\mathcal{E}}}$ and $s_{\mathcal{I}} \in \mathbb{R}^{n_{\mathcal{I}}}$; the objective $\phi^{\mathrm{s}}(x, s; \nu)$ is simply a smooth reformulation of the exact ℓ_1-penalty function for (1.1). This new problem is not only smooth but regular – it satisfies the Mangasarian–Fromovitz constraint qualification, and thus has bounded multipliers, for all fixed $\nu > 0$. Furthermore, the problem only involves inequality constraints and is thus well suited to an interior-point approach (Gould, Orban and Toint (2003*b*); see also Tits, Wächter, Bakhtiari, Urban and Lawrence (2003) for a simplified variant).

One other possibility is to balance satisfaction of centrality and feasibility against optimality using a filter. The central idea is to compute a primal–dual step for (5.15) in a manner similar to that described in (5.16). But now, instead of defining a new iterate by a linesearch along the step, or by some classical trust region scheme, a two-dimensional filter with conflicting objectives (see Section 2.1)

$$\theta_1(x, s_{\mathcal{I}}, y, z_{\mathcal{I}}) = \|c_{\mathcal{E}}(x)\| + \|c_{\mathcal{I}}(x) \; - s_{\mathcal{I}}\| + \left\| Z_{\mathcal{I}} s_{\mathcal{I}} - \frac{z_{\mathcal{I}}^T s_{\mathcal{I}}}{n_{\mathcal{I}}} e \right\|$$

$$\text{and} \quad \theta_2(x, s_{\mathcal{I}}, y, z_{\mathcal{I}}) = \frac{z_{\mathcal{I}}^T s_{\mathcal{I}}}{n_{\mathcal{I}}} + \|\nabla_{(x,s_{\mathcal{I}})} \ell(x, s_{\mathcal{I}}, y, z_{\mathcal{I}})\|$$

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

is used to accept or reject the step; here $\ell(x, s_{\mathcal{I}}, y, z_{\mathcal{I}})$ is the Lagrangian of (5.15). The first objective, θ_1, measures feasibility and centrality of the vector $(x, s_{\mathcal{I}}, y, z_{\mathcal{I}})$ while θ_2 attempts to measure optimality. The resulting algorithm, which decomposes the primal–dual step into normal (towards the central path) and tangential (to the central path) components whose sizes are controlled by a trust region scheme, is globally convergent to first-order critical points (Ulbrich, Ulbrich and Vicente 2004).

Practicalities

Many practical issues are to be considered with extreme care when implementing path-following methods. Among those, we have already touched on the treatment of indefiniteness, degeneracy, unboundedness, poor scaling and handling of feasible sets with no strict interior. We now briefly comment on two other outstanding issues, the choice of the initial barrier parameter and its update.

The initial value of μ, although irrelevant in theory, is crucial in practice and may determine the success of a method within the allowed limits. Most algorithms set the initial barrier parameter to some prescribed constant value which seems to perform well on average over a large class of problems, *e.g.*, $\mu_0 = 0.1$.

For the formulation (5.12), if initial values for $s_{\mathcal{I}}$ and $z_{\mathcal{I}}$ are determined using (4.22), the initial value $\mu_0 = s_{\mathcal{I}}^T z_{\mathcal{I}}/n_{\mathcal{I}}$ is reminiscent of linear programming (Wright 1997) to obtain good centrality at the initial point. To take scaling into account and in an attempt to locate nearby points on the central path, one might set $\mu_0 = \max_i \|\nabla c_i(x_0)\|_\infty$ and perform a heuristic test by selecting the value of μ producing the smallest residual in the primal–dual system among the values $0.01\mu_0$, $0.1\mu_0$, μ_0, $10\mu_0$ and $100\mu_0$ – this rule is used by interior-point codes in the GALAHAD library (Gould *et al.* 2003*a*). Perhaps more usefully, if $P_{J_{\mathcal{E}}(x)}(v)$ denotes the orthogonal projection of v onto the null-space of $J_{\mathcal{E}}(x)$ and $\phi(x;\mu)$ is the objective of (5.11), Gay *et al.* (1998) suggest computing

$$\mu_{\rm LS} = \mathrm{argmin}_{\mu>0} \|P_{J_{\mathcal{E}}(x_0)}(\nabla_x \phi(x;\mu))\|,$$

and subsequently setting the initial barrier parameter for (5.11) to the value

$$\mu_0 = \min(100, \max(1, \mu_{\rm LS})).$$

For (5.12), the same recipe involving $P_{J(x)}$ and the objective $\phi(x, s;\mu)$ is appropriate.

In short-step, long-step or predictor–corrector methods for linear programming (Wright 1997) and convex quadratic programming, the barrier parameter is updated using a rule similar to $\mu_{k+1} = \sigma_k s_{\mathcal{I}k}^T z_{\mathcal{I}k}/n_{\mathcal{I}}$, where $0 < \sigma_k < 1$ is a centering parameter. More traditional rules, such as

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

$\mu_{k+1} = \sigma_k \mu_k$ with $0 < \sigma_k < 1$ are commonplace in nonlinear programming, given that there is no concept of duality gap, and virtually all convergence theory has been established for such rules.

In the framework (5.11), Gay *et al.* (1998) suggest the rule

$$\mu_{k+1} = \min\left(\mu_k, \sigma_k \frac{c_{\mathcal{I}}(x)^T z_{\mathcal{I}}}{n_{\mathcal{I}}}\right) \quad \text{where} \quad \sigma_k = \min\left(0.2, 100 \frac{c_{\mathcal{I}}(x)^T z_{\mathcal{I}}}{n_{\mathcal{I}}}\right),$$

where $z_{\mathcal{I}}$ are the estimates of the Lagrange multipliers associated to the inequality constraints of (1.1) at x_k. This rule is clearly reminiscent of linear programming and enforces that $\{\mu_k\}$ be decreasing. For some problems, this monotone behaviour causes difficulties and, sometimes, failure in practice, and more *dynamic* rules are investigated, such as the laxer

$$\mu_{k+1} = \sigma \frac{c_{\mathcal{I}}(x)^T z_{\mathcal{I}}}{n_{\mathcal{I}}},$$

with $0 < \sigma < 1$, which allows the barrier parameter to increase (Bakry, Tapia, Tsuchiya and Zhang 1996). Similar rules have been used in the framework of (5.12), using $s_{\mathcal{I}k}^T z_{\mathcal{I}k}/n_{\mathcal{I}}$ instead of $c_{\mathcal{I}}(x)^T z_{\mathcal{I}}/n_{\mathcal{I}}$.

Vanderbei and Shanno (1999) note that, in practice, it is important to keep individual complementarity pairs clustered together. Using the formulation (5.12), they define

$$\xi_k = \frac{\min_i s_{ki} z_{ki}}{s_{\mathcal{I}k}^T z_{\mathcal{I}k}/n_{\mathcal{I}}}$$

to measure deviation from complementarity and use the heuristic update

$$\mu_{k+1} = 0.1 \min\left(0.05 \frac{1-\xi_k}{\xi_k}, 2\right)^3 \frac{s_{\mathcal{I}k}^T z_{\mathcal{I}k}}{n_{\mathcal{I}}}.$$

Such rules have had some success in practice but are unfortunately not covered by convergence theory and can indeed cause failure if μ becomes too small prematurely or diverges.

Problems with equilibrium constraints

Several formulations of mathematical programs with equilibrium constraints (MPECs) are given in the literature. Generalizing mathematical programs with complementarity constraints (MPCCs), their trait is the presence of a constraint of the form

$$0 \leq F_1(x) \perp F_2(x) \geq 0, \tag{5.23}$$

where $F_1, F_2 : \mathbb{R}^n \to \mathbb{R}^{n_{\mathrm{CC}}}$ and, for $x, y \in \mathbb{R}^{n_{\mathrm{CC}}}$, the notation $x \perp y$ is understood componentwise as meaning $x_i y_i = 0$ for all $i = 1, \ldots, n_{\mathrm{CC}}$. Such a constraint might originate, *e.g.*, from variational inequalities,

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

optimality conditions of the inner problem in a bilevel setting, or from an economic equilibrium requiring that either the price or the excess production for a product be zero. In game theory, F_1 and F_2 might represent the strategy of the leader and the follower, respectively. In design problems, F_1 is the design while F_2 is the response of the system. It is easily seen that problems with a constraint of the form (5.23) violate the Mangasarian–Fromovitz constraint qualification at every feasible point. Such problems thus always have unbounded sets of multipliers which typically consist in *rays*. We refer the reader to the recent overview by Leyffer (2003) for references.

Practical implementations able to reliably treat such problems remain rare and in an active development stage and there is much room left for improvement and the advent of new methods. Of particular importance is the impact of the formulation of the complementarity constraints on the performance of algorithms. An additional difficulty appears when studying interior methods for (1.1) with constraints of the form (5.23) as no central path exists. To circumvent this issue, most practical methods consider a sequence of *relaxed problems* with nonempty strict interior (Scheel and Scholtes 2000), and an interior method is applied to them. A rather simple modification of the step described in the filter linesearch interior algorithm of Wächter and Biegler (2004) is described by Raghunathan and Biegler (2003) who perform a single interior-point iteration per relaxed problem. This modification ensures nonsingularity of the step-defining augmented matrix and alleviates the need for centrality conditions. Numerical difficulties may appear, for in the limit, the strict interior of the feasible set vanishes. DeMiguel, Friedlander, Nogales and Scholtes (2004) propose an alternative where this limit is nonempty, removing the need to modify the search directions.

Anitescu (2000) reformulates MPCCs with nonempty Lagrange multiplier sets by smoothing an ℓ_∞-penalty function. The resulting nonlinear program depends on an elastic variable and has an isolated local minimizer at a solution of the MPCC which, under a quadratic growth condition, can be approached with a finite penalty parameter. This last problem may be solved, *e.g.*, using an SQP approach.

Luo, Pang and Ralph (1998) propose a disjunctive approach, in which the feasible region is decomposed in *branches*, also called *local pieces*. A single SQP step is performed on the nonlinear program defined by the current piece, and all pieces must be examined. Superlinear convergence holds under uniqueness of the multipliers.

Using elastic variables in a manner similar to Anitescu (2000), Benson, Sen, Shanno and Vanderbei (2003) reformulate the MPCC by smoothing an ℓ_∞-penalty function. Under strict complementarity, multipliers at a solution are bounded. The algorithm of Vanderbei and Shanno (1999) implemented

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

in the LOQO package is used to solve the penalty subproblems, using an *ad hoc* rule to update the penalty parameter.

Convergence properties of algorithms for (5.23) typically rely on MPCC-specific regularity conditions, *e.g.*, strong stationarity, the so-called MPCC-LICQ, a strong constraint qualification, and the MPCC-SOSC, a specialized second-order condition. A form of strict complementarity usually ensures fast local convergence. For complete details regarding MPCCs and MPECs, we refer the reader to Luo, Pang and Ralph (1996).

Finally, filter methods can also be adapted for the solution of mixed complementarity problems. Ulbrich (2004*a*) uses a reformulation of the problem into semi-smooth equations, to which a filter method for least-squares (in a variant very close to that described in Section 2.2) is then applied. Although preliminary experiments are interesting, extensive numerical evidence is still missing and the effectiveness of the approach remains to be confirmed.

General convex programs

We finally consider the special case of problems of the form (1.1), in which f is convex and the constraints define a convex feasible set. Interior methods for such problems inherit many properties of those for linear and convex quadratic programming. Algorithms for the latter may therefore relatively painlessly be extended to the former. In particular, the multiple target tracking strategy of Gondzio (1996) generalizes, a key being the reduction of the Newton matrix for the primal–dual equation to a quasi-definite matrix. The method has the peculiarity of defining one barrier parameter per constraint.

The fact that there is a great deal of well-understood theory covering the convex case, and that efficient algorithms from linear programming carry over does not imply by any means that tracking the central path is an easy task. Indeed, even for infinitely differentiable convex data, the central path can exhibit an infinite number of segments of constant length and assume the shape of an 'antenna' or zigzag infinitely (Gilbert, Gonzaga and Karas 2002),

To control the step-size, linesearch-based methods for general convex programming use the ℓ_2 merit function (5.18). The rationale for this approach is that for sufficiently large values of $\nu > 0$, the direction d computed from (5.16) is a descent direction for (5.18) whenever the problem is strictly convex. The additional difficulty introduced by the use of such a merit function is the need to manage the penalty parameter. For most practical purposes, simple updating rules such as $\nu_{k+1} = 10\nu_k$ suffice. More clever rules (approximately) compute the smallest value $\nu_{\min}$ of ν which makes d a descent direction for the merit function, and set $\nu_{k+1} = 10\nu_{\min}$. The linesearch procedure next determines an appropriate step-size based on a fraction-to-the-boundary rule and an Armijo-type acceptance condition.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

Software

Perhaps the most widely known SQP method is SNOPT (Gill *et al.* 2002), a worthy successor to the augmented-Lagrangian-based MINOS (Murtagh and Saunders 1982). Both methods are especially designed for the case where there are relatively few degrees of freedom – and most successful in this case – and neither requires second derivatives. The augmented-Lagrangian-based LANCELOT (Conn *et al.* 1992) operates at the other extreme, being most effective when there are relatively few general constraints, and is capable of running without gradients if necessary – (group) partial separability (Griewank and Toint 1982*b*, Conn *et al.* 1990) allows for the efficient estimation of derivatives. More modern SQP interior-point hybrids like LOQO (Vanderbei and Shanno 1999), KNITRO (Byrd *et al.* 2000) and NLPSPR (Betts and Frank 1994) are effective regardless of the relative number of (active) constraints. Of the filter-based methods, both the trust region SQP-based FilterSQP (Fletcher and Leyffer 1998) and the linesearch interior-point-based IPOPT (Wächter and Biegler 2004) have proved to be robust and efficient. The primal–dual method of Forsgren and Gill (1998) is being implemented in the object-oriented code IOTR which acts as a template for implementing interior-point algorithms. Some codes – for example, the augmented-Lagrangian-based PENNON (Kočvara and Stingl 2003) – have even wider scope, permitting semi-definite matrix constraints. Others, such as CONOPT (Drud 1994) and LSGRG2 (Smith and Lasdon 1992), use (generalized) reduced gradient methods not even covered in this survey. A welcome development has certainly been the flurry of papers – see for example those just cited – comparing and contrasting rival nonlinear programming packages. At this stage, algorithm development is still so rapid that it is impossible to identify the best method(s). We urge potential users to try the award-winning NEOS server (Dolan, Fourer, Moré and Munson 2002, Czyzyk *et al.* 1998)

`www-neos.mcs.anl.gov`

to compare many of the leading contenders.

Turning to convex programming, both MOSEK (Andersen and Andersen 2000, Andersen and Ye 1998) – which is based on a homogeneous model (Andersen and Ye 1999) – and NLPHOPDM (Epelly, Gondzio and Vial 2000) – which applies a multiple target tracking strategy (Gondzio 1996) – are designed for general problems, having evolved from linear programming beginnings. The same is true of PDCO (Saunders and Tomlin 1996), which implements the regularization scheme of Section 4.3. PDCO has been successfully used to solve large-scale entropy maximization problems using Shannon's entropy function $S(x) = -\sum_i x_i \log(x_j)$ as objective and has proved able of solving a maximum entropy model of web traffic with $662,463$ variables and $51,152$ sparse constraints in 12 iterations.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

6. Conclusion

We have reviewed recent developments in algorithms for large-scale optimization, successively considering the unconstrained, bound-constrained, linearly constrained and nonlinearly constrained cases. Emphasis has been put on the underlying principles and theoretical underpinnings of the described methods as well as on practical issues and software.

We are aware that, despite our best efforts, the picture remains incomplete and biased by our experience. This is reflected, for instance, in our lack of cover of neighbouring subjects such as variational inequalities and nonsmooth problems, despite their intrinsic interest. It is nevertheless hoped that the overview presented will make the field of nonlinear programming and its application to solving large problems easier to understand, both for scholars and practitioners.

Acknowledgements

The work of the first author was supported by the EPSRC grant GR/S42170, and that of the second author by NSERC grant RGPIN299010-04 and PIED grant 131FR88. The work of the third author has been conducted in the framework of the Interuniversity Attraction Poles Programme of the Belgian Science Policy Agency. The authors are indebted to Annick Sartenaer for her comments on a draft of the manuscript.

REFERENCES

M. Al-Baali (2003), Quasi-Newton algorithms for large-scale nonlinear least-squares, in Di Pillo and Murli (2003), pp. 1–21.

E. D. Andersen and K. D. Andersen (2000), The MOSEK interior point optimizer for linear programming: An implementation of the homogeneous algorithm, in *High Performance Optimization* (T. T. H. Frenk, K. Roos and S. Zhang, eds), Kluwer, pp. 197–232.

E. D. Andersen and Y. Ye (1998), 'A computational study of the homogeneous algorithm for large-scale convex optimization', *Comput. Optim. Appl.* **10**, 243–269.

E. D. Andersen and Y. Ye (1999), 'On a homogeneous algorithm for the monotone complementarity problem', *Math. Program.* **84**(2), 375–399.

M. Anitescu (2000), On using the elastic mode in nonlinear programming approaches to mathematical programs with complementarity constraints, Preprint ANL/MCS-P864-1200, Argonne National Laboratory, IL, USA.

E. Arian, M. Fahl and E. W. Sachs (2000), Trust-region proper orthogonal decomposition for flow control, Technical Report 2000-25, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA, USA.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

A. S. El Bakry, R. A. Tapia, T. Tsuchiya and Y. Zhang (1996), 'On the formulation and theory of Newton interior point methods for nonlinear programming', *J. Optim. Theory Appl.* **89**(3), 507–541.

R. E. Bank, P. E. Gill and R. F. Marcia (2003), Interior point methods for a class of elliptic variational inequalities, in Biegler, Ghattas, Heinkenschloss and Van Bloemen Waanders (2003), pp. 218–235.

H. Y. Benson, A. Sen, D. F. Shanno and R. J. Vanderbei (2003), Interior-point algorithms, penalty methods and equilibrium problems, Technical Report ORFE-03-02, Operations Research and Financial Engineering, Princeton University.

H. Y. Benson, R. J. Vanderbei and D. F. Shanno (2002), 'Interior-point methods for nonconvex nonlinear programming: Filter methods and merit functions', *Comput. Optim. Appl.* **23**, 257–272.

S. J. Benson, L. C. McInnes, J. Moré and J. Sarich (2004), Scalable algorithms in optimization: Computational experiments, Preprint ANL/MCS-P1175-0604, Mathematics and Computer Science, Argonne National Laboratory, Argonne, IL, USA. To appear in *Proc. 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization* (*MA&O*) *Conference, August 30–September 1, 2004*.

D. P. Bertsekas (1976), 'On the Goldstein–Levitin–Poljak gradient projection method', *IEEE Trans. Automat. Control* **AC-21**, 174–184.

D. P. Bertsekas (1995), *Nonlinear Programming*, Athena Scientific, Belmont, MA, USA.

J. T. Betts and S. O. Erb (2003), 'Optimal low thrust trajectory to the moon', *SIAM J. Appl. Dyn. Syst.* **2**(2), 144–170.

J. T. Betts and P. D. Frank (1994), 'A sparse nonlinear optimization algorithm', *J. Optim. Theory Appl.* **82**(3), 519–541.

T. Biegler, O. Ghattas, M. Heinkenschloss and B. Van Bloemen Waanders, eds (2003), *High Performance Algorithms and Software for Nonlinear Optimization*, Springer, Heidelberg/Berlin/New York.

G. Biros and O. Ghattas (2000), 'A Lagrange–Newton–Krylov–Schur method for PDE-constrained optimization', *SIAG/OPT Views-and-News* **11**(2), 12–18.

R. E. Bixby, M. Fenlon, Z. Gu, E. Rothberg and R. Wunderling (2000), MIP: theory and practice; closing the gap, in *System Modelling and Optimization: Methods, Theory and Applications* (M. J. D. Powell and S. Scholtes, eds), Kluwer, Dordrecht, Netherlands, pp. 10–49.

P. T. Boggs and J. W. Tolle (1995), Sequential quadratic programming, in *Acta Numerica*, Vol. 4, Cambridge University Press, pp. 1–51.

P. T. Boggs and J. W. Tolle (2000), 'Sequential quadratic programming for large-scale nonlinear optimization', *Comput. Appl. Math.* **124**, 123–137.

P. T. Boggs, P. D. Domich, J. E. Rogers and C. Witzgall (1996), 'An interior point method for general large scale quadratic programming problems', *Ann. Oper. Res.* **62**, 419–437.

P. T. Boggs, A. J. Kearsley and J. W. Tolle (1999*a*), 'A global convergence analysis of an algorithm for large scale nonlinear programming problems', *SIAM J. Optim.* **9**(4), 833–862.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

P. T. Boggs, A. J. Kearsley and J. W. Tolle (1999*b*), 'A practical algorithm for general large scale nonlinear optimization problems', *SIAM J. Optim.* **9**(3), 755–778.

J. F. Bonnans, J.-Ch. Gilbert, C. Lemaréchal and C. Sagastizábal (1997), *Optimisation Numérique: Aspects Théoriques et Pratiques*, Vol. 27 of *Mathématiques & Applications*, Springer.

J. M. Borwein (1982), 'Necessary and sufficient conditions for quadratic minimality', *Numer. Funct. Anal. Optim.* **5**, 127–140.

J. H. Bramble (1993), *Multigrid Methods*, Longman, New York.

M. A. Branch, T. F. Coleman and Y. Li (1999), 'A subspace, interior and conjugate gradient method for large-scale bound-constrained minimization problems', *SIAM J. Sci. Comput.* **21**(1), 1–23.

A. Brandt (1977), 'Multi-level adaptative solutions to boundary value problems', *Math. Comp.* **31**(138), 333–390.

W. L. Briggs, V. E. Henson and S. F. McCormick (2000), *A Multigrid Tutorial*, second edn, SIAM, Philadelphia, USA.

A. Brooke, D. Kendrick and A. Meeraus (1988), *GAMS: A User's Guide*, The Scientific Press, Redwood City, USA.

J. V. Burke and J. J. Moré (1988), 'On the identification of active constraints', *SIAM J. Numer. Anal.* **25**(5), 1197–1211.

J. V. Burke and J. J. Moré (1994), 'Exposing constraints', *SIAM J. Optim.* **4**(3), 573–595.

J. V. Burke and A. Weigmann (1997), Notes on limited memory BFGS updating in a trust-region framework, Technical report, Department of Mathematics, University of Washington, Seattle, Washington, USA.

J. V. Burke, J. J. Moré and G. Toraldo (1990), 'Convergence properties of trust region methods for linear and convex constraints', *Math. Program.* **47**(3), 305–336.

R. H. Byrd, J.-Ch. Gilbert and J. Nocedal (2000), 'A trust region method based on interior point techniques for nonlinear programming', *Math. Program., Ser. A* **89**(1), 149–185.

R. H. Byrd, N. I. M. Gould, J. Nocedal and R. A. Waltz (2004*a*), 'An algorithm for nonlinear optimization using linear programming and equality constrained subproblems', *Math. Program., Ser. B* **100**(1), 27–48.

R. H. Byrd, N. I. M. Gould, J. Nocedal and R. A. Waltz (2004*b*), On the convergence of successive linear-quadratic programming algorithms, Technical Report RAL-TR-2004-032, Rutherford Appleton Laboratory, Chilton, Oxfordshire, UK.

R. H. Byrd, M. E. Hribar and J. Nocedal (1999), 'An interior point method for large scale nonlinear programming', *SIAM J. Optim.* **9**(4), 877–900.

R. H. Byrd, P. Lu, J. Nocedal and C. Zhu (1995), 'A limited memory algorithm for bound constrained optimization', *SIAM J. Sci. Comput.* **16**(5), 1190–1208.

R. H. Byrd, J. Nocedal and R. B. Schnabel (1994), 'Representations of quasi-Newton matrices and their use in limited memory methods', *Math. Program.* **63**(2), 129–156.

R. H. Byrd, J. Nocedal and R. A. Waltz (2003), 'Feasible interior methods using slacks for nonlinear optimization', *Comput. Optim. Appl.* **26**, 35–61.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

P. H. Calamai and J. J. Moré (1987), 'Projected gradient methods for linearly constrained problems', *Math. Program.* **39**(1), 93–116.

Y. Chabrillac and J.-P. Crouzeix (1984), 'Definiteness and semidefiniteness of quadratic forms revisited', *Linear Algebra Appl.* **63**, 283–292.

C. M. Chin and R. Fletcher (2003), 'On the global convergence of an SLP-filter algorithm that takes EQP steps', *Math. Program.* **96**(1), 161–177.

T. F. Coleman (1994), Linearly constrained optimization and projected preconditioned conjugate gradients, in *Proc. Fifth SIAM Conference on Applied Linear Algebra* (J. Lewis, ed.), SIAM, Philadelphia, USA, pp. 118–122.

T. F. Coleman and A. R. Conn (1982), 'Nonlinear programming via an exact penalty function method: Asymptotic analysis', *Math. Program.* **24**(3), 123–136.

T. F. Coleman and L. A. Hulbert (1989), 'A direct active set algorithm for large sparse quadratic programs with simple bounds', *Math. Program., Ser. B* **45**(3), 373–406.

T. F. Coleman and Y. Li, eds (1990), *Large Scale Numerical Optimization*, SIAM, Philadelphia, USA.

T. F. Coleman and Y. Li (1994), 'On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds', *Math. Program.* **67**(2), 189–224.

T. F. Coleman and Y. Li (1996), 'An interior trust region approach for nonlinear minimization subject to bounds', *SIAM J. Optim.* **6**(2), 418–445.

T. F. Coleman and Y. Li (2000), 'A trust region and affine scaling interior point method for nonconvex minimization with linear inequality constraints', *Math. Program., Ser. A* **88**, 1–31.

B. Colson and Ph. L. Toint (2003), Optimizing partially separable functions without derivatives, Technical Report 03/20, Department of Mathematics, University of Namur, Namur, Belgium.

A. R. Conn, N. I. M. Gould and Ph. L. Toint (1988*a*), 'Global convergence of a class of trust region algorithms for optimization with simple bounds', *SIAM J. Numer. Anal.* **25**(2), 433–460. See also same journal **26** (1989), 764–767.

A. R. Conn, N. I. M. Gould and Ph. L. Toint (1988*b*), 'Testing a class of methods for solving minimization problems with simple bounds on the variables', *Math. Comp.* **50**, 399–430.

A. R. Conn, N. I. M. Gould and Ph. L. Toint (1990), An introduction to the structure of large scale nonlinear optimization problems and the LANCELOT project, in *Computing Methods in Applied Sciences and Engineering* (R. Glowinski and A. Lichnewsky, eds), SIAM, Philadelphia, USA, pp. 42–51.

A. R. Conn, N. I. M. Gould and Ph. L. Toint (1992), LANCELOT*: A Fortran package for Large-scale Nonlinear Optimization* (*Release A*), Springer Series in Computational Mathematics, Springer, Heidelberg/Berlin/New York.

A. R. Conn, N. I. M. Gould and Ph. L. Toint (1994), Large-scale nonlinear constrained optimization: a current survey, in *Algorithms for Continuous Optimization: The State of the Art* (E. Spedicato, ed.), Vol. 434 of *NATO ASI Series C: Mathematical and Physical Sciences*, Kluwer, Dordrecht, Netherlands, pp. 287–332.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

A. R. Conn, N. I. M. Gould and Ph. L. Toint (1996), 'Numerical experiments with the LANCELOT package (Release A) for large-scale nonlinear optimization', *Math. Program., Ser. A* **73**(1), 73–110.

A. R. Conn, N. I. M. Gould and Ph. L. Toint (1997), Methods for nonlinear constraints in optimization calculations, in Duff and Watson (1997), pp. 363–390.

A. R. Conn, N. I. M. Gould and Ph. L. Toint (2000*a*), *Trust-Region Methods*, SIAM, Philadelphia, USA.

A. R. Conn, N. I. M. Gould, D. Orban and Ph. L. Toint (2000*b*), 'A primal-dual trust-region algorithm for non-convex nonlinear programming', *Math. Program., Ser. B* **87**(2), 215–249.

B. L. Contesse (1980), 'Une caractérisation complète des minima locaux en programmation quadratique', *Numer. Math.* **34**(3), 315–332.

CPLEX 6.0 (1998), *High-Performance Linear, Integer and Quadratic Programming Software*, ILOG SA, Gentilly, France. `www.cplex.com`.

J. Czyzyk, M. Mesnier and J. Moré (1998), 'The NEOS server', *IEEE J. Comput. Sci. Engr.* **5**, 68–75.

Y. H. Dai and Y. Yuan (2000), 'A nonlinear conjugate gradient method with a strong global convergence property', *SIAM J. Optim.* **10**(1), 177–182.

R. S. Dembo and T. Steihaug (1983), 'Truncated-Newton algorithms for large-scale unconstrained optimization', *Math. Program.* **26**(2), 190–212.

R. S. Dembo, S. C. Eisenstat and T. Steihaug (1982), 'Inexact-Newton methods', *SIAM J. Numer. Anal.* **19**(2), 400–408.

A.-V. DeMiguel, M. P. Friedlander, F. J. Nogales and S. Scholtes (2004), An interior-point method for MPECs based on strictly feasible relaxations, Technical Report ANL/MCS-P1150-0404, Argonne National Laboratory, IL, USA.

N. Deng, Y. Xiao and F. Zhou (1993), 'Nonmonotonic trust region algorithms', *J. Optim. Theory Appl.* **76**(2), 259–285.

J. E. Dennis and R. B. Schnabel (1983), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, USA. Reprinted as Vol. 16 of *Classics in Applied Mathematics*, SIAM, Philadelphia, USA.

J. E. Dennis, D. M. Gay and R. E. Welsh (1981), 'An adaptive nonlinear least squares algorithm', *ACM Trans. Math. Software* **7**(3), 348–368.

G. Di Pillo and F. Gianessi, eds (1996), *Nonlinear Optimization and Applications*, Plenum Publishing, New York.

G. Di Pillo and F. Gianessi, eds (1999), *Nonlinear Optimization and Related Topics*, Vol. 2, Kluwer, Dordrecht, Netherlands.

G. Di Pillo and A. Murli, eds (2003), *High Performance Algorithms and Software in Nonlinear Optimization*, Kluwer, Dordrecht, Netherlands.

E. Dolan (2001), The NEOS server 4.0 administrative guide, Technical Memorandum ANL/MCS-TM-250, Argonne National Laboratory, IL.

E. D. Dolan, R. Fourer, J. J. Moré and T. S. Munson (2002), 'Computing a trust region step', *SIAM News* **35**(5), 8–9.

A. S. Drud (1994), 'CONOPT: A large scale GRG code', *ORSA J. Comput.* **6**, 207–216.

I. Duff and A. Watson, eds (1997), *The State of the Art in Numerical Analysis*, Oxford University Press, Oxford.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

J. C. Dunn (1981), 'Global and asymptotic convergence rate estimates for a class of projected gradient processes', *SIAM J. Control Optim.* **19**, 368–400.

J.-P. Dussault (1995), 'Numerical stability and efficiency of penalty algorithms', *SIAM J. Numer. Anal.* **32**(1), 296–317.

O. Epelly, J. Gondzio and J.-P. Vial (2000), An interior-point solver for smooth convex optimization with an application to environmental-energy-economic models, Technical Report 2000.08, Logilab, HEC, University of Geneva, Switzerland.

F. Facchinei, J. Judice and J. Soares (1998), 'An active set Newton algorithm for large-scale nonlinear programs with box constraints', *SIAM J. Optim.* **8**(1), 158–186.

F. Facchinei, S. Lucidi and L. Palagi (2002), 'A truncated Newton algorithm for large scale box constrained optimization', *SIAM J. Optim.* **12**(4), 1100–1125.

M. Fahl and E. Sachs (2003), Reduced order modelling approaches to PDE-constrained optimization based on proper orthogonal decomposition, in Biegler *et al.* (2003), pp. 268–281.

M. C. Ferris and T. S. Munson (2000), Interior-point methods for massive support vector machines, Data Mining Institute Technical Report 00-05, Computer Science Department, University of Wisconsin, Madison, WI, USA.

A. V. Fiacco and G. P. McCormick (1968), *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Wiley, Chichester, UK. Reprinted as *Classics in Applied Mathematics*, SIAM, Philadelphia, USA (1990).

M. Fisher (1998), Minimization algorithms for variational data assimilation, in *Recent Developments in Numerical Methods for Atmospheric Modelling*, ECMWF, pp. 364–385.

R. Fletcher (1971), 'A general quadratic programming algorithm', *J. Inst. Math. Appl.* **7**, 76–91.

R. Fletcher (1981), *Practical Methods of Optimization: Constrained Optimization*, Wiley, Chichester, UK.

R. Fletcher (1987*a*), *Practical Methods of Optimization*, second edn, Wiley, Chichester, UK.

R. Fletcher (1987*b*), Recent developments in linear and quadratic programming, in *The State of the Art in Numerical Analysis* (A. Iserles and M. J. D. Powell, eds), Oxford University Press, Oxford, pp. 213–243.

R. Fletcher (2000), 'Stable reduced Hessian updates for indefinite quadratic programming', *Math. Program.* **87**(2), 251–264.

R. Fletcher and S. Leyffer (1998), User manual for filterSQP, Numerical Analysis Report NA/181, Department of Mathematics, University of Dundee, Dundee, UK.

R. Fletcher and S. Leyffer (2002), 'Nonlinear programming without a penalty function', *Math. Program.* **91**(2), 239–269.

R. Fletcher and C. M. Reeves (1964), 'Function minimization by conjugate gradients', *Computer Journal* **7**, 149–154.

R. Fletcher and E. Sainz de la Maza (1989), 'Nonlinear programming and nonsmooth optimization by successive linear programming', *Math. Program.* **43**(3), 235–256.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

R. Fletcher, N. I. M. Gould, S. Leyffer, Ph. L. Toint and A. Wächter (2002*a*), 'Global convergence of trust-region SQP-filter algorithms for nonlinear programming', *SIAM J. Optim.* **13**(3), 635–659.

R. Fletcher, S. Leyffer and Ph. L. Toint (2002*b*), 'On the global convergence of a filter-SQP algorithm', *SIAM J. Optim.* **13**(1), 44–59.

A. Forsgren (2002), 'Inertia-controlling factorizations for optimization algorithms', *Appl. Numer. Math.* **43**(1–2), 91–107.

A. Forsgren and P. E. Gill (1998), 'Primal-dual interior methods for nonconvex nonlinear programming', *SIAM J. Optim.* **8**(4), 1132–1152.

A. Forsgren and W. Murray (1993), 'Newton methods for large-scale linear equality-constrained minimization', *SIAM J. Matrix Anal. Appl.* **14**(2), 560–587.

A. Forsgren, P. E. Gill and M. H. Wright (2002), 'Interior-point methods for nonlinear optimization', *SIAM Review* **44**, 525–597.

R. Fourer, D. M. Gay and B. W. Kernighan (2003), *AMPL: A Modeling Language for Mathematical Programming*, second edn, Brooks/Cole-Thompson Learning, Pacific Grove, CA, USA.

M. P. Friedlander and M. A. Saunders (2005), 'A globally convergent linearly constrained Lagrangian method for nonlinear optimization', *SIAM J. Optim.*, to appear.

D. M. Gay, M. L. Overton and M. H. Wright (1998), A primal-dual interior method for nonconvex nonlinear programming, in *Advances in Nonlinear Programming* (Y. Yuan, ed.), Kluwer, Dordrecht, Netherlands, pp. 31–56.

E. M. Gertz and Ph. E. Gill (2004), 'A primal-dual trust region algorithm for nonlinear optimization', *Math. Program., Ser. A* **100**(1), 49–94.

E. M. Gertz and S. J. Wright (2003), 'Object-oriented software for quadratic programming', *Trans. ACM Math. Software* **29**(1), 58–81.

E. M. Gertz, J. Nocedal and A. Sartenaer (2003), A starting-point strategy for nonlinear interior methods, Technical Report OTC 2003/4, Optimization Technology Center, Evanston, IL, USA.

J.-Ch. Gilbert and C. Lemaréchal (1989), 'Some numerical experiments with variable-storage quasi-Newton algorithms', *Math. Program., Ser. B* **45**(3), 407–435.

J.-Ch. Gilbert and J. Nocedal (1992), 'Global convergence properties of conjugate gradient methods for optimization', *SIAM J. Optim.* **2**(1), 21–42.

J.-Ch. Gilbert, C. C. Gonzaga and E. Karas (2002), Examples of ill-behaved central paths in convex optimization, Technical Report 4179, INRIA, Rocquencourt, Le Chesnay, France.

P. E. Gill, W. Murray and M. A. Saunders (2002), 'SNOPT: An SQP algorithm for large-scale constrained optimization', *SIAM J. Optim.* **12**(4), 979–1006.

P. E. Gill, W. Murray and M. H. Wright (1981), *Practical Optimization*, Academic Press, London.

P. E. Gill, W. Murray, M. A. Saunders and M. H. Wright (1990), A Schur-complement method for sparse quadratic programming, in *Reliable Scientific Computation* (M. G. Cox and S. J. Hammarling, eds), Oxford University Press, pp. 113–138.

P. E. Gill, W. Murray, M. A. Saunders and M. H. Wright (1991), 'Inertia-controlling methods for general quadratic programming', *SIAM Review* **33**(1), 1–36.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

M. J. Goldsmith (1999), Sequential quadratic programming methods based on indefinite Hessian approximations, PhD thesis, Dept of Management Science and Engineering, Stanford University, CA, USA.

J. Gondzio (1996), 'Multiple centrality corrections in a primal-dual method for linear programming', *Comput. Optim. Appl.* **6**, 137–156.

J. Gondzio (1997), 'Presolve analysis of linear programs prior to applying an interior point method', *INFORMS J. Comput.* **9**(1), 73–91.

J. Gondzio and A. Grothey (2003*a*), Parallel interior point solver for structured quadratic programs: Application to financial planning problems, Technical Report MS-03-001, School of Mathematics, University of Edinburgh.

J. Gondzio and A. Grothey (2003*b*), 'Reoptimization with the primal-dual interior point method', *SIAM J. Optim.* **13**(3), 842–864.

C. C. Gonzaga, E. Karas and M. Vanti (2003), 'A globally convergent filter method for nonlinear programming', *SIAM J. Optim.* **14**(3), 646–669.

N. I. M. Gould (1985), 'On practical conditions for the existence and uniqueness of solutions to the general equality quadratic-programming problem', *Math. Program.* **32**(1), 90–99.

N. I. M. Gould (1986), 'On the accurate determination of search directions for simple differentiable penalty functions', *IMA J. Numer. Anal.* **6**, 357–372.

N. I. M. Gould (1991), 'An algorithm for large-scale quadratic programming', *IMA J. Numer. Anal.* **11**(3), 299–324.

N. I. M. Gould (1999), 'On modified factorizations for large-scale linearly-constrained optimization', *SIAM J. Optim.* **9**(4), 1041–1063.

N. I. M. Gould (2003), 'Some reflections on the current state of active-set and interior point methods for constrained optimization', *SIAG/OPT Views-and-News* **14**(1), 2–7.

N. I. M. Gould, and Ph. L. Toint (2000*a*), A quadratic programming bibliography, Numerical Analysis Group Internal Report 2000-1, Rutherford Appleton Laboratory, Chilton, Oxfordshire, UK.

N. I. M. Gould and Ph. L. Toint (2000*b*), SQP methods for large-scale nonlinear programming, in *System Modelling and Optimization: Methods, Theory and Applications* (M. J. D. Powell and S. Scholtes, eds), Kluwer, Dordrecht, Netherlands, pp. 149–178.

N. I. M. Gould and Ph. L. Toint (2002*a*), 'An iterative working-set method for large-scale non-convex quadratic programming', *Appl. Numer. Math.* **43**(1–2), 109–128.

N. I. M. Gould and Ph. L. Toint (2002*b*), Numerical methods for large-scale non-convex quadratic programming, in *Trends in Industrial and Applied Mathematics* (A. H. Siddiqi and M. Kočvara, eds), Kluwer, Dordrecht, Netherlands, pp. 149–179.

N. I. M. Gould and Ph. L. Toint (2003*a*), FILTRANE: A Fortran 95 filter-trust-region package for solving systems of nonlinear equalities, nonlinear inequalities and nonlinear least-squares problems, Technical Report 03/15, Rutherford Appleton Laboratory, Chilton, Oxfordshire, UK.

N. I. M. Gould and Ph. L. Toint (2003*b*), Global convergence of a hybrid trust-region SQP-filter algorithm for general nonlinear programming, in *System*

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

Modeling and Optimization XX (E. Sachs and R. Tichatschke, eds), Kluwer, Dordrecht, Netherlands, pp. 23–54.

N. I. M. Gould and Ph. L. Toint (2004*a*), How mature is nonlinear optimization?, in *Applied Mathematics Entering the 21st Century: Invited Talks from the ICIAM 2003 Congress* (J. M. Hill and R. Moore, eds), SIAM, Philadelphia, USA, pp. 141–161.

N. I. M. Gould and Ph. L. Toint (2004*b*), 'Preprocessing for quadratic programming', *Math. Program., Ser. B* **100**(1), 95–132.

N. I. M. Gould and Ph. L. Toint (2005), Global convergence of a non-monotone trust-region filter algorithm for nonlinear programming, in *Proc. 2004 Gainesville Conference on Multilevel Optimization* (W. Hager, ed.), Kluwer, Dordrecht, Netherlands, to appear.

N. I. M. Gould, M. E. Hribar and J. Nocedal (2001), 'On the solution of equality constrained quadratic problems arising in optimization', *SIAM J. Sci. Comput.* **23**(4), 1375–1394.

N. I. M. Gould, S. Leyffer and Ph. L. Toint (2005), 'A multidimensional filter algorithm for nonlinear equations and nonlinear least-squares', *SIAM J. Optim.* **15**(1), 17–38.

N. I. M. Gould, S. Lucidi, M. Roma and Ph. L. Toint (1999), 'Solving the trust-region subproblem using the Lanczos method', *SIAM J. Optim.* **9**(2), 504–525.

N. I. M. Gould, S. Lucidi, M. Roma and Ph. L. Toint (2000), 'Exploiting negative curvature directions in linesearch methods for unconstrained optimization', *Optim. Methods Software* **14**(1–2), 75–98.

N. I. M. Gould, D. Orban and Ph. L. Toint (2003*a*), 'GALAHAD: A library of thread-safe Fortran 90 packages for large-scale nonlinear optimization', *ACM Trans. Math. Software* **29**(4), 353–372.

N. I. M. Gould, D. Orban and Ph. L. Toint (2003*b*), An interior-point ℓ_1-penalty method for nonlinear optimization, Technical Report RAL-TR-2003-0xx, Rutherford Appleton Laboratory, Chilton, Oxfordshire, UK.

N. I. M. Gould, C. Sainvitu and Ph. L. Toint (2004), A filter-trust-region method for unconstrained optimization, Technical Report 04/03, Department of Mathematics, University of Namur, Belgium.

S. Gratton, A. Sartenaer and Ph. L. Toint (2004), Recursive trust-region methods for multilevel nonlinear optimization (Part I): Global convergence and complexity, Technical Report 04/06, Department of Mathematics, University of Namur, Belgium.

A. Griewank (2000), *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, Vol. 19 of *Frontiers in Applied Mathematics*, SIAM, Philadelphia, USA.

A. Griewank and Ph. L. Toint (1982*a*), 'Local convergence analysis for partitioned quasi-Newton updates', *Numer. Math.* **39**, 429–448.

A. Griewank and Ph. L. Toint (1982*b*), On the unconstrained optimization of partially separable functions, in *Nonlinear Optimization 1981* (M. J. D. Powell, ed.), Academic Press, London, pp. 301–312.

A. Griewank and Ph. L. Toint (1982*c*), 'Partitioned variable metric updates for large structured optimization problems', *Numer. Math.* **39**, 119–137.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

R. E. Griffith and R. A. Stewart (1961), 'A nonlinear programming technique for the optimization of continuous processing systems', *Management Science* **7**, 379–392.

L. Grippo, F. Lampariello and S. Lucidi (1986), 'A nonmonotone line search technique for Newton's method', *SIAM J. Numer. Anal.* **23**(4), 707–716.

L. Grippo, F. Lampariello and S. Lucidi (1989), 'A truncated Newton method with nonmonotone line search for unconstrained optimization', *J. Optim. Theory Appl.* **60**(3), 401–419.

W. Gropp and J. Moré (1997), Optimization environments and the NEOS server, in *Approximation Theory and Optimization* (M. D. Buhmann and A. Iserles, eds), Cambridge University Press, pp. 167–182.

C. Guéret, C. Prins and M. Seveaux (2002), *Applications of Optimization with Xpress-MP*, Dash Optimization. `www.dashoptimization.com`.

M. Gulliksson, I. Söderkvist and P.-A. Wedin (1997), 'Algorithms for constrained and weighted nonlinear least-squares', *SIAM J. Optim.* **7**(1), 208–224.

W. Hackbusch (1995), *Multi-Grid Methods and Applications*, Vol. 4 of *Series in Computational Mathematics*, Springer, Heidelberg/Berlin/New York.

W. W. Hager (2001), 'Minimizing a quadratic over a sphere', *SIAM J. Optim.* **12**(1), 188–208.

W. W. Hager and H. Zhang (2003), CG_DESCENT: A conjugate-gradient method with guaranteed descent, Technical report, Department of Mathematics, University of Florida, Gainesville, USA.

W. W. Hager, D. W. Hearn and P. M. Pardalos, eds (1994), *Large Scale Optimization: State of the Art*, Kluwer, Dordrecht, Netherlands.

M. Heinkenschloss, M. Ulbrich and S. Ulbrich (1999), 'Superlinear and quadratic convergence of affine-scaling interior-point Newton methods for problems with simple bounds without strict complementarity assumption', *Math. Program.* **86**(3), 615–635.

M. R. Hestenes and E. Stiefel (1952), 'Methods of conjugate gradients for solving linear systems', *J. Res. National Bureau of Standards* **49**, 409–436.

W. Hock and K. Schittkowski (1981), *Test Examples for Nonlinear Programming Codes*, Vol. 187 of *Lecture Notes in Economics and Mathematical Systems*, Springer, Heidelberg/Berlin/New York.

B. Jansen, C. Roos, T. Terlaky and J.-Ph. Vial (1996), 'Primal-dual target following algorithms for linear programming', *Ann. Oper. Res.* **62**, 197–231.

L. C. Kaufman (1999), 'Reduced storage, quasi-Newton trust region approaches to function optimization', *SIAM J. Optim.* **10**(1), 56–69.

M. Kočvara and M. Stingl (2003), 'PENNON, a code for nonconvex nonlinear and semidefinite programming', *Optim. Methods Software* **18**(3), 317–333.

M. Lalee, J. Nocedal and T. D. Plantenga (1998), 'On the implementation of an algorithm for large-scale equality constrained optimization', *SIAM J. Optim.* **8**(3), 682–706.

R. D. Leone, A. Murli, P. M. Pardalos and G. Toraldo, eds (1998), *High Performance Algorithms and Software in Nonlinear Optimization*, Kluwer, Dordrecht, Netherlands.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

M. Lescrenier (1991), 'Convergence of trust region algorithms for optimization with bounds when strict complementarity does not hold', *SIAM J. Numer. Anal.* **28**(2), 476–495.

E. S. Levitin and B. T. Polyak (1966), 'Constrained minimization problems', *USSR Comput. Math. Math. Phys.* **6**, 1–50.

A. S. Lewis and M. L. Overton (1996), Eigenvalue optimization, in *Acta Numerica*, Vol. 5, Cambridge University Press, pp. 149–190.

M. Lewis and S. G. Nash (2002), Practical aspects of multiscale optimization methods for VLSICAD, in *Multiscale Optimization and VLSI/CAD* (J. Cong and J. R. Shinnerl, eds), Kluwer, Dordrecht, Netherlands, pp. 265–291.

M. Lewis and S. G. Nash (2005), 'Model problems for the multigrid optimization of systems governed by differential equations', *SIAM J. Sci. Comput.*, to appear.

S. Leyffer (2003), 'Mathematical programs with complementarity constraints', *SIAG/OPT Views-and-News* **14**(1), 15–18.

C. Lin and J. J. Moré (1999*a*), 'Incomplete Cholesky factorizations with limited memory', *SIAM J. Sci. Comput.* **21**(1), 24–45.

C. Lin and J. J. Moré (1999*b*), 'Newton's method for large bound-constrained optimization problems', *SIAM J. Optim.* **9**(4), 1100–1127.

D. C. Liu and J. Nocedal (1989), 'On the limited memory BFGS method for large-scale optimization', *Math. Program., Ser. B* **45**(3), 503–528.

D. G. Luenberger (1984), *Linear and Nonlinear Programming*, 2nd edn, Addison-Wesley, Reading, MA, USA.

L. Lukšan (1993), 'Inexact trust region method for large sparse nonlinear least-squares', *Kybernetica* **29**(4), 305–324.

L. Lukšan (1994), 'Inexact trust region method for large sparse systems of nonlinear equations', *J. Optim. Theory Appl.* **81**(3), 569–590.

L. Lukšan (1996), 'Hybrid methods for large sparse nonlinear least-squares', *J. Optim. Theory Appl.* **89**(3), 575–595.

L. Lukšan and J. Vlček (1998), 'Indefinitely preconditioned inexact Newton method for large sparse equality constrained nonlinear programming problems', *Numer. Linear Algebra Appl.* **5**(3), 219–247.

Z.-Q. Luo, J. S. Pang and D. Ralph (1996), *Mathematical Programs with Equilibrium Constraints*, Cambridge University Press, Cambridge.

Z.-Q. Luo, J. S. Pang and D. Ralph (1998), Piecewise sequential quadratic programming for mathematical programs with complementarity constraints, in *Multilevel Optimization: Complexity and Applications* (A. Migdala *et al.*, ed.), Kluwer.

O. L. Mangasarian (1980), 'Locally unique solutions of quadratic programs, linear and non-linear complementarity problems', *Math. Program., Ser. B* **19**(2), 200–212.

N. Maratos (1978), Exact penalty function algorithms for finite-dimensional and control optimization problems, PhD thesis, University of London.

M. Marazzi and J. Nocedal (2001), Feasibility control in nonlinear optimization, in *Foundations of Computational Mathematics* (A. DeVore, A. Iserles and E. Suli, eds), Vol. 284 of *London Mathematical Society Lecture Note Series*, Cambridge University Press, pp. 125–154.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

D. Q. Mayne and E. Polak (1976), 'Feasible directions algorithms for optimisation problems with equality and inequality constraints', *Math. Program.* **11**(1), 67–80.

S. Mehrotra (1992), 'On the implementation of a primal-dual interior point method', *SIAM J. Optim.* **2**, 575–601.

J. L. Morales and J. Nocedal (2000), 'Automatic preconditioning by limited memory quasi-Newton updating', *SIAM J. Optim.* **10**(4), 1079–1096.

J. J. Moré (2003), 'Terascale optimal PDE solvers', Talk at the ICIAM 2003 Conference in Sydney.

J. J. Moré and D. C. Sorensen (1983), 'Computing a trust region step', *SIAM J. Sci. Statist. Comput.* **4**(3), 553–572.

J. J. Moré and D. J. Thuente (1994), 'Line search algorithms with guaranteed sufficient decrease', *ACM Trans. Math. Software* **20**(3), 286–307.

J. J. Moré and G. Toraldo (1991), 'On the solution of large quadratic programming problems with bound constraints', *SIAM J. Optim.* **1**(1), 93–113.

J. J. Moré and S. J. Wright (1993), *Optimization Software Guide*, Vol. 14 of *Frontiers in Applied Mathematics*, SIAM, Philadelphia, USA.

W. Murray and F. J. Prieto (1995), 'A sequential quadratic programming algorithm using an incomplete solution of the subproblem', *SIAM J. Optim.* **5**(3), 590–640.

W. Murray and M. H. Wright (1992), Project Lagrangian methods based on the trajectories of penalty and barrier functions, Numerical Analysis Manuscript 92-01, AT&T Bell Laboratories.

B. A. Murtagh and M. A. Saunders (1982), 'A projected Lagrangian algorithm and its implementation for sparse non-linear constraints', *Math. Program. Studies* **16**, 84–117.

K. G. Murty and S. N. Kabadi (1987), 'Some NP-complete problems in quadratic and nonlinear programming', *Math. Program.* **39**(2), 117–129.

S. G. Nash (1984), 'Newton-type minimization via the Lanczos method', *SIAM J. Numer. Anal.* **21**(4), 770–788.

S. G. Nash (2000*a*), 'A multigrid approach to discretized optimization problems', *Optim. Methods Software* **14**, 99–116.

S. G. Nash (2000*b*), 'A survey of truncated-Newton methods', *J. Comput. Appl. Math.* **124**, 45–59.

S. G. Nash and J. Nocedal (1991), 'A numerical study of the limited memory BFGS method and the truncated-Newton method for large-scale optimization', *SIAM J. Optim.* **1**(3), 358–372.

S. G. Nash and A. Sofer (1990), 'Assessing a search direction within a truncated-Newton method', *Oper. Res. Lett.* **9**(4), 219–221.

Y. Nesterov and A. Nemirovskii (1994), *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM, Philadelphia, USA.

J. Nocedal (1980), 'Updating quasi-Newton matrices with limited storage', *Math. Comp.* **35**, 773–782.

J. Nocedal (1992), Theory of algorithms for unconstrained optimization, in *Acta Numerica*, Vol. 1, Cambridge University Press, pp. 199–242.

J. Nocedal (1997), Large scale unconstrained optimization, in Duff and Watson (1997), pp. 311–338.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

J. Nocedal and S. J. Wright (1999), *Large Sparse Numerical Optimization*, Series in Operations Research, Springer, Heidelberg/Berlin/New York.

E. O. Omojokun (1989), Trust region algorithms for optimization with nonlinear equality and inequality constraints, PhD thesis, University of Colorado, Boulder, Colorado, USA.

C. C. Paige and M. A. Saunders (1982), 'LSQR: an algorithm for sparse linear equations and sparse least squares', *Trans. ACM Math. Software* **8**, 43–71.

E. Polak and G. Ribière (1969), 'Note sur la convergence de méthodes de directions conjuguées', *Revue Francaise d'Informatique et de Recherche Opérationelle* **16-R1**, 35–43.

B. T. Polyak (1969), 'The conjugate gradient method in extremal problems', *USSR Comput. Math. Math. Phys.* **9**, 94–112.

M. J. D. Powell (1977), 'Restart procedures for the conjugate gradient method', *Math. Program.* **12**(2), 241–254.

M. J. D. Powell (1998), Direct search algorithms for optimization calculations, in *Acta Numerica*, Vol. 7, Cambridge University Press, pp. 287–336.

C. J. Price and Ph. L. Toint (2004), Exploiting problem structure in pattern-search methods for unconstrained optimization, Technical Report November, Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand.

R. Pytlak (1998), 'An efficient algorithm for large-scale nonlinear programming problems with simple bounds on the variables', *SIAM J. Optim.* **8**(2), 532–560.

A. U. Raghunathan and L. T. Biegler (2003), Interior point methods for Mathematical Programs with Complementarity Constraints (MPCCs), Technical Report, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA.

F. Rendl and H. Wolkowicz (1997), 'A semidefinite framework for trust region subproblems with applications to large scale minimization', *Math. Program.* **77**(2), 273–299.

J. Renegar (2001), *A Mathematical View of Interior-Point Methods in Convex Optimization*, MPS/SIAM series on Optimization, SIAM, Philadelphia, PA, USA.

S. M. Robinson (1974), 'Perturbed Kuhn–Tucker points and rates of convergence for a class of nonlinear programming algorithms', *Math. Program.* **7**(1), 1–16.

M. A. Saunders and J. A. Tomlin (1996), Solving regularized linear programs using barrier methods and KKT systems, Technical Report SOL 96-4, Department of EESOR, Stanford University, Stanford, CA, USA.

H. Scheel and S. Scholtes (2000), 'Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity', *Math. Oper. Res.* **25**, 1–22.

T. Schlick (1993), 'Modified Cholesky factorizations for sparse preconditioners', *SIAM J. Sci. Comput.* **14**(2), 424–445.

J. A. Scott, Y. Hu and N. I. M. Gould (2004), An evaluation of sparse direct symmetric solvers: An introduction and preliminary findings, Numerical Analysis Group Internal Report 2004-1, Rutherford Appleton Laboratory, Chilton, Oxfordshire, UK.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

S. Smith and L. Lasdon (1992), 'Solving large sparse nonlinear programs using GRG', *ORSA J. Comput.* **4**, 1–15.
D. C. Sorensen (1997), 'Minimization of a large-scale quadratic function subject to a spherical constraint', *SIAM J. Optim.* **7**(1), 141–161.
E. Spedicato, ed. (1994), *Algorithms for Continuous Optimization: The State of the Art*, Vol. 434 of *NATO ASI Series C: Mathematical and Physical Sciences*, Kluwer, Dordrecht, Netherlands.
T. Steihaug (1983), 'The conjugate gradient method and trust regions in large scale optimization', *SIAM J. Numer. Anal.* **20**(3), 626–637.
A. Tits, A. Wächter, S. Bakhtiari, T. J. Urban and C. T. Lawrence (2003), 'A primal-dual interior-point method for nonlinear programming with strong global and local convergence properties', *SIAM J. Optim.* **14**(1), 173–199.
M. J. Todd (2001), Semidefinite optimization, in *Acta Numerica*, Vol. 10, Cambridge University Press, pp. 515–560.
Ph. L. Toint (1981), Towards an efficient sparsity exploiting Newton method for minimization, in *Sparse Matrices and Their Uses* (I. S. Duff, ed.), Academic Press, London, pp. 57–88.
Ph. L. Toint (1987), 'On large scale nonlinear least squares calculations', *SIAM J. Sci. Statist. Comput.* **8**(3), 416–435.
Ph. L. Toint (1988), 'Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space', *IMA J. Numer. Anal.* **8**, 231–252.
Ph. L. Toint (1996), 'An assessment of non-monotone linesearch techniques for unconstrained optimization', *SIAM J. Sci. Statist. Comput.* **17**(3), 725–739.
Ph. L. Toint (1997), 'A non-monotone trust-region algorithm for nonlinear optimization subject to convex constraints', *Math. Program.* **77**(1), 69–94.
M. Ulbrich (2004*a*), 'A multidimensional filter trust-region method for mixed complementarity problems', Talk at ICCOPT 1, Troy, USA.
M. Ulbrich, S. Ulbrich and L. N. Vicente (2004), 'A globally convergence primal-dual interior-point filter method for nonlinear programming', *Math. Program., Ser. B* **100**(2), 379–410.
S. Ulbrich (2004*b*), 'On the superlinear local convergence of a filter-SQP method', *Math. Program., Ser. B* **100**(1), 217–245.
R. J. Vanderbei (1995), 'Symmetric quasi-definite matrices', *SIAM J. Optim.* **5**, 100–113.
R. J. Vanderbei (1999), 'LOQO: An interior point code for quadratic programming', *Optim. Methods Software* **12**, 451–484.
R. J. Vanderbei and D. F. Shanno (1999), 'An interior point algorithm for nonconvex nonlinear programming', *Comput. Optim. Appl.* **13**, 231–252.
S. A. Vavasis (1990), 'Quadratic programming is is NP', *Inform. Process. Lett.* **36**(2), 73–77.
S. A. Vavasis (1991), Convex quadratic programming, in *Nonlinear Optimization: Complexity Issues*, Oxford University Press, Oxford, pp. 36–75.
A. Wächter (2002), An interior point algorithm for large-scale nonlinear optimization with applications in process engineering, PhD thesis, Department of Chemical Engineering, Carnegie-Mellon University, Pittsburgh, PA, USA.
A. Wächter and L. T. Biegler (2000), 'Failure of global convergence for a class of interior point methods for nonlinear programming', *Math. Program.* **88**(3), 565–574.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press

A. Wächter and L. T. Biegler (2003*a*), Line search filter methods for nonlinear programming: Local convergence, Technical Report RC23033(W0312-090), T. J. Watson Research Center, Yorktown Heights, NY, USA.

A. Wächter and L. T. Biegler (2003*b*), Line search filter methods for nonlinear programming: Motivation and global convergence, Technical Report RC23036(W0304-181), T. J. Watson Research Center, Yorktown Heights, NY, USA.

A. Wächter and L. T. Biegler (2004), On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Research report RC 23149, IBM T. J. Watson Research Center, Yorktown Heights, NY, USA.

M. H. Wright (1992), Interior methods for constrained optimization, in *Acta Numerica*, Vol. 1, Cambridge University Press, pp. 341–407.

S. J. Wright (1997), *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, USA.

H. Yamashita, H. Yabe and T. Tanabe (2004), 'A globally and superlinearly convergent primal-dual interior point trust region method for large scale constrained optimization', *Math. Program.* Online First DOI 10.1007/s10107-004-0508-9.

Y. Ye (1997), *Interior Point Algorithms, Theory and Analysis*, Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley, New York, USA.

E. A. Yildirim and S. J. Wright (2002), 'Warm-start strategies in interior-point methods for linear programming', *SIAM J. Optim.* **12**(3), 782–810.

Y. Yuan, ed. (1998), *Advances in Nonlinear Programming*, Kluwer, Dordrecht, Netherlands.

Y. Yuan (2000), 'On the truncated conjugate-gradient method', *Math. Program., Ser. A* **87**(3), 561–573.

Y. Zhang (1994), 'On the convergence of a class of infeasible interior-point methods for the horizontal linear complementarity problem', *SIAM J. Optim.* **4**(1), 208–227.

C. Zhu, R. H. Byrd, P. Lu and J. Nocedal (1997), 'Algorithm 778. L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization', *ACM Trans. Math. Software* **23**(4), 550–560.

https://doi.org/10.1017/S0962492904000248 Published online by Cambridge University Press