METHODS FOR LARGE-SCALE UNCONSTRAINED OPTIMIZATION

GIOVANNI FASANO
Dipartimento di Matematica Applicata, Università Ca' Foscari, Venezia, Italy

The general large-scale unconstrained optimization problem can be formulated as follows:

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a real-valued objective function, and the parameter n is *large*. Observe that in general the minimization in Equation (1) refers to the search for a *local minimizer* of $f(x)$. Standard assumptions ensure that the problem (1) admits solution. In particular, if the level set $\mathcal{L}_0 = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ is compact and $f(x)$ is continuous on $\mathcal{L}_0$, then by the Weierstrass theorem (1) has a solution. Moreover, if the objective function is coercive, that is, $\lim_{\|x\| \to +\infty} f(x) = +\infty$ (which usually holds in real applications), then all the level sets of $f(x)$ are compact.

The definition of "*large n*" is a little vague for the problem (1), in that it does not specify exactly a range of values and is straightforwardly machine dependent. Indeed, broadly speaking, n may be considered large on a particular machine, if standard optimization techniques for Equation (1) become progressively inadequate when n increases, so that *ad hoc* methods must be adopted. Anyway, 10^6–10^7 may be considered a reference value for n, on a serial machine, in order to consider Equation (1) a large-scale continuous optimization problem. Real-life applications increasingly yield complex models that require efficient techniques (see the Nonlinear Optimization Models collected by Robert Vanderbei on `http://www.orfe.princeton.edu/~ rvdb/ampl/nlmodels`). This has led in the last decades to a growing interest for large-scale optimization, so that the latter topic can be considered by this time a mature field of research.

Optimization methods for small- and medium-scale problems [1–4] are usually hardly adaptable to large-scale problems. Indeed, though they may be still effective in most of the cases, on problems where n is large they often prove not to be efficient enough (i.e., they are unable to solve the problems using a reasonable workload).

As for most of the algorithms for continuous unconstrained optimization, methods for large values of n claim for suitable search directions and stepsizes along them, too. In particular, the choice of the search directions is responsible for the *efficiency* of the methods (i.e., the rate of convergence), while a proper choice of the steplength ensures the *effectiveness*. When n is large, iterative methods typically show a reduced computational burden with respect to direct methods. They generate a sequence of iterates $\{x_k\}$ approaching a solution, which usually follows one of the following patterns:

1. $x_{k+1} = x_k + \alpha_k d_k$ in case convergence to a stationary point x^* is sought, which *satisfies the first-order necessary optimality conditions*;
2. $x_{k+1} = x_k + \phi_d(\alpha_k) d_k + \phi_u(\alpha_k) u_k$ in case convergence to a stationary point x^* is sought, which *satisfies the second order necessary optimality conditions*;

where $d_k \in \mathbb{R}^n$ is a search direction that approximates a Newton-type direction at x_k, $u_k \in \mathbb{R}^n$ is a search direction that takes into account the local curvature of the objective function at x_k, $\phi_d(\cdot)$, and $\phi_u(\cdot)$ are polynomial functions of degree at most two, and α_k is a suitable stepsize. The schemes in the classes (1) and (2) may typically be differentiated on the basis of the information on $f(x)$, which is available at the iterate x_k. In particular, if at x_k we can simply use $f(x_k)$, $\nabla f(x_k)$, and a partial information on $\nabla^2 f(x_k)$, then we will be able to generate $\{x_k\}$ according to 1.

Wiley Encyclopedia of Operations Research and Management Science, edited by James J. Cochran

On the other hand, if in addition we get information on the eigenpair associated with the smallest eigenvalue of the Hessian matrix $\nabla^2 f(x_k)$, we can generate the sequence $\{x_k\}$ according to (2). Observe that the last statement does not imply that a full knowledge (and possibly the storage) of $\nabla^2 f(x_k)$ is required (which is commonly considered cumbersome). When n increases, though in principle the entries of the Hessian matrix could be computed by finite differences, iterative methods commonly need to perform only the product *Hessian* × *vector*, which is computationally cheaper. Alternatively, in case $f(x)$ is explicitly defined and is not a *black box*, automatic differentiation [5] is often adopted to provide information on $\nabla^2 f(x_k)$.

GLOBAL CONVERGENCE AND GRADIENT METHODS

A very general convergence result may be proved for the iterative scheme (1). Indeed, introducing the *forcing function* $\sigma() : \mathbb{R}^+ \to \mathbb{R}^+$, such that $\lim_{k\to\infty} \sigma(w_k) = 0$ implies $\lim_{k\to\infty} w_k = 0$, we have [6]:

Proposition 1. *Let $\{x_k\}$ be the sequence generated by the scheme (1) starting from $x_0 \in \mathbb{R}^n$. Suppose the level set $\mathcal{L}_0$ is compact and for any k we have $d_k \neq 0$ if $\nabla f(x_k) \neq 0$, with $f(x_{k+1}) \leq f(x_k)$. Assume that*

1. *we have*

$$\lim_{k\to\infty} \frac{\nabla f(x_k)^{\mathrm{T}} d_k}{\|d_k\|} = 0,$$

2. *for any k for which $d_k \neq 0$*

$$\frac{|\nabla f(x_k)^{\mathrm{T}} d_k|}{\|d_k\|} \geq \sigma\left(\|\nabla f(x_k)\|\right).$$

Then, either there exists a finite index $\hat{k}$ such that $\nabla f(x_{\hat{k}}) = 0$, or the infinite sequence $\{x_k\}$ satisfies

- *$x_k \in \mathcal{L}_0$ for any k;*
- *the sequence $\{f(x_k)\}$ converges;*
- *for any infinite subsequence $\mathcal{K}$ of indices*

$$\lim_{k\to\infty,\, k\in\mathcal{K}} \|\nabla f(x_k)\| = 0.$$

A very simple class of iterative methods for large-scale unconstrained optimization, which satisfy the hypotheses of Proposition 1, is given by the so-called *gradient methods* [4,7], which may be specified in the form

$$x_{k+1} = x_k - \alpha_k D_k \nabla f(x_k), \tag{2}$$

with $D_k \in \mathbb{R}^{n\times n}$ positive definite for any k. Observe that whenever $\nabla f(x_k) \neq 0$ the search direction $d_k = -D_k \nabla f(x_k)$ in Equation (2) is of descent at the nonstationary point x_k, and satisfies statement (2) of Proposition 1. Moreover, an Armijo-type linesearch procedure [8] should be adopted to compute α_k in Equation (2) in order to satisfy also statement (1) of Proposition 1. Of course, when $D_k = I$, for any k, then the iteration (2) reduces to the standard *steepest descent method*, which is equivalent to minimize the First-order local model of the objective function. The steepest descent method yields a linear convergence rate, and it is not scale invariant under a coordinate transformation. In case the evaluation of $f(x)$ is expensive (e.g., for several simulation problems [9]), a constant value for the stepsize α_k may be adopted in place of the linesearch procedure. On the other hand, a computationally efficient choice for α_k and D_k was suggested in Barzilai and Borwein [10] and Raydan [11], where $D_k = I$, for any k, and

$$\alpha_k = \frac{\|x_k - x_{k-1}\|^2}{(x_k - x_{k-1})^{\mathrm{T}}\left[\nabla f(x_k) - \nabla f(x_{k-1})\right]}.$$

Finally, when $D_k = [\nabla^2 f(x_k)]^{-1}$ and $\alpha_k = 1$ the iteration (2) reduces to *Newton's method*, which corresponds to minimize the local quadratic model of $f(x)$ at x_k. The latter technique is appealing thanks to its quadratic rate of convergence and the scale invariance. However, when $\nabla^2 f(x_k)$ is dense and n is large, it may not be a suitable choice. In the latter case, fully computing $[\nabla^2 f(x_k)]^{-1}$ by means of a Cholesky factorization or a symmetric indefinite factorization requires nearly $1/3n^3$ floating-point operations, while for sparse Hessians the workload is proportional to the sparsity pattern.

INEXACT AND TRUNCATED NEWTON METHODS

The application of Newton's method requires one to compute the direction $d_k = -[\nabla^2 f(x_k)]^{-1}\nabla f(x_k)$, which is equivalent to solving *Newton's equation*

$$\nabla^2 f(x_k)d = -\nabla f(x_k). \tag{3}$$

When the iterate x_k is still "far" from the stationary point x^*, an accurate solution of Equation (3) may be unjustified; thus, when n is large the solution of Equation (3) may be obtained by an iterative procedure. In addition, in order to preserve the global convergence of Newton's method, a suitable choice of α_k must be computed (a globalization technique), either based on a *linesearch approach* [1,12,13] or a *trust-region approach* [14,15]. On these guidelines, for large values of n, *inexact Newton method* has been proposed in the literature [16]. The underpinning idea of these schemes is that we can balance the computational burden of solving Equation (3) and the accuracy of the solution obtained. In particular, from Dembo *et al*. [16], if the approximate solution d_k of Equation (3) is computed, such that (truncation rule)

$$\lim_{k\to\infty} \frac{\|\nabla^2 f(x_k)d_k + \nabla f(x_k)\|}{\|\nabla f(x_k)\|} = 0 \tag{4}$$

holds, then the sequence of iterates $\{x_k\}$ is *superlinearly convergent* to a stationary point x^*. Broadly speaking, the condition (4) states that when k increases, the gradient of the quadratic local model of $f(x)$ at x_k approaches zero "more quickly" than the gradient $\nabla f(x_k)$ of the objective function. When n is large, iterative methods such as the *conjugate gradient* (CG) may be used to compute an approximate solution d_k of Equation (3), such that Equation (4) holds. The latter philosophy is based on *truncated Newton methods* [17–19], which proved to be a very efficient tool [20,21]. An example of truncated Newton method, based on a linesearch approach to ensure the global convergence, is the following:

Linesearch-based Truncated Newton scheme

```
Set $x_0 \in \mathbb{R}^n$
Set $\eta_k \in [0, 1)$ for any $k$, with $\{\eta_k\} \to 0$
OUTER ITERATIONS
for $k = 0, 1, \ldots$
    Compute $\nabla f(x_k)$; if $\|\nabla f(x_k)\|$ is small then STOP
        INNER ITERATIONS
        Compute $d_k$ which approximately solves Equation (3)
        and satisfies the truncation rule
            $\|\nabla^2 f(x_k)d_k + \nabla f(x_k)\| \le \eta_k \|\nabla f(x_k)\|$
    Compute the steplength $\alpha_k$ by an Armijo-type linesearch scheme
    Update $x_{k+1} = x_k + \alpha_k d_k$
endfor
```

Observe that at the outer iteration k a second-order expansion of the function $f(x)$ is considered, whose stationary point (if any) is detected by solving Equation (3). The number of inner iterations, which are necessary to approximately solve Newton's equation (3), depends on the current parameter η_k. Thus, a fine (and more expensive) solution of Equation (3) is required only when $k \to \infty$. The truncation rule (4) is often replaced in practice by more efficient conditions [22,23].

The Hessian matrix in Equation (3) may be possibly indefinite. Thus, additional safeguard is required for the choice of the iterative solver, since for instance a negative curvature direction for function $f(x)$ at x_k may be detected, or a pivot breakdown may occur with the CG method. The Lanczos process and suitable extensions of the CG method (namely, the planar CG) may successfully be used as alternatives to the CG [24–28]. Furthermore, finite differences have also been used in the past, in order to approximately compute the information on the Hessian matrix, within a truncated Newton method (see TNPACK in Schlick and Fogelson [29]).

Three other relevant issues have been studied in the last two decades, in order to drastically improve the performance of truncated Newton methods: the *preconditioning strategies* to solve Newton's equation, the exploitation of *negative curvature directions* of the objective function, and the introduction of *nonmonotone globalization* schemes. The first issue studies efficient preconditioners of the matrix $\nabla^2 f(x_k)$, for the specific case of large n, when the Hessian is either positive definite or indefinite [14,30–32]. The second issue needs a specific treatment when n is large, since it deals with the careful choice for the iterative solver of Newton's equation. Moreover, an accurate exploration of the negative curvatures of $f(x)$ is essential in order to prove the convergence of truncated methods toward solutions that satisfy the second-order optimality conditions [33,34]. The latter task is pursued by computing the pair of promising search directions (d_k, u_k) at the outer iteration k [20,25, 35–37], by means of efficient iterative techniques. Roughly speaking, d_k contains information on the local convexity of $f(x)$ at x_k, while u_k conveys information on the local nonconvexity of $f(x)$, and approximates the eigenvector corresponding to the minimum negative eigenvalue of $\nabla^2 f(x_k)$. Then, in order to guarantee the convergence to second-order points, the two directions may be combined to produce the next iterate, as in (see pattern (2) mentioned in the introductory text.)

$$x_{k+1} = x_k + \phi_d(\alpha_k)d_k + \phi_u(\alpha_k)u_k,$$

where α_k is chosen by a proper linesearch procedure. The introduction of nonmonotone globalization schemes, in truncated Newton methods, was first proposed in Grippo *et al*. [38] (see also Grippo *et al*. [13], Ferris *et al*. [35], and Lucidi *et al*. [25], Ferris *et al*. [35]). Considering the standard Armijo linesearch scheme $f(x_k + \alpha d_k) \le f(x_k) + \gamma\alpha\nabla f(x_k)^{\mathrm{T}} d_k$, where $\gamma \in (0, 1/2)$, the main idea behind a nonmonotone approach generalizes the monotone decreasing of $f(x_k)$, as in the scheme

$$f(x_k + \alpha d_k) \le \max_{0\le i\le M}\{f(x_{k-i})\} + \gamma\alpha\nabla f(x_k)^{\mathrm{T}} d_k,$$

where the integer M indicates the "memory" of the scheme. The nonmonotone approach proves to be particularly efficient in the case of highly nonlinear functions, where enforcing the monotonic decrease of $f(x)$ may cause the algorithms to be trapped within narrow valleys.

Global convergence of truncated Newton methods may be proved either using a linesearch framework, or adopting a *trust-region* approach [14]. A trust-region scheme provides the new iterate $x_{k+1} = x_k + d_k$ at step k (i.e., now d_k includes both the search direction and the steplength), by solving the constrained subproblem

$$\begin{aligned} \min_d \quad & Q_k(d) \\ \text{s.t.} \quad & \|\Lambda_k d\| \le \Delta_k, \end{aligned} \tag{5}$$

where $Q_k(d) = f(x_k) + \nabla f(x_k)^{\mathrm{T}} d + \frac{1}{2} d^{\mathrm{T}} \nabla^2 f(x_k) d$ and Λ_k is a scaling matrix (possibly $\Lambda_k = I$, for any k), which may be regarded as introducing an implicit preconditioning. The basic trust-region approach adaptively assesses the trust-region radius Δ_k in Equation (5) with the following idea: as long as the solution d_k of Equation (5) yields a value for ρ_k close to 1, where

$$\rho_k = \frac{f(x_k) - f(x_k + d_k)}{Q_k(0) - Q_k(d_k)}, \tag{6}$$

$Q_k(d)$ is a "trusted" local model of $f(x)$ and the radius Δ_{k+1} may be possibly larger than Δ_k. On the contrary, if ρ_k in Equation (6) is relatively small, then the model $Q_k(d)$ is not reliable and the radius Δ_{k+1} at the next iteration satisfies $\Delta_{k+1} < \Delta_k$.

When n is large, the trust-region approach requires both a strategy to assess the parameter Δ_k and an efficient procedure to approximately solve Equation (5). Among the first techniques to solve Equation (5), the *dogleg* schemes proposed to investigate the solution over a suitable one-dimensional arc [39,40].

In Byrd *et al.* [41], a pair of feasible directions for Equation (5) is preliminarily computed, then the minimization of $Q_k(d)$ is performed over the two-dimensional manifold associated with the latter vectors. Another approach to solve Equation (5) imposes the KKT conditions of Equation (5), Q1 and solves a resulting perturbation of the Newton equation for $f(x)$ [42,43].

Since the previous approaches may recur to direct solvers (e.g., Cholesky factorization), when n is large, a different strategy is suggested in Steihaug [44] and Toint [45]. Here, CG is used to minimize $f(x)$; then, a piecewise path connecting the feasible iterates generated by CG is considered, and can be followed up to the boundary of the trust region. Alternatively, both the steepest descent direction (moving to the Cauchy point[1] on the boundary of the trust region) and possibly a negative curvature direction (in case $\nabla^2 f(x_k)$ is indefinite) are investigated. An example of a robust and reliable truncated Newton method, in a trust region rather than a linesearch framework, is given by the package LANCELOT [21]. Trust-region methods have several advantages, including the fact that they show strong convergence properties [14]. However, in case for several values of k the Hessian matrix $\nabla^2 f(x_k)$ is indefinite, these methods may be inefficient, inasmuch as they can stop the inner iterations prematurely, when approximately minimizing $Q_k(d)$. These drawbacks have been addressed and partially overcome in Gould *et al.* [26], by using the Lanczos process for the solution of large symmetric and indefinite linear systems, and exploiting its relation with the CG (see also Stoer [46]).

NONLINEAR CONJUGATE GRADIENT METHODS

The CG used for the minimization of a strictly convex quadratic function has been extended to the minimization of nonlinear functions, by suitably modifying the computation of its coefficients [47–49]. In particular, suppose that at step k the directions $\{p_1, \ldots, p_k\}$ were generated by the Nonlinear Conjugate Gradient (NCG). Then, the steplength α_k is chosen to satisfy the standard first Wolfe linesearch condition (minimization along the direction p_k)

$$f(x_k + \alpha_k p_k) \le f(x_k) + \gamma \alpha_k \nabla f(x_k)^{\mathrm{T}} p_k, \qquad \gamma \in (0, 1/2) \tag{7}$$

and either one of the following additional relations (Wolfe conditions)

$$\nabla f(x_k + \alpha_k p_k)^{\mathrm{T}} p_k \ge \theta \nabla f(x_k)^{\mathrm{T}} p_k, \qquad \theta \in (\gamma, 1) \tag{8}$$

$$|\nabla f(x_k + \alpha_k p_k)^{\mathrm{T}} p_k| \le \theta |\nabla f(x_k)^{\mathrm{T}} p_k|, \qquad \theta \in (\gamma, 1) \tag{9}$$

(theoretical reasons may impose the more restrictive condition $\theta \in (\gamma, 1/2)$). Furthermore, in the NCG, it is possible to adopt different formulae for the computation of the coefficient β_k used in the iteration $p_k = -\nabla f(x_k) + \beta_k p_{k-1}$. Very standard formulae for β_k in the literature are [48]

$$\begin{aligned}
&\text{Fletcher–Reeves:} \\
&\qquad \frac{\|\nabla f(x_k)\|^2}{\|\nabla f(x_{k-1})\|^2}, \\
&\text{Polak–Ribiere:} \\
&\qquad \frac{\left[\nabla f(x_k) - \nabla f(x_{k-1})\right]^{\mathrm{T}} \nabla f(x_k)}{\|\nabla f(x_{k-1})\|^2}, \\
&\text{Hestenes–Stiefel:} \\
&\qquad \frac{\left[\nabla f(x_k) - \nabla f(x_{k-1})\right]^{\mathrm{T}} \nabla f(x_k)}{\left[\nabla f(x_k) - \nabla f(x_{k-1})\right]^{\mathrm{T}} p_{k-1}}.
\end{aligned} \tag{10}$$

We report below a general scheme for the NCG. We remark that, apart from Equation (10), several other choices for the coefficient β_k have been studied in the literature [48].

[1] The Cauchy point is the minimizer of $Q_k(d)$ along the direction $-\nabla f(x_k)$, subject to the trust-region bound.

Nonlinear Conjugate Gradient (NCG) method

```
Step 0: Choose $x_0 \in \mathbb{R}^n$, set $k = 0$
Step 1: Compute $\nabla f(x_k)$. If $\nabla f(x_k) = 0$ then STOP
Step 2: If $k \neq 0$ compute $\beta_k$ as in Equation (10). Compute the direction
    $p_k = \begin{cases} -\nabla f(x_k) & k = 0 \\ -\nabla f(x_k) + \beta_k p_{k-1} & k \geq 1 \end{cases}$
Step 3: Choose $\alpha_k$ such that Equation (7) is satisfied, along with either Equation (8) or (9)
Step 4: Set $x_{k+1} = x_k + \alpha_k p_k$, $k = k + 1$ and go to Step 1
```

Unlike the CG, the NCG does not generally retain global convergence properties, because of the loss of conjugacy caused by the nonlinearity of $f(x)$. In order to enforce global convergence, a periodic restart may be imposed, either when $k > n$ or if a test on the conjugacy loss, like

$$|\nabla f(x_k)^{\mathrm{T}} \nabla f(x_{k-1})| > \sigma \|\nabla f(x_{k-1})\|^2, \quad \sigma \in (0, 1)$$

is satisfied. However, when n is large, the latter choice is impracticable (very rarely adopted) and proved to be inefficient. Using the formula of β_k, proposed by Fletcher–Reeves, in case α_k is computed by an exact linesearch procedure [50], or by an Armijo-type linesearch based on the strong Wolfe condition (9) with $\theta < 1/2$ [2,51], the global converge is retained and $\liminf_{k\to\infty} \|\nabla f(x_k)\| = 0$. In Gilbert and Nocedal [49], the global convergence was also proved for the case $|\beta_k| \leq \beta_k^{FR}$, where β_k^{FR} refers to the formula proposed by Fletcher and Reeves.

Though the choice of β_k^{PR} made by Polak–Ribiere proved to be computationally more efficient than β_k^{FR} [2,48], convergence properties for the NCG with β_k^{PR} have always been more difficult to investigate. In particular, the global convergence was proved under strong convexity assumption of $f(x)$, using $\beta_k = \max\{0, \beta_k^{PR}\}$, with an inexact linesearch procedure which also satisfies Equation (9). Only recently [52], by using a more sophisticated linesearch procedure, the global convergence of NCG using β_k^{PR} was proved (in the sense that $\lim_{k\to\infty} \|\nabla f(x_k)\| = 0$), without the strong convexity assumption on $f(x)$. An extension of the NCG, which uses the Polak–Ribiere value β_k^{PR}, is implemented in the routine VA14, of the Harwell Subroutine Library [53].

QUASI-NEWTON METHODS AND PARTIALLY SEPARABLE FUNCTIONS

Quasi-Newton methods are a powerful tool for large-scale optimization. They are used to solve Newton's equation, without using the second-order derivatives. In Particular, these methods generate the sequence $\{x_k\}$ by iteratively solving the modified Newton's equation

$$B_k d = -\nabla f(x_k), \quad \text{with } B_k \text{ positive definite}$$

and computing $x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f(x_k)$, where B_k in some sense *approximates* the Hessian matrix $\nabla^2 f(x_k)$ of $f(x)$. More specifically, B_{k+1} is the solution of the subproblem

$$\begin{aligned} \min_B \quad & \|B - B_k\|_F \\ \text{s.t.} \quad & B = B^{\mathrm{T}} \\ & Bs_k = y_k, \end{aligned}$$

where $\|\cdot\|_F$ indicates the Frobenius norm, $s_k = x_{k+1} - x_k$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$, so that no second-order derivatives are required. Several quasi-Newton schemes have been developed, using approximations of either $\nabla^2 f(x_k)$ or $[\nabla^2 f(x_k)]^{-1}$ [2,12,48,54,55]. One of the most effective, in small-scale and medium-scale optimization, is BFGS (*Broyden–Fletcher–Goldfarb–Shanno*). Here, a sequence $\{H_k\}$ of approximations of $[\nabla^2 f(x_k)]^{-1}$ is generated, with H_k being positive definite, by solving the subproblem

$$\begin{array}{rl} \min_H & \|H - H_k\|_F \\ \text{s.t.} & H = H^{\mathrm{T}} \\ & s_k = H y_k, \end{array}$$

so that $x_{k+1} = x_k - \alpha_k H_k \nabla f(x_k)$. Moreover, the global convergence properties can be proved by computing the steplength α_k via an Armijo-type linesearch procedure, satisfying also Equation (9). It can be easily proved [2] that the matrices $\{H_k\}$ satisfy the relation

$$H_{k+1} = V_k^{\mathrm{T}} H_k V_k + \rho_k s_k s_k^{\mathrm{T}},$$

where $V_k = I - \rho_k y_k s_k^{\mathrm{T}}$ and $\rho_k = 1/y_k^{\mathrm{T}} s_k$, so that the pairs $\{(s_k, y_k)\}$ are necessary and sufficient to generate the whole sequence $\{H_k\}$. We remark that an interesting relationship between NCG and BFGS can be stated (see Nocedal and Wright [2] or Pytlak [48]), both for quadratic and general nonlinear functions.

In a large-scale setting, the storage of V_k cannot be proposed; Thus, the formula to update H_{k+1} is suitably modified by using just the most *m recent* pairs (s_i, y_i), $i = 1, \dots, m$. On this guideline, the matrix H_k is computed as

$$\begin{aligned} H_k = {} & (V_{k-1}^{\mathrm{T}} \dots V_{k-m}^{\mathrm{T}}) H_k^0 (V_{k-m} \dots V_{k-1}) \\ & + \rho_{k-m} (V_{k-1}^{\mathrm{T}} \cdots V_{k-m+1}^{\mathrm{T}}) \\ & \quad s_{k-m} s_{k-m}^{\mathrm{T}} (V_{k-m+1} \dots V_{k-1}) \\ & + \rho_{k-m+1} (V_{k-1}^{\mathrm{T}} \cdots V_{k-m+2}^{\mathrm{T}}) \\ & \quad s_{k-m+1} s_{k-m+1}^{\mathrm{T}} (V_{k-m+2} \dots V_{k-1}) \\ & + \cdots \\ & + \rho_{k-1} s_{k-1} s_{k-1}^{\mathrm{T}}, \end{aligned}$$

and a practical choice of H_k^0 is often $H_k^0 = \gamma_k I$, with $\gamma_k = s_{k-1}^{\mathrm{T}} y_{k-1} / y_{k-1}^{\mathrm{T}} y_{k-1}$. The storage of H_k now requires only the pairs (s_i, y_i), $i \le m$, so that it can be defined by setting the parameter m. The resulting new sequence $\{H_k\}$ yields a very efficient method called *L-BFGS* [56–58] (where "L" stands for *limited memory*).

Observe that in practice very small values of m are used (say $3 \le m \le 8$) and the L-BFGS method often may be competitive with truncated Newton methods or the NCG [59]. In addition, thanks to the simplified structure of H_k, iterative Krylov-based methods may be easily applied when performing the product $H_k \nabla f(x_k)$ in the iteration $x_{k+1} = x_k - \alpha_k H_k \nabla f(x_k)$. In order to cope with problems of illconditioning of the Hessian matrix, a combined approach between L-BFGS and the discrete Newton method was studied in Byrd *et al*. [60]. The L-BFGS method was coded and is available in the routine `VA15`, of the Harwell Subroutine Library [53]. The memory storage of the NCG implemented in the routine `VA14` is $6n$, while the memory storage of the L-BFGS method implemented in the routine `VA15` is $2nm + 4n$.

When the scale n of problem (1) is large, also another strategy may be adopted to tackle a minimum point: namely, *partially separable functions* can be exploited. Suppose $f(x)$ is a partially separable function, that is, it can be written in the form

$$f(x) = \sum_{i=1}^{p} f_i(x), \tag{11}$$

where each of the functions $f_1(x), \dots, f_p(x)$ depends just on a subset of the unknowns x. Evidently, by Equation (11) the gradient and the Hessian matrix of $f(x)$ are given by

$$\nabla f(x) = \sum_{i=1}^{p} \nabla f_i(x); \quad \nabla^2 f(x) = \sum_{i=1}^{p} \nabla^2 f_i(x).$$

Thus, drawing our inspiration from the quasi-Newton methods, let B be a quasi-Newton approximation of $\nabla^2 f(x)$, while B_i, $i = 1, \dots, p$, is a quasi-Newton approximation of $\nabla^2 f_i(x)$. Then, numerical experiences proved that $\sum_{i=1}^{p} B_i$ is often a better approximation of $\nabla^2 f(x)$ than B, as long as a satisfactory quasi-Newton approximation of each $\nabla^2 f_i(x)$ is given.

The AMPL modeling language [61] contains a specific tool, which is able to compute the approximation B_i of $\nabla^2 f_i(x)$ by automatic differentiation [5], so that a sparse representation of $\nabla^2 f(x)$ is available. For other references on this topic see also Griewank and Toint [62,63] and Malmedy Toint [64] and the references therein.

MULTIGRID OPTIMIZATION

Optimization-based multigrid methods are a class of algorithms which have had a fast development in the last decade [65–67], since efficient methods for the optimization of large-scale nonlinear systems governed by differential equations are sought. In particular, let us consider the problem

$$\begin{array}{ll} \min_x & f[x, u(x)] \\ \text{s.t.} & S[x, u(x)] = 0, \end{array} \qquad (12)$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, and the variables $u = u(x)$, with $u : \mathbb{R}^n \to \mathbb{R}^m$, are commonly called *state unknowns* and are implicitly defined by the system of partial differential equations (PDE) $S[x, u(x)] = 0$ (*state equations*).

Optimization-based multigrid methods for solving Equation (12) consider a family of optimization problems, associated with Equation (12), each corresponding to a different discretization of the system $S[x, u(x)] = 0$. Intuitively speaking, the workload to solve Equation (12) by choosing a finer discretization (*grid*) is larger than that by choosing a coarser grid. These methods use the computation on a coarser grid (which implies the solution of unconstrained optimization subproblems), in order to improve an approximate solution of problem (12) on a finer grid.

A very general algorithm for the latter purpose is MG/Opt, which is described in Nash [66]. This algorithm does not apply to a specific family of grids; moreover, it relies on the use of optimization models along with a linesearch procedure, in order to guarantee the global convergence of the method. Further details and a numerical experience for multigrid methods are given in Lewis and Nash [67] and the references therein.

Similar ideas for unconstrained optimization problems have been investigated in Toint *et al*. [68] and Gratton *et al*. [69], in order to solve very large-scale trust-region subproblems. Here, some convergence issues related to solutions satisfying both first- and second-order optimality conditions were studied.

REFERENCES

1. Fletcher R. An overview of unconstrained optimization. In: Spedicato E, editor. Algorithms for continuous optimization. The state of the art. Kluwer Academic Publishers; 1994.pp. 109–143. Q2
2. Nocedal J, Wright S. Numerical optimization. 2nd ed. Springer series in operations research and financial engineering. New York: Springer; 2006.
3. Nocedal J. Large scale unconstrained optimization. In: Watson GA, Duff I, editors. The state of the art in numerical analysis. Oxford: Oxford University Press; 1997. pp. 311–338.
4. Bertsekas DP. Nonlinear programming. 2nd ed.. Belmont (MA): Athena Scientific; 1999.
5. Griewank A. Automatic differentiation. Philadelphia: SIAM; 2000.
6. Orthega JM. Introduction to parallel and vector solution of linear systems (frontiers In Computer Science). Plenum Press; 1988.
7. Bertsekas DP. Constrained optimization and Lagrange multiplier methods. Athena Scientific; 1996.
8. Armijo L. Minimization of functions having continuous partial derivatives. Pacific J Math 1966;3:1–3.
9. Alexandrov NM, Hussaini MY, editors. Multidisciplinary design optimization - state of the art. In: Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization. SIAM Proceedings Series. 1997; Q3
10. Barzilai J, Borwein JM. Two point step size gradient method. IMA J Numer Anal 1988;8:141–148.
11. Raydan M. The Barzilai and Borwein gradient method for large scale unconstrained minimization problems. SIAM J Optim 1997;7:26–33.
12. Fletcher R. Practical methods of optimization. New York: John Wiley & Sons; 1987.
13. Grippo L, Lampariello F, Lucidi S. A class of nonmonotone stabilization methods in unconstrained optimization. Numer Math 1991;59:779–805.
14. Conn AR, Gould NIM, Toint PhL. Trust region methods. Philadelphia: SIAM; 2000.

15. Shultz GA, Schnabel RB, Byrd RH. A family of trust-region-based algorithms for unconstrained minimization. SIAM J Numer Anal 1985;22:47–67.
16. Dembo R, Eisenstat S, Steihaug T. Inexact Newton methods. SIAM J Numer Anal 1982;19:400–408.
17. Dembo RS, Steihaug T. Truncated-Newton algorithms for large-scale unconstrained optimization. Math Program 1983;26:190–212.
18. Nash S. A survey of Truncated-Newton methods. J Comput Appl Math 2000;124:45–59.
19. Dembo R, Steihaug T. Truncated-Newton algorithms for large-scale unconstrained optimization. Math Program 1983;26:190–212.
20. Gould NIM, Lucidi S, Roma M, *et al*. Exploiting negative curvature directions in linesearch methods for unconstrained optimization. Optim Methods Softw 2000;14:75–98.
21. Conn AR, Gould NIM, Toint PhL. LANCELOT: a Fortran package for large-scale nonlinear optimization (Release A). Heidelberg, Berlin: Springer; 1992.
22. Nash S, Sofer A. Assessing a search direction within a Truncated-Newton method. Oper Res Lett 1990;9:219–221.
23. Fasano G, Lucidi S. A nonmonotone truncated Newton-Krylov method exploiting negative curvature directions, for large scale unconstrained optimization. Optim Lett 2009; 3:521–535.
24. Nash S. Newton-type minimization via the Lanczos method. SIAM J Numer Anal 1984;21:770–788.
25. Lucidi S, Rochetich F, Roma M. Curvilinear stabilization techniques for Truncated Newton methods in large scale unconstrained Optimization. SIAM J Optim 1998;8:916–939.
26. Gould NIM, Lucidi S, Roma M, *et al*. Solving the trust-region subproblem using the Lanczos method. SIAM J Optim 1999;9:504–525.
27. Fasano G. Planar-conjugate gradient algorithm for large-scale unconstrained optimization, part 1: theory. J Optim Theory Appl 2005;125:523–541.
28. Fasano G. Planar-conjugate gradient algorithm for large-scale unconstrained optimization, part 2: application. J Optim Theory Appl 2005;125:543–558.
29. Schlick T, Fogelson A. TNPACK - a truncated Newton package for large-scale problems: I. Algorithm and usage. ACM Trans Math Softw 1992;18:46–70.
30. Nash S. Preconditioning of truncated-Newton methods. SIAM J Sci Stat Comput 1985;6:599–616.
31. Morales JL, Nocedal J. Automatic preconditioning by limited memory quasi-Newton updating. SIAM J Optim 2000;10:1079–1096.
32. Roma M. A dynamic scaling based preconditioning for truncated Newton methods in large scale unconstrained optimization. Optim Methods Softw 2005;20:693–713.
33. Moré JJ, Sorensen DC. On the use of directions of negative curvature in a modified Newton method. Math Program 1979;16:1–20.
34. Fasano G, Roma M. Iterative computation of negative curvature directions in large scale optimization. Comput Optim Appl 2007;38:81–104.
35. Ferris MC, Lucidi S, Roma M. Nonmonotone curvilinear linesearch methods for unconstrained optimization. Comput Optim Appl 1996;6:117–136.
36. Goldfarb D. Curvilinear path steplength algorithms for minimization which use directions of negative curvature. Math Program 1980;18:31–40.
37. Olivares A, Moguerza JM, Prieto FJ. Nonconvex optimization using negative curvature within a modified linesearch. Eur J Oper Res 2008;189:706–722.
38. Grippo L, Lampariello F, Lucidi S. A truncated Newton method with nonmonotone linesearch for unconstrained optimization. J Optim Theory Appl 1989;60:401–419.
39. Powell MJD. A new algorithm for unconstrained optimization. In: Mangasarian O, Ritter K, editors. Nonlinear programming. New York: Academic Press; 1970. pp. 31–65.
40. Dennis J, Mei H. Two new unconstrained optimization algorithms which use function and gradient values. J Optim Theory Appl 1979;28:453–482.
41. Byrd R, Schnabel R, Schultz G. An approximate solution of the trust region problem by minimization over two-dimensional subspaces. Math Program 1988;40:247–263.
42. Sorensen DC. Newton's method with a model trust-region modification. SIAM J Sci Stat Comput 1982;19:409–427.
43. Moré JJ, Sorensen DC. Computing a trust region step. SIAM J Sci Stat Comput 1983;4:553–572.
44. Steihaug T. The conjugate gradient method and trust regions in large-scale optimization. SIAM J Numer Anal 1983;20:626–637.

45. Toint PhL. Towards an efficient sparsity exploiting Newton method for minimization. In: Duff IS, editor. Sparse matrices and their uses. London: Academic Press; 1981. pp. 57–88.
46. Stoer J. Solution of large linear systems of equations by conjugate gradient type methods. In: Bachem A, Grotschel M, Korte B, editors. Mathematical programming. The state of the art. Berlin/Heidelberg: Springer; 1983. pp. 504–565.
47. Hestenes M. Conjugate direction methods in optimization. New York: Springer; 1980.
48. Pytlak R. Conjugate gradient algorithms in nonconvex optimization. Berlin/Heidelberg: Springer; 2009.
49. Gilbert J, Nocedal J. Global convergence properties of conjugate gradient methods for optimization. SIAM J Optim 1992;2:21–42.
50. Zoutendijk G. Nonlinear programming computational methods. In: Abadie J, editor. Integer and nonlinear programming. Amsterdam: North-Holland Publishing Co.; 1970. pp. 37–86.
51. Al-Baali M. Descent property and global converge of the Flether-Reeves method with inexact line search. IMA J Numer Anal 1985;5:121–124.
52. Grippo L, Lucidi S. A globally convergent version of the Polak-Ribiere conjugate gradient method. Math Program 1997;78:375–391.
53. Harwell Subroutine Library. A catalogue of subroutines. Harwell, England: AEA Technology; 1998.
54. Davidon WC. Variable metric method for minimization. SIAM J Optim 1991;1:1–17.
55. Broyden CG. The convergence of a class of double-rank minimization algorithms, part I and II. J Inst Math Appl 1970;6:76–90, 222–231.
56. Nocedal J. Updating Quasi-Newton matrices with limited storage. Math Comput 1980;35:773–782.
57. Al-Baali M. Extra-updates criterion for limited memory BFGS algorithm for large scale nonlinear optimization. J Complex 2002;18:557–572.
58. Burdakov OP, Martinez JM, Pilotta EA. A limited-memory multipoint symmetric secant method for bound constrained optimization. Ann Oper Res 2002;117:51–70.
59. Nash S, Nocedal J. A numerical study of a limited memory BFGS method and the truncated Newton method for large scale optimization. SIAM J Optim 1991;1:358–372.
60. Byrd R, Nocedal J, Zhu C. Towards a discrete Newton method with memory for large-scale optimization. In: Di Pillo G, Giannessi F, editors. Nonlinear optimization and applications. Plenum Publishing; 1995. pp. 1–12.
61. Fourer R, Gay DM, Kernighan BW. AMPL: a modeling language for mathematical programming. South San Francisco (CA): The Scientific Press; 1993.
62. Griewank A, Toint PhL. Partitioned variable metric updates for large structured optimization problems. Numer Math 1982;39:119–137.
63. Griewank A, Toint PhL. Local convergence analysis of partitioned quasi-Newton updates. Numer Math 1982;39:429–448.
64. Malmedy V, Toint PhL. Approximating Hessians in multilevel unconstrained optimization. Report 08/19. FUNDP - University of Namur; 2008.
65. Lewis RM, Nash SG. A multigrid approach to the optimization of systems governed by differential equations. AIAA Paper 2000–4890. Reston (VA): American Institute of Aeronautics and Astronautics; 2000.
66. Nash SG. A multigrid approach to discretized optimization problems. J Comput Appl Math 2000;14:99–116.
67. Lewis RM, Nash SG. Model problems for the multigrid optimization of systems governed by differential equations. SIAM J Sci Comput 2005;26:1811–1837.
68. Toint PhL, Tomanos D, Weber-Mendonca M. A multilevel algorithm for solving the trust-region subproblem. Optim Methods Softw 2009;24:299–311.
69. Gratton S, Mouffe M, Sartenaer A, *et al*. Numerical experience with a recursive trust-region method for multilevel nonlinear optimization. Optim Methods Softw 2010;25:359–386.

Queries in Article eorms0521

Q1. Please confirm if the abbreviations “KKT” and “AMPL” need to be spelt out. If yes, please provide the expansions.

Q2. Please provide the place of publication for references 1, 6, 7 and 60.

Q3. Please provide the place of publication and publishers name for reference 9.

Q4. Please note that two additional keywords have been added to the list provided by you, as the minimum number keywords required by this article is five. Please confirm if the suggested keywords are fine.

Please note that the abstract and keywords will not be included in the printed book, but are required for the online presentation of this book which will be published on Wiley's own online publishing platform.

If the abstract and keywords are not present below, please take this opportunity to add them now.
The abstract should be a short paragraph upto 200 words in length and keywords between 5 to 10 words.

Abstract: In this article, we review methods for the solution of unconstrained optimization problems, where the number of unknowns is large. We first describe the basics of unconstrained optimization, then we consider the iterative methods that are commonly used within large-scale optimization. The techniques described here explicitly use information on the objective function and some of its derivatives. Extensions to effective quasi-Newton methods for partially separable optimization are detailed.

Q4 **Keywords**: unconstrained Optimization; nonconvex large-scale problems; iterative methods; Newton methods; grid