CENTRO DE MATEMÁTICA E APLICAÇÕES FUNDAMENTAIS
PrePrint Pre–2011–001
http://cmaf.ptmat.fc.ul.pt/preprints.html

A gradient-type algorithm for optimization with constraints

Cristian Barbarosie[1] and **Sérgio Lopes**[1,2]

[1]Centro de Matemática e Aplicações Fundamentais, Universidade de Lisboa
Av. Prof. Gama Pinto, 2, 1649-003 Lisboa, Portugal
e-mail: {barbaros,slopes}@ptmat.fc.ul.pt

[2]Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa
Rua Conselheiro Emídio Navarro, 1, 1959-007 Lisboa, Portugal
e-mail: slopes@deea.isel.pt

February 15, 2011

Abstract

An algorithm is described for solving minimization problems with equality constraints; this algorithm was proposed by one of the authors in a previous paper. A local convergence result is proven. No convexity assumptions are made: the algorithm works for a fully nonlinear setting. The algorithm is then extended, by means of an active set strategy, to account for inequality constraints as well. Some numerical tests are presented.

Keywords: nonlinear programming, constrained minimization.

1 Introduction

We propose a numerical method for the minimization (or maximization) of a functional, subject to constraints. This method has already been used by one of the authors (see [1]); in the present paper, the convergence of the algorithm is proven and an extension to inequality constraints is proposed. Acknowledging this is a heavily studied field, with the contribution of several different authors, we nevertheless feel that some of the most popular methods nowadays (for instance, interior point methods) do not seem very natural and depend on rather heavy computational procedures (tipically the issue of feasibility, even when its violation does not render the problem ill-posed). Our methodology can be regarded as a gradient method applied in the direction tangent to the manifold determined by the constraints, together with a Newton method applied in the orthogonal direction. This method, although not very fast (it has linear convergence) is quite natural, easy to implement, and has the advantage of requiring solely the first derivatives of the objective and of the constraint functions. The idea behind the algorithm resembles the one proposed in [2], except for all the aspects surrounding the penalty function approach – which we do not adopt.

Many algorithms for nonlinear optimization problems seek only a local solution, a feasible point at which the objective function is smaller than at all other nearby feasible points. They do not always find a *global solution*, which is a point with the lowest function value among all

feasible points. Global solutions are needed in some applications, but for many problems they are difficult to recognize and even more difficult to locate. General nonlinear problems, both constrained and unconstrained, may possess local solutions that are not global ones. It should be noted, however, that global optimization algorithms usually require the solution of many local optimization problems. The present text focuses on local solutions.

The following notation will be used: $x \in \mathbb{R}^n$ is the vector of *variables* (also called *unknowns* or *parameters*); f is the *objective function*, a scalar function of x that we want to minimize or maximize; the *constraints* will be modelled by a vector function $g : \mathbb{R}^n \to \mathbb{R}^m$. The Jacobian matrix of a vector function g will be denoted by Dg while its transpose will be denoted by ∇g. In particular, for a scalar function f, ∇f will be the usual gradient. The Hessian matrix of f will be denoted by $D^2 f$.

2 Unconstrained optimization

Unconstrained optimization is the minimization or maximization of a scalar function f defined on the whole $\mathbb{R}^n$, or on an open subset of $\mathbb{R}^n$. Well known optimality conditions are:

Theorem 2.1. *Assume f is a continuously differentiable function having a local minimum point at $x^* \in \mathbb{R}^n$. Then $\nabla f(x^*) = 0$.*

Theorem 2.2. *Assume f is a twice continuously differentiable function, and let $x^* \in \mathbb{R}^n$ be a stationary point, i.e., such that $\nabla f(x^*) = 0$. If $D^2 f(x^*)$ is positive definite, then x^* is a (strict) local minimum point of f.*

Minimization algorithms require the user to supply a starting point, which will be denoted by x_0. Then, at each iteration, the algorithm chooses a direction δ_k and searches along this direction, from the current iterate x_k, for a new iterate with a lower function value. The distance to move along δ_k, the *step length*, is commonly chosen after a finite number of trial step lengths; this strategy is known as *line search* . This kind of procedure is useful for obtaining convergence from "remote" initial approximations x_0, which is not our main concern. Besides, once a locally convergent algorithm has been devised, with the step length taken to be constant throughout, one can always modify it to encompass line search in order to enhance its convergence properties (this is usually the order things are done anyway).

It is quite natural to look for a *descent direction*, that is, a direction δ_k such that $\langle \nabla f(x_k), \delta_k \rangle < 0$, where $\langle\ ,\ \rangle$ denotes the usual dot product in $\mathbb{R}^n$. The *steepest descent direction* $\delta_k = -\nabla f(x_k)$ is the most obvious choice. This *steepest descent method* has the advantage of requiring the calculation of first derivatives only, but it can be quite slow.

The following standard results will be used (they can be easily found in textbooks on Functional Analysis):

Theorem 2.3 (**Banach fixed-point theorem**). *Assume that K is a nonempty closed set in a Banach space E (with norm $\|\ \|$), and further, that $S : K \to K$ is a contractive mapping (i.e., a Lipschitzian mapping with Lipschitz constant L strictly lower than one). Then there exists a unique $x^* \in K$ such that $x^* = S(x^*)$ and, for any $x_0 \in K$, the sequence (x_k) defined by $x_{k+1} = S(x_k)$, $k \in \mathbb{N}_0$, stays in K and converges to x^*. Furthermore, the following estimate holds: $\|x_k - x^*\| \leqslant L^k \|x_0 - x^*\|$, for all $k \in \mathbb{N}_0$.*

Corollary 2.4. *Let $S : E \to E$ be a continuously Fréchet differentiable operator and $x^* \in E$ a point such that $S(x^*) = x^*$. If the Fréchet derivative of S at x^* has operator norm strictly lower than one, then the conclusions of the previous theorem hold with $K = \{x \in E :\ \|x - x^*\| \leqslant r\}$,*

for some $r > 0$. In the finite dimensional case $E = \mathbb{R}^n$, this is equivalent to the requirement that $\|DS(x^*)\| < 1$ for some natural norm.[1]

The classical local convergence result for the steepest descent method is now presented. The assumptions, as well as the proof, are somewhat different than the usual ones encountered in most of the literature, in the sense that we regard the method as a fixed-point iteration. This choice suits best our reasoning for the algorithm to come in the next sections.

Theorem 2.5. *Assume that f is a twice continuously differentiable function whose Hessian matrix $D^2f(x^*)$ at a local minimizer x^* is positive definite. Then there exists $r > 0$ such that, given $x_0 \in \bar{B}_r(x^*) = \{x \in \mathbb{R}^n : \|x - x^*\|_2 \leqslant r\}$, the steepest descent method $x_{k+1} = x_k - \eta\nabla f(x_k)$, $k \in \mathbb{N}_0$, converges linearly to x^* for sufficiently small step lengths $\eta > 0$.*

Proof. Taking $S : \mathbb{R}^n \to \mathbb{R}^n$ defined as $S(x) = x - \eta\nabla f(x)$, the steepest descent method becomes $x_{k+1} = S(x_k)$, $k \in \mathbb{N}_0$. Since we are not interested in proving global convergence, the contractivity property will not be needed in all of $\mathbb{R}^n$, but only locally near x^*. By Corollary 2.4, it suffices to check that $\|DS(x^*)\| < 1$ for some natural norm.

It is clear that $DS(x^*) = I - \eta D^2f(x^*)$ is a symmetric matrix; then we know that the ℓ_2 norm of $DS(x^*)$ coincides with the spectral radius of this same matrix (see [3, sec. 1.4]). The eigenvalues of $DS(x^*)$ take the form $1 - \eta\mu_i^*$ $(1 \leqslant i \leqslant n)$, where $\mu_1^* \geqslant \cdots \geqslant \mu_n^*$ are the eigenvalues of $D^2f(x^*)$; given that the latter are all positive, we have $1 - \eta\mu_i^* \in [1 - \eta\mu_1^*, 1 - \eta\mu_n^*]$ $(1 \leqslant i \leqslant n)$ and the choice $0 < \eta < \frac{2}{\mu_1^*}$ implies that $[1 - \eta\mu_1^*, 1 - \eta\mu_n^*] \subset]-1, 1[$. Hence, one gets $\|DS(x^*)\|_2 = \rho(DS(x^*))$ strictly lower than one. □

3 Constrained optimization

In constrained optimization, besides the *objective function* $f : \mathbb{R}^n \to \mathbb{R}$, a *constraint function* $g : \mathbb{R}^n \to \mathbb{R}^m$ is given defining certain equations (or inequations) that the unknown vector x must satisfy. If one is required to minimize f, the optimization problem can, for instance, be written (considering only equality constraints) :

$$\min_{x \in \mathcal{C}} f(x), \qquad \mathcal{C} = \{x \in \mathbb{R}^n : \ g(x) = 0\}. \tag{$\mathcal{P}$}$$

In coordinate notation :

$$\mathcal{C} = \{x \in \mathbb{R}^n : \ g_i(x) = 0, \ 1 \leqslant i \leqslant m\}.$$

The level set $\mathcal{C}$ is usually called the set of *feasible points*.

Constrained optimization problems arise from models in which constraints play an essential role, for instance in imposing budgetary and shape constraints in a structural design problem. Unconstrained optimization problems arise directly in many practical applications, and also as reformulations of constrained ones, if the constraints are replaced by penalization terms added to the objective function having the effect of discouraging violations of the constraints, or by other means (*e.g.* by parametrizing the set $\mathcal{C}$).

Definition 3.1. A point $x \in \mathbb{R}^n$ satisfying the constraint $g(x) = 0$ is said to be a *regular point* if the gradient vectors $\nabla g_1(x), \nabla g_2(x), \ldots, \nabla g_m(x)$ are linearly independent. In other words, the Jacobian matrix $Dg(x)$ should have full rank (equal to m).

[1] A matrix norm that is associated with a vector norm is called a *natural norm*.

Note that at a regular point x the constraint function g is a submersion, giving $\mathcal{C}$ the appropriate geometrical concept, namely that of a submanifold of $\mathbb{R}^n$; the tangent subspace to $\mathcal{C}$ is given by $\mathcal{T}_x = \{\tau \in \mathbb{R}^n : \ Dg(x)\tau = 0\}$. Note also that $m < n$; in fact, $m \geqslant n$ would yield a discrete set of feasible points, a situation which is outside the scope of the present paper.

The following result, relating the distance to the manifold $\mathcal{C}$ and the norm of the constraint functions defining it, will be used in Section 5. The statement is that, locally, those two quantities have the same order of magnitude. Part of the proof relies on the implicit function theorem, much in the same vein like a standard result about manifolds defined as inverse images (see [4, secs. 1.1, 1.2]).

Lemma 3.2. *Let $x^* \in \mathcal{C}$ be a regular point. Then there exist positive constants C_1, C_2 and r, such that, for every $x \in B_r(x^*)$, there holds: $C_1 \operatorname{dist}(x, \mathcal{C}) \leqslant \|g(x)\| \leqslant C_2 \operatorname{dist}(x, \mathcal{C})$.*

Proof. The second inequality is straightforward: since $U \cap \mathcal{C}$ is compact for every compact neighbourhood U of x^*, given x close enough to x^* it is always possible to consider $y_x \in \mathcal{C}$ satisfying $\|x - y_x\| = \inf_{y \in \mathcal{C}} \|x - y\| = \operatorname{dist}(x, \mathcal{C})$; a simple Taylor expansion about y_x yields $g(x) = O(\|x - y_x\|)$, that is, $\|g(x)\| \leqslant C_2 \operatorname{dist}(x, \mathcal{C})$.

The first inequality is far less trivial. Being x^* a regular point, there is a nonzero $m \times m$ minor of $Dg(x^*)$; for simplicity's sake, assume that it is the one featuring the last m columns of that matrix. Then, by the implicit function theorem, there is a neighbourhood of x^* where the last m variables $\tilde{x} = (x_{n-m+1}, \ldots, x_n)$ are function of the first $n - m$ variables $\bar{x} = (x_1, \ldots, x_{n-m})$; more precisely, splitting $\mathbb{R}^n$ into $\mathbb{R}^{n-m} \times \mathbb{R}^m$, there are neighbourhoods U of $\bar{x}^*$ and V of $\tilde{x}^*$, and a continuously differentiable function $\varphi : U \to V$, such that $(U \times V) \cap \mathcal{C}$ is the graph of the map $\bar{x} \mapsto g(\bar{x}, \varphi(\bar{x}))$.

Now, let $D_{\tilde{x}} g(x)$ represent the matrix formed by the last m columns of $Dg(x)$. Using the fact that $(\bar{x}, \varphi(\bar{x})) \in \mathcal{C}$ and the mean value theorem, one can write

$$\|g(x)\| = \|g(\bar{x}, \tilde{x}) - g(\bar{x}, \varphi(\bar{x}))\| = \|D_{\tilde{x}} g(\bar{x}, \tilde{y})(\tilde{x} - \varphi(\bar{x}))\|,$$

where $\tilde{y}$ lies in the line segment joining $\tilde{x}$ to $\varphi(\bar{x})$. Next, define $\alpha : \mathbb{R}^n \to \mathbb{R}$ in the following way: $\alpha(x) = \min_{\tilde{\delta}} \|D_{\tilde{x}} g(x)\tilde{\delta}\|$, where the minimum is computed over all vectors $\tilde{\delta} \in \mathbb{R}^m$ such that $\|\tilde{\delta}\| = 1$. Note that α is well defined, since the unit sphere is compact and the map $\tilde{\delta} \mapsto \|D_{\tilde{x}} g(x)\tilde{\delta}\|$ is obviously continuous. Since $D_{\tilde{x}} g(x^*)$ is an invertible matrix, $\alpha(x^*) > 0$; thus, a property like lower semicontinuity will suffice to ensure that α is bounded below away from zero near x^*. The conclusion of Lemma 3.2 follows then easily, because

$$\|D_{\tilde{x}} g(\bar{x}, \tilde{y})(\tilde{x} - \varphi(\bar{x}))\| \geqslant \alpha(\bar{x}, \tilde{y}) \|\tilde{x} - \varphi(\bar{x})\| \geqslant C_1 \|(\bar{x}, \tilde{x}) - \underbrace{(\bar{x}, \varphi(\bar{x}))}_{\in \mathcal{C}}\| \geqslant C_1 \operatorname{dist}(x, \mathcal{C}).$$

For proving that α is lower semi-continuous,[2] let $x_k \to x$ and take a subsequence (x_{p_k}) such that $\liminf \alpha(x_k) = \lim \alpha(x_{p_k})$ and $\alpha(x_{p_k}) = \|D_{\tilde{x}} g(x_{p_k}) \tilde{\delta}_{p_k}\|$. Using the compactness of the unit sphere, pick a subsequence of $\tilde{\delta}_{p_k}$ converging to some $\tilde{\delta}$; then, $\liminf \alpha(x_k) = \|D_{\tilde{x}} g(x)\tilde{\delta}\| \geqslant \alpha(x)$. □

The *optimality conditions* for constrained optimization problems are more complicated than for the unconstrained case.

[2] Note that α is obviously upper semicontinuous, since it is the lower envelope of the $\tilde{\delta}$-indexed family of (continuous, in particular upper semicontinuous) functions given by $x \mapsto \|D_{\tilde{x}} g(x)\tilde{\delta}\|$. So, in fact, α is a continuous function.

Theorem 3.3. *If $x^* \in \mathbb{R}^n$ is a solution of $(\mathcal{P})$ and x^* is a regular point, then there exists a unique $\lambda^* \in \mathbb{R}^m$ (called the* Lagrange multiplier*) such that the following conditions hold:*

$$\begin{cases} \nabla f(x^*) + \nabla g(x^*)\lambda^* = 0, \\ g(x^*) = 0. \end{cases} \tag{1}$$

In coordinate notation:

$$\begin{cases} f_{,j}(x^*) + \sum_{i=1}^m \lambda_i^* g_{i,j}(x^*) = 0,\ 1 \leqslant j \leqslant n, \\ g_i(x^*) = 0,\ 1 \leqslant i \leqslant m. \end{cases}$$

These equations are often refered to as *Karush-Kuhn-Tucker conditions* or *KKT conditions* for short. They are necessary for optimality, but not sufficient. For the more general case, with inequality constraints also, see [5, p. 160].

A sufficient optimality condition can be given by the action of the Hessian matrices $D^2 f(x^*)$ and $D^2 g_i(x^*)$ of f and g_i $(1 \leqslant i \leqslant m)$, respectively, over tangent vectors to $\mathcal{C}$ at x^*. This sufficient optimality condition involves also the values of the Lagrange multipliers λ_i^* $(1 \leqslant i \leqslant m)$. The following result can be found in many textbooks on optimization (see, for instance, [5, sec. 11.4]).

Theorem 3.4. *Suppose there are $x^* \in \mathbb{R}^n$ and $\lambda^* \in \mathbb{R}^m$ such that the KKT conditions (1) hold. Suppose also that the matrix $H^* = D^2 f(x^*) + \sum_{i=1}^m \lambda_i^* D^2 g_i(x^*)$ is positive definite on $\mathcal{T}_{x^*}$, that is, for any nonzero vector τ tangent to $\mathcal{C}$, there holds $\langle H^*\tau, \tau\rangle > 0$. Then x^* is a strict local minimizer of $(\mathcal{P})$.*

4 A gradient algorithm for equality constrained problems

A typical case in structural design arises when engineers adjust the parameters (variables) to optimize the performance of a structure while keeping a prescribed *cost*. In such a framework, the constraint function g appearing in $(\mathcal{P})$ is thought of as a *cost function*, a scalar function that (in a broad sense) stands for the structure's "price" (or more precisely, the difference between the cost function and a prescribed "price"). For presentation purposes, the discussion will be initially restricted to this model problem $(m = 1)$ and subsequently extended to account for multiple constraints.

For the treatment of $(\mathcal{P})$ we will try to follow, to a certain extent, some of the ideas in section 2. The question of which search direction one should consider does not have an immediate answer. There is more than one aspect to cover as the iterations progress: decreasing the function value of f while solving the equation $g = 0$. The approach proposed in [1] sets up a direction that targets both goals simultaneously, much in the manner of the well known work by Rosen in [6] but with two major differences: *the iterates do not necessarily satisfy the constraints* and *no projection matrices are used.*

Given an iterate x_k, the increment δ_k that defines the next iterate, $x_{k+1} = x_k + \delta_k$, is the sum of two components: one of them is the increment $-\eta\nabla f(x_k)$ (with $\eta > 0$ fixed) corresponding to the steepest descent algorithm; the other one aims at fulfilling the constraint equation $g = 0$ and has the form $-\lambda_k \nabla g(x_k)$, where $\lambda_k \in \mathbb{R}$ is a sort of Lagrange multiplier:

$$\delta_k = -\eta\nabla f(x_k) - \lambda_k \nabla g(x_k)$$

This multiplier is defined adaptively in a natural way by imposing the Newton-type condition

$$\langle \nabla g(x_k), \delta_k \rangle = -g(x_k)$$

which is immediately solvable:

$$\lambda_k = \frac{g(x_k) - \eta\langle \nabla g(x_k), \nabla f(x_k)\rangle}{\|\nabla g(x_k)\|^2}. \tag{2}$$

With this choice of the multiplier, the whole procedure amounts to perform a "tangential gradient method" to minimize f, together with a unidimensional Newton method to solve the constraint equation $g = 0$.

To better understand the last assertion, consider the following reasoning. In the neighborhood of a solution x^* there are two main directions to consider from x_k: the direction $\nabla g(x_k)$, orthogonal to the level set

$$\mathcal{C}_k = \{y \in \mathbb{R}^n : \ g(y) = g(x_k)\},$$

and the subspace orthogonal to it (whose vectors are tangent to $\mathcal{C}_k$ at x_k). In this latter subspace we have to minimize f (note that, since the solution x^* should minimize f in a level set of g, $\mathcal{C}$, there is no point in decreasing f along directions other than tangent ones); in the direction $\nabla g(x_k)$ we want to solve the equation $g = 0$, moving the next iterate closer to $\mathcal{C}$. To clarify things further, take an orthogonal basis of $\mathbb{R}^n$ determined by the unit vector $\nu_k = \|\nabla g(x_k)\|^{-1}\nabla g(x_k)$; then we can write $\nabla f(x_k) = \tau_k + a_k\nu_k$, where $a_k \in \mathbb{R}$ and $\tau_k \perp \nu_k$. With these notations we get

$$\lambda_k = \frac{g(x_k) - \eta a_k\|\nabla g(x_k)\|}{\|\nabla g(x_k)\|^2} = \frac{g(x_k)}{\|\nabla g(x_k)\|^2} - \frac{\eta a_k}{\|\nabla g(x_k)\|}$$

and therefore

$$\delta_k = -\eta\nabla f(x_k) - \lambda_k\nabla g(x_k) = -\eta\tau_k - \frac{g(x_k)}{\|\nabla g(x_k)\|}\nu_k.$$

The tangential component of δ_k is a descent direction for f at x_k:

$$\langle \nabla f(x_k), -\eta\tau_k\rangle = \langle \tau_k + a_k\nu_k, -\eta\tau_k\rangle = -\eta\|\tau_k\|^2 < 0.$$

The normal component of δ_k, equal to $-\frac{g(x_k)}{\|\nabla g(x_k)\|}\nu_k$, clearly alludes to a one-dimensional Newton method.

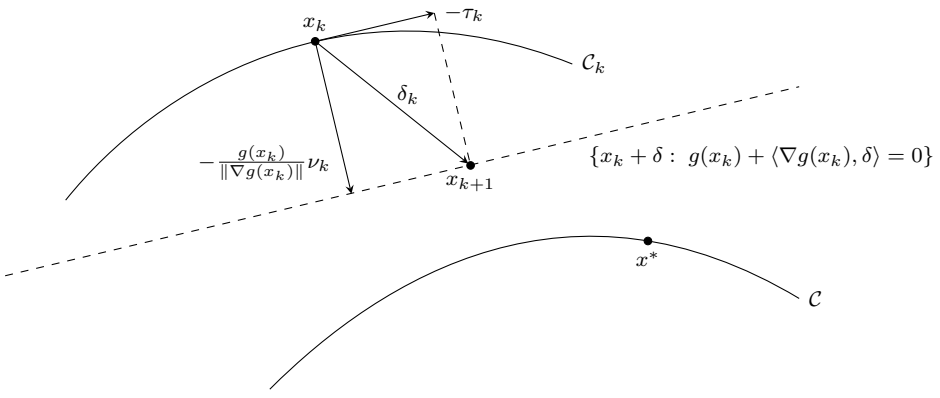

Figure 1: Structure of the step.

As mentioned at the beginning of this section, the algorithm generalizes naturally to vector-valued constraint functions $g : \mathbb{R}^n \to \mathbb{R}^m$ (with $m < n$). In this case $\lambda_k \in \mathbb{R}^m$ but the iterates are defined in a similar fashion by

$$x_{k+1} = x_k - \eta\nabla f(x_k) - \nabla g(x_k)\lambda_k,$$

where (2) now transforms to

$$Dg(x_k)\nabla g(x_k)\lambda_k = g(x_k) - \eta Dg(x_k)\nabla f(x_k). \tag{3}$$

In coordinate notation :

$$(x_{k+1})_j = (x_k)_j - \eta f_{,j}(x_k) - \sum_{i=1}^{m}(\lambda_k)_i g_{i,j}(x_k),\ 1 \leqslant j \leqslant n,$$

where

$$\sum_{i=1}^{m}\sum_{j=1}^{n} g_{l,j}(x_k)g_{i,j}(x_k)(\lambda_k)_i = g_l(x_k) - \sum_{j=1}^{n}\eta g_{l,j}(x_k)f_{,j}(x_k),\ 1 \leqslant l \leqslant m.$$

This linear system of equations uniquely determines λ_k if $Dg(x_k)$ has full rank (equal to m); see Definition 3.1 and the comments following it. Even in the case of vector-valued constraints, the method can be interpreted geometrically as a steepest descent method in the directions tangent to $\mathcal{C}_k$ combined with a Newton method in the directions normal to $\mathcal{C}_k$.

Algorithm 4.1.
INPUT: initial approximation x_0, step length $\eta > 0$, tolerance $\varepsilon > 0$, maximum number of iterations N.
OUTPUT: approximate solution x or message of failure.
Step 1 Set $k = 1$.
Step 2 While $k \leqslant N$ do Steps 3–7.
  Step 3 Compute λ by solving $Dg(x_0)\nabla g(x_0)\lambda = g(x_0) - \eta Dg(x_0)\nabla f(x_0)$.
  Step 4 Set $x = x_0 - \eta\nabla f(x_0) - \nabla g(x_0)\lambda$.
  Step 5 If $\|x - x_0\| < \varepsilon$ then OUTPUT(x);
    STOP.
  Step 6 Set $k = k + 1$.
  Step 7 Set $x_0 = x$.
Step 8 OUTPUT('The method failed after N iterations.');
  STOP.

5 Convergence results

We now state and prove the main theorem regarding the method proposed in section 4.

Theorem 5.1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ ($m < n$) be twice continuously differentiable functions. Let $x^* \in \mathcal{C}$ be a regular point such that the matrix $H^* = D^2 f(x^*) + \sum_{i=1}^{m}\lambda_i^* D^2 g_i(x^*)$ is positive definite on $\mathcal{T}_{x^*}$ (see Theorem 3.4). Then there exists $r > 0$ such that, given $x_0 \in \bar{B}_r(x^*)$, the sequence of iterates defined by*

$$x_{k+1} = x_k - \eta\nabla f(x_k) - \nabla g(x_k)\lambda_k, \quad k \in \mathbb{N}_0, \tag{4}$$

with λ_k determined by (3), converges linearly to x^* for sufficiently smal step lenghts $\eta > 0$.

The reasoning follows the same patern of the proof of Theorem 2.5, but a bit more care will have to be exercised in this case. First of all, an auxiliary result is established.

Lemma 5.2. *let $P \not\equiv 0$ be an orthogonal projection on $\mathbb{R}^n$. If $A \not\equiv 0$ is a self-adjoint linear operator on $\mathbb{R}^n$, then $v \neq 0$ is an eigenvector of PA, associated with the eigenvalue $\mu \neq 0$, if and only if*

(i) $v \in Ran(P)$,

(ii) $(A - \mu I)v \in Ker(P)$.

Hence, the following estimate of the spectral radius holds: $\rho(PA) \leqslant \rho(A\big|_{Ran(P)})$.

Proof. The "if" part of the assertion is trivial. The "only if" part follows basically from the fact that, P being an orthogonal projection, one has the direct sum decomposition $\mathbb{R}^n = Ker(P) \oplus Ran(P)$. Hence, given an eigenpair $v \neq 0$ and $\mu \neq 0$ of PA, there are unique $v_1 \in Ker(P)$ and $v_2 \in Ran(P)$ such that $Av = v_1 + v_2$; but then, $PAv = \mu v$ reads $v_2 = \mu v$. Therefore, it must be $v \in Ran(P)$ and $Av - \mu v = v_1 \in Ker(P)$.

The last estimate is now obvious, since $\rho(PA) = \rho(PA\big|_{Ran(P)})$ and the spectral radius of an operator is dominated by the ℓ^2 norm of that same operator (recall also that $\|P\|_2 = 1$ and that the spectral radius of a self-adjoint operator equals its ℓ^2 norm). □

Remark 5.3. Another useful result regarding spectral radii and matrix norms (whose proof can also be found in [3, sec. 1.4]), is that for any square matrix A and $\varepsilon > 0$, there exists a natural norm with the property that $\|A\| < \rho(A) + \varepsilon$. Adding to this fact the considerations made in Corollary 2.4, one concludes that contractivity properties of differentiable operators $S : \mathbb{R}^n \to \mathbb{R}^n$ are essentially governed by the spectral radius of their Jacobian matrices: if $\rho(DS(x)) < 1$, it always exists a vector norm for which S is locally contractive around x.

Proof of Theorem 5.1. We begin by rewriting the algorithm to display its fixed point nature. Assuming that (3) has a unique solution

$$\lambda_k = [Dg(x_k)\nabla g(x_k)]^{-1}[g(x_k) - \eta Dg(x_k)\nabla f(x_k)], \tag{5}$$

putting this expression into (4) yields

$$\begin{aligned} x_{k+1} = x_k - \eta \underbrace{\left[I - \nabla g(x_k)[Dg(x_k)\nabla g(x_k)]^{-1} Dg(x_k)\right]}_{P(x_k)} \nabla f(x_k) \\ - \underbrace{\nabla g(x_k)[Dg(x_k)\nabla g(x_k)]^{-1}}_{K(x_k)} g(x_k); \end{aligned}$$

so $x_{k+1} = S(x_k)$, upon defining $S(x) = x - P(x)\nabla f(x) - K(x)g(x)$. Because x^* is a regular point, $Dg(x^*)$ has full rank – the same is true then for $Dg(x)$ at nearby x – and the operator S is thus well defined locally around x^*. Because of Remark 5.3, one is left to establish $\rho(DS(x^*)) < 1$.

$K(x)$ is clearly a right inverse of $Dg(x)$ and it is not difficult to prove that $P(x)$ is the matrix of the orthogonal projection onto the tangent subspace $\mathcal{T}_x$ to $\mathcal{C}_x = \{y \in \mathbb{R}^n : \ g(y) = g(x)\}$ at x. There are some trivial relations involving $P(x)$, $K(x)$ and $Dg(x)$, namely: $K(x)Dg(x) = I - P(x)$, $P(x)K(x) = 0$ and $P(x)\nabla g(x) = 0$; in view of this last equality, one can write

$$S(x) = x - \eta P(x)[\nabla f(x) + \nabla g(x)\lambda^*] - K(x)g(x),$$

and it is now easy to see, due to the KKT conditions (1), that the Jacobian matrix of S at x^* is given by

$$\begin{aligned} DS(x^*) &= I - \eta P(x^*)H^* - K(x^*)Dg(x^*) \\ &= I - \eta P(x^*)H^* - [I - P(x^*)] = P(x^*)(I - \eta H^*). \end{aligned}$$

Since $I - \eta H^*$ is a symmetric matrix and $P(x^*)$ is the orthogonal projection's matrix onto $\mathcal{T}_{x^*}$, precisely the subspace where H^* is positive definite, recalling Lemma 5.2 and the proof of Theorem 2.5, the conclusion is now at hand. □

Remark 5.4. The "true" Lagrange multiplier λ^* can be easily approximated because the functional expression defining λ_k, using either (2) or (5) depending on the number of constraints, evaluates to $\eta\lambda^*$ at x^*; since g, Dg and ∇f are all continuous, for x_k near x^* we have $\lambda^* \approx \eta^{-1}\lambda_k$.

Remark 5.5. The constraints converge faster than the iterates. In fact, a simple Taylor expansion about x_k yields

$$g(x_{k+1}) = \underbrace{g(x_k) + Dg(x_k)(x_{k+1} - x_k)}_{=0,\text{ by the Newton-type condition}} + O(\|x_{k+1} - x_k\|^2);$$

thus, if $L < 1$ designates the (local) contractivity constant of the operator S (see the previous proof), it is clear that $\|x_{k+1} - x_k\|^2 = O(L^{2k})$. However, notice that this is not quadratic convergence, but simply an improved linear one.

Remark 5.6. In view of Lemma 3.2, one can assert that locally around x^*, $\|g(x_k)\|$ provides a reasonable estimate of $\operatorname{dist}(x_k, \mathcal{C})$. Thus, Remark 5.5 implies that the distance between x_k and the manifold $\mathcal{C}$ defined by the constraints converges to zero faster than the distance $\|x_k - x^*\|$.

6 Extension to inequality constraints

We now consider the problem

$$\min_{x\in\mathcal{C}} f(x), \qquad \mathcal{C} = \{x \in \mathbb{R}^n :\ g(x) \leqslant 0\},$$

where the inequality is to be understood componentwise:

$$\mathcal{C} = \{x \in \mathbb{R}^n :\ g_i(x) \leqslant 0,\ 1 \leqslant i \leqslant m\}.$$

As in the previous algorithm, one special feature is kept: the iterates are not necessarily feasible. The strategy is that of a usual active set method; this means that, at each iteration, the constraints are partitioned into two separate groups: those inequalities to be treated as active and the ones to be treated as inactive (these are essentially ignored).

The necessary optimality conditions for this sort of problem are better expressed in terms of the active constraints at a solution $x^* \in \mathcal{C}$; so, let $\mathcal{A}^*$ be the set of those indices:

$$\mathcal{A}^* = \{i \in \mathbb{N} :\ 1 \leqslant i \leqslant m,\ g_i(x^*) = 0\}.$$

The KKT conditions can then be written as follows:

$$\begin{cases} \nabla f(x^*) + \sum_{i\in\mathcal{A}^*} \lambda_i^* \nabla g_i(x^*) = 0, \\ g_i(x^*) = 0, \quad i \in \mathcal{A}^*, \\ g_i(x^*) < 0, \quad i \notin \mathcal{A}^*, \\ \lambda_i^* \geqslant 0, \qquad i \in \mathcal{A}^*, \\ \lambda_i^* = 0, \qquad i \notin \mathcal{A}^*. \end{cases} \tag{6}$$

The first two equations are simply the optimality conditions for the equality constrained problem obtained by requiring the active constraints to be zero. The third condition ensures that inactive constraints are satisfied and the last one specifies that they have null Lagrange multipliers attached. The fourth condition is most important for practical purposes: Lagrange multipliers associated with active constraints must be non-negative. This fact will provide a simple criterion to deactivate constraints along the iterations.

The procedure relies on Algorithm 4.1, since at each iteration the active inequalities are to be treated as equality constraints, and on the criteria to activate and deactivate inequalities.

Algorithm 6.1.
INPUT: initial approximation x_0, step lenght $\eta > 0$, tolerance $\varepsilon > 0$, number of iterations N.
OUTPUT: approximate solution x or message of failure.
Step 1 Set $k = 1$.
Step 2 Set $\mathcal{A} = \varnothing$. (No active constraints.)
Step 3 While $k \leqslant N$ do Steps 4–11.
Step 4 For $i = 1, \ldots, m$ do
If $g_i(x_0) > 0$ then set $\mathcal{A} = \mathcal{A} \cup \{i\}$; (Constraint $g_i \leqslant 0$ is set active.)
Step 5 Solve $\sum_{j \in \mathcal{A}} \lambda_j \langle \nabla g_i(x_0), \nabla g_j(x_0) \rangle = g_i(x_0) - \eta \langle \nabla g_i(x_0), \nabla f(x_0) \rangle$, $i \in \mathcal{A}$.
Step 6 Set $i = \arg\min_{j \in \mathcal{A}} \lambda_j$.
Step 7 If $\lambda_i < 0$ then set $\mathcal{A} = \mathcal{A} \setminus \{i\}$; (Constraint $g_i \leqslant 0$ is set inactive.)
GOTO Step 5.
Step 8 Set $x = x_0 - \eta \nabla f(x_0) - \sum_{i \in \mathcal{A}} \lambda_i \nabla g_i(x_0)$.
Step 9 If $\|x - x_0\| < \varepsilon$ then OUTPUT(x);
STOP.
Step 10 Set $k = k + 1$.
Step 11 Set $x_0 = x$.
Step 12 OUTPUT('The method failed after N iterations.');
STOP.

Remark 6.2. Different criteria for deactivating constraints (steps 6 and 7 of the above algorithm) can be considered. For instance, one could deactivate at once all the constraints with negative multipliers. It is an interessting question whether these approaches are equivalent or not. A theoretical study of this issue is the object of ongoing work.

Remark 6.3. By this time, it is also clear how to proceed for a problem like

$$\min_{x \in \mathcal{C}} f(x), \qquad \mathcal{C} = \{x \in \mathbb{R}^n : \ g(x) = 0, \ h(x) \leqslant 0\}.$$

At each iteration, the active inequality constraints join the lot of equality constraints and the treatment is as previously.

Convergence proofs for such methods usually assume some idealized procedure that is hardly employed in practice. We prefer not to state any kind of result, but would like to point out that a successful active set algorithm depends a great deal on the method chosen to solve equality constrained problems. In general, convergence cannot be guaranteed and *zigzagging*[3] can sometimes occur, although experience shows it to be a rare phenomenon.

7 Numerical tests

In all of the following examples, which are well known standard tests from the literature (see for instance [7]), the algorithm stops as soon as the distance between two successive iterates is strictly lower than some prescribed tolerance (which is 10^{-5}, unless stated otherwise). Predictably, when the algorithm converges, it does so (in general) to the nearest solution from the initial guess x_0.

The feasibility of a computed point x_k is tested, of course, by evaluating $g(x_k)$. Most importantly, the first equation in (1), or (6), must be checked; in order to do so, we take the approximation introduced in Remark 5.4.

[3] The set of active constraints changes many times.

Test problem 1

The objective function $f : \mathbb{R}^7 \to \mathbb{R}$ is given by $f(x) = -x_1x_2x_3$ and is subject only to equality constraints: $x_1 - 4.2\sin^2 x_4 = 0$, $x_2 - 4.2\sin^2 x_5 = 0$, $x_3 - 4.2\sin^2 x_6 = 0$ and $x_1 + 2x_2 + 2x_3 - 7.2\sin^2 x_7 = 0$. This problem has multiple solutions with function value $f^* = -3.456$. Departing from $x_0 = (0.4, 2.4, 2.3, 0.1, 1.5, 1.5, 0.4)$, convergence was obtained, using a step size $\eta = 0.09$, to $x^* = (2.4, 1.2, 1.2, \arcsin\sqrt{4/7}, \arcsin\sqrt{2/7}, \arcsin\sqrt{2/7}, \frac{\pi}{2})$.

Test problem 2

The objective function $f : \mathbb{R}^5 \to \mathbb{R}$ is given by

$$f(x) = (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_3 - 1)^2 + (x_4 - 1)^4 + (x_5 - 1)^6$$

and is again subject solely to equality constraints: ${x_1}^2x_4 + \sin(x_4 - x_5) - 2\sqrt{2} = 0$ and $x_2 + {x_3}^4{x_4}^2 - 8 - \sqrt{2} = 0$. The best known solution is $x^* = (1.166172, 1.182111, 1.380257, 1.506036, 0.6109203)$ and it has function value $f^* = 0.24150513$. The initial guess and the step size taken were $x_0 = (2.2, 2.3, 2.1, 2.1, 2.2)$ and $\eta = 0.1$.

Test problem 3

This is an inequality constrained problem, where $f : \mathbb{R}^3 \to \mathbb{R}$ has the expression

$$f(x) = 5x_1 + \frac{50000}{x_1} + 20x_2 + \frac{72000}{x_2} + 10x_3 + \frac{144000}{x_3}$$

and the constraint is given by $\frac{4}{x_1} + \frac{32}{x_2} + \frac{120}{x_3} - 1 \leqslant 0$; bounds on the variables yield three additional constraints: $x_i \geqslant 10^{-5}$, $i = 1, 2, 3$. $x^* = (108.7347175, 85.12613942, 204.3247078)$ is the best known solution and $f^* = 6299.842428$ the function value. In this test, $x_0 = (10, 8, 20)$ and $\eta = 6$.

Test problem 4

Another inequality constrained problem, with $f : \mathbb{R}^7 \to \mathbb{R}$ given by

$$f(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + {x_3}^4 + 3(x_4 - 11)^2 + 10{x_5}^6 + 7{x_6}^2 + {x_7}^4 - 4x_6x_7 - 10x_6 - 8x_7$$

and constraints $7x_1 + 3x_2 + 10{x_3}^2 + x_4 - x_5 - 282 \leqslant 0$, $4{x_1}^2 + {x_2}^2 - 3x_1x_2 + 2{x_3}^2 + 5x_6 - 11x_7 \leqslant 0$, $23x_1 + {x_2}^2 + 6{x_6}^2 - 8x_7 - 196 \leqslant 0$ and $2{x_1}^2 + 3{x_2}^4 + x_3 + 4{x_4}^2 + 5x_5 - 127 \leqslant 0$. The best known solution is $x^* = (2.330499, 1.951372, -0.4775414, 4.365726, -0.6244870, 1.038131, 1.594227)$ and it has function value $f^* = 680.6300573$. The starting point x_0 chosen was the origin and the step size taken was $\eta = 0.04$.

Test problem 5

This test features both equality and inequality constraints. The objective function $f : \mathbb{R}^4 \to \mathbb{R}$ is $f(x) = x_1x_4(x_1 + x_2 + x_3) + x_3$, subject to ${x_1}^2 + {x_2}^2 + {x_3}^2 + {x_4}^2 - 40 = 0$ and $25 - x_1x_2x_3x_4 \leqslant 0$; bounds on the variables yield eight additional inequality constraints: $1 \leqslant x_i \leqslant 5$, $i = 1, 2, 3, 4$. The best known solution is $x^* = (1, 4.7429994, 3.8211503, 1.3794082)$ and $f^* = 17.0140173$ its function value. The initial approximation and step size considered were $x_0 = (3.4, 2.3, 2.1, 2.6)$ and $\eta = 0.08$.

Results

In the table that follows, the notation $\nabla \mathcal{L}_k(x_k)$ stands for $\nabla f(x_k) + \sum_{i \in \mathcal{A}^*} \frac{(\lambda_k)_i}{\eta} \nabla g_i(x_k)$, where $\mathcal{A}^*$ is the set of indices of the active constraints at x^* (that is, all of the equality constraints – if present – plus the active inequality constraints); this is, of course, aimed at the evaluation of the first order KKT condition.

Test	k	$\|x_k - x^*\|_2$	$\|\nabla \mathcal{L}_k(x_k)\|_2$	$\|g(x_k)\|_2$	$\|f(x_k) - f^*\|$
1	134	0.000135479	0.000102493	2.98492×10^{-12}	6.93188×10^{-9}
2	129	0.000119545	8.915×10^{-5}	2.72385×10^{-11}	4.11584×10^{-9}
3	119	0.000133964	6.10348×10^{-7}	4.06576×10^{-16}	7.84785×10^{-8}
4	69	5.93988×10^{-6}	0.000248801	1.13135×10^{-10}	7.47091×10^{-8}
5	61	9.14261×10^{-5}	0.000108068	1.16631×10^{-10}	6.00117×10^{-9}

Table 1: Computed results for Tests 1–5.

A final test

To close this section, we present a simple example to illustrate a most desired feature of a minimization algorithm: the ability to distinguish between local minima and local maxima.

The goal is to minimize $f : \mathbb{R}^2 \to \mathbb{R}$ given by $f(x) = 2x_1x_2$, subject to the constraint $g(x) = {x_1}^2 + {x_2}^2 - 1 = 0$ (*i.e.* to minimize f over the unit circle). The problem clearly has two minimizers, $x^* = (\pm\frac{1}{\sqrt{2}}, \mp\frac{1}{\sqrt{2}})$, with function value $f^* = -1$; obviously, there are also two maximizers $y^* = (\pm\frac{1}{\sqrt{2}}, \pm\frac{1}{\sqrt{2}})$.

x_0	k	$\|x_k - x^*\|_2$	$\|\nabla \mathcal{L}_k(x_k)\|_2$	$\|g(x_k)\|$	$\|f(x_k) - f^*\|$
$(0.71, 0.69)$	11	8.64352×10^{-13}	3.44053×10^{-6}	1.23467×10^{-12}	1.23479×10^{-12}
$(-0.69, -0.68)$	12	7.37537×10^{-13}	3.1207×10^{-6}	1.04987×10^{-12}	1.04983×10^{-12}

Table 2: Departing near local maximizers.

So, even for $x_0 \approx y^*$, the iterates still find their way through x^*. The only way to trick the algorithm is by starting it *exactly* at a local maximum point.

8 Final remarks

In conclusion we would just like to underline that, even without any line search strategy, the performance of the algorithms is quite safisfactory. Algorithm 4.1 converges linearly; in addition, the distance to the manifold $\mathcal{C}$ defined by the constraints converges to zero faster (see Remark 5.6). This leads us to think that some non negligeable improvements could be achieved with just a bit of tune-up (and all of it is done using first derivatives only). We reinforce one of the main properties of this algorithm: it does not get (easily) tricked by local maxima.[4]

Future work includes the study of other possible criteria for deactivating inequality constraints (see Remark 6.2) and the generalization of Algorithm 6.1 to multi-objective optimization ([8]).

[4] This often happens with algorithms that merely attempt to solve the KKT conditions as a system of nonlinear equations, for instance via the Newton method.

References

[1] C. Barbarosie, *Shape Optimization of Periodic Structures*, Computational Mechanics, **30**(3), 235–246, 2003.

[2] J.T. Betts, *An Accelerated Multiplier Method for Nonlinear Programming*, Journal of Optimization Theory and Applications, **21**(2), 137–174, 1977.

[3] P.G. Ciarlet, *Introduction à l'Analyse Numérique Matricielle et à l'Optimisation*, Masson, 1990.

[4] L. Nicolaescu, *Lectures on the Geometry of Manifolds*, World Scientific, 1996.

[5] J. Bonnans, J. Gilbert, C. Lemaréchal, C. Sagastizábal, *Numerical Optimization – Theoretical and Practical Aspects*, Springer, 2003.

[6] J. Rosen, *The Gradient Projection Method for Nonlinear Programming, II. Nonlinear Constraints*, Journal of the Society for Industrial and Applied Mathematics, **9**, 514–532, 1961.

[7] W. Hock, K. Schittkowski, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economics and Mathematical Systems, **187**, Springer-Verlag, 1981.

[8] C. Barbarosie, S. Lopes, *Multi-objective Optimization via an Active Set Method*, in preparation.