SECTION C: CONTINUOUS OPTIMISATION
LECTURE 2: THE DESCENT METHOD AND LINE-SEARCHES

HONOUR SCHOOL OF MATHEMATICS, OXFORD UNIVERSITY
HILARY TERM 2005, DR RAPHAEL HAUSER

1. Unconstrained Optimisation. The subject of this chapter is the unconstrained minimisation problem

$$\min_{x\in\mathbb{R}^n} \; f(x), \tag{1.1}$$

where f is a continuous objective function. Note that no constraints imposed on the decision variables! Furthermore, we usually assume that f is C^2 with Lipschitz-continuous Hessian, that is, there exists $\Lambda > 0$ such that

$$\|D^2f(x) - D^2f(y)\| \le \Lambda\|x-y\| \quad \forall\, x, y \in \mathbb{R}^n.$$

In all that follows $\|x\| = \sqrt{\sum x_i^2}$ denotes the Euclidean norm of a vector $x \in \mathbb{R}^n$ and $\langle\cdot,\cdot\rangle$ is the corresponding Euclidean inner product. If $A : \mathbb{R}^n \to \mathbb{R}^m$ is a linear map, then $\|A\|$ denotes the operator norm defined by the Euclidean norms on $\mathbb{R}^n$ and $\mathbb{R}^m$, that is,

$$\|A\| = \inf\{\lambda > 0 :\ \|Ax\| \le \lambda\|x\| \ \forall\, x \in \mathbb{R}^n\}.$$

The gradient $\nabla f(x)$ of a function $f : \mathbb{R}^n \to \mathbb{R}$ is sometimes denoted by $g_f(x)$, and its Hessian $D^2f(x)$ by $H_f(x)$. The Jacobian $Df(x)$ of a function $f : \mathbb{R}^n \to \mathbb{R}^m$ is sometimes denoted by $J_f(x)$. Note: if $m = 1$ then $J_f(x) = g_f(x)^{\mathrm{T}}$. We will also use the so-called "big O" notation: we say that a function $g(x)$ is *of order* $\|x\|^k$ and write $g(x) = O(\|x\|^k)$ if there exists a constant $c > 0$ and a $\delta > 0$ such that $|g(x)| \le c\|x\|^k$ whenever $\|x\| < \delta$.

Example 1.1 (Risk minimisation under shortselling). *Let us go back to Example 2 of Lecture 1. By eliminating $x_n = 1 - \sum_{i=1}^{n-1} x_i$ we can get rid of the constraint*

$$\sum_{i=1}^{n} x_i = 1.$$

Furthermore, if we allow short-selling of assets, the constraints

$$x_i \ge 0 \quad (i = 1, \dots, n)$$

are no longer imposed. Finally, let us suppose all the assets considered have the same expected return $\mu_i \equiv \mu$, so that the constraint

$$\sum_{i=1}^{n} \mu_i x_i \ge b$$

can be omitted. The investor's aim is to minimise the risk, which can be modelled as

$$\begin{aligned}\min_{x\in\mathbb{R}^{n-1}} \; f(x_1,\dots,x_{n-1}) = &\sum_{i=1}^{n-1}\sum_{j=1}^{n-1}\sigma_{ij}x_ix_j + \sum_{j=1}^{n-1}\sigma_{nj}\big(1-\sum_{i=1}^{n-1}x_i\big)x_j \\ &+ \sum_{i=1}^{n-1}\sigma_{in}x_i\big(1-\sum_{j=1}^{n-1}x_j\big) + \sigma_{nn}\big(1-\sum_{i=1}^{n-1}x_i\big)\big(1-\sum_{j=1}^{n-1}x_j\big).\end{aligned}$$

Since the objective function f is a quadratic (degree 2) polynomial in the decision variables $x_1, \dots, x_{n-1}$, we have $f \in C^\infty$. Moreover, the Hessian $D^2f(x)$ is the same $(n-1)\times(n-1)$ matrix

$$\begin{bmatrix} 1 & & 0 & -1 \\ & \ddots & & -1 \\ 0 & & 1 & -1 \end{bmatrix} \begin{bmatrix} \sigma_{11} & \dots & \sigma_{1n} \\ & \ddots & \\ \sigma_{n1} & \dots & \sigma_{nn} \end{bmatrix} \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \\ -1 & \dots & -1 \end{bmatrix}$$

for all x, and hence $x \mapsto D^2f(x)$ is a constant function, which is of course Lipschitz-continuous: $\|D^2f(x) - D^2f(y)\| = 0 \le 0 \times \|x-y\| \quad \forall\, x, y \in \mathbb{R}^{n-1}$.

Example 1.2. *On a CAD system it takes n parameters $x_1, \dots, x_n$ to define the shape of a car. An engineer has a piece of software which takes the design parameters $x \in \mathbb{R}^n$ as input and computes the air resistance $f(x)$ of the corresponding fuselage as output. The software contains typically millions of lines of code, but for theoretical reasons it is known that $f \in C^2$. Using an* automatic differentiation *system, the engineer can automatically produce a piece of software that computes directional derivatives*

$$D_vf(x) = \frac{d}{dt}f(x+tv), \qquad D_{u,v}f(x) = \frac{d^2}{ds\,dt}f(x+su+tv).$$

How to choose the design parameters so as to minimise the drag on the fuselage?

Note that in this example the objective function is not available explicitly. This is typical for many applications. In fact, evaluating the objective function might even involve measurements in a physical experiment. Besides from appearing as subproblems in constrained optimisation procedures, unconstrained optimisation problems also appear in many applications directly.

2. Optimality Conditions for Unconstrained Minimisation. A well designed optimization algorithm should be able to recognise when an approximate minimum has been attained. We therefore need a mathematical characterisation of local minimisers.

At school we all learned that in the univariate case, a necessary condition is that $f'(x) = 0$, and that second derivatives help deciding whether x is a local maximiser or minimiser. The same idea works in higher dimensions:

Theorem 2.1.

(i) If $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable at $x^* \in \mathbb{R}^n$ and has a local minimum there, then $\nabla f(x^*) = 0$, that is, x^* is a stationary point *of f. This is a first order necessary optimality condition, because it involves first derivatives, or the first order Taylor approximation of f.*

(ii) If $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable at $x^* \in \mathbb{R}^n$ and has a local minimum there, then the Hessian $D^2f(x^*)$ is positive semidefinite, that is, $h^{\mathrm{T}}D^2f(x^*)h \ge 0$ for all $h \in \mathbb{R}^n$. This is a second order *necessary optimality condition.*

(iii) If $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable at $x^* \in \mathbb{R}^n$, and if $\nabla f(x^*) = 0$ and $D^2f(x^*)$ is positive definite, that is, if $h^{\mathrm{T}}D^2f(x^*)h > 0$ for all $h \in \mathbb{R}^n \setminus \{0\}$, then x^* is a local minimiser of f. These are sufficient optimality conditions.

Proof. (i) Since x^* is a local minimiser, there exists $\epsilon > 0$ such that

$$f(x^*) \le f(x^* + h), \quad \forall h \in B_\epsilon(0), \tag{2.1}$$

where $B_\epsilon(0)$ is the open ball of radius ϵ around the origin in $\mathbb{R}^n$. But this implies that

$$\langle \nabla f(x^*), h\rangle = \lim_{t\to 0} \frac{f(x^*+th) - f(x^*)}{t} \ge \lim_{t\to 0} \frac{f(x^*) - f(x^*)}{t} = 0, \quad \forall h \in \mathbb{R}^n.$$

In particular, when we apply this inequality to $h = -\nabla f(x^*)$, we find

$$0 \le \langle \nabla f(x^*), -\nabla f(x^*)\rangle = -\|\nabla f(x^*\|^2 \le 0.$$

(ii) Taking part (i) into account, the second order Taylor approximation of f around x^* is

$$\begin{aligned} f(x^*+h) &= f(x^*) + \langle \nabla f(x^*), h\rangle + \frac{1}{2} h^{\mathrm{T}} D^2 f(x^*) h + O(\|h\|^3) \\ &\overset{(i)}{=} f(x^*) + \frac{1}{2} h^{\mathrm{T}} D^2 f(x^*) h + O(\|h\|^3). \end{aligned} \tag{2.2}$$

If $D^2 f(x^*)$ is not positive semidefinite, then there exists a nonzero vector $p \in \mathbb{R}^n$ such that $p^{\mathrm{T}} \nabla f(x^*) p < 0$, and then we have

$$f(x^* + tp) = f(x^*) + \frac{t^2}{2} p^{\mathrm{T}} D^2 f(x^*) p + O(t^3 \|p\|^3).$$

Let $c, \delta > 0$ be such that $|O(\|h\|^3)| \le c\|h\|^3$ for all $h \in B_\delta(0)$, and let ϵ be chosen as in part (i). Then for all

$$t < \min\left(\frac{\epsilon}{\|p\|}, \frac{\delta}{\|p\|}, \frac{-p^{\mathrm{T}} D^2 f(x^*) p}{2c\|p\|^3} \right)$$

we have

$$\frac{t^2}{2} p^{\mathrm{T}} D^2 f(x^*) p + O(\|tp\|^3) < \frac{t^2}{2} p^{\mathrm{T}} D^2 f(x^*) p + c\|p\|^3 t^3 < 0,$$

and then $f(x^* + tp) < f(x^*)$, contradicting (2.1), because $\|tp\| < \epsilon$.

(iii) Let us again consider the second order Taylor approximation (2.2) of the function $f(x^* + h)$ as a function of h. Since $D^2 f(x^*)$ is positive definite symmetric, all of its eigenvalues $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_n > 0$ are strictly positive and there exists an orthonormal basis $v_1, \ldots, v_n$ of eigenvectors corresponding to these eigenvalues. That is to say,

$$\begin{aligned} \langle v_i, v_i\rangle &= 1, \quad \forall i, \\ \langle v_i, v_j\rangle &= 0, \quad \forall i \ne j, \\ D^2 f(x^*) v_i &= \sigma_i v_i, \quad \forall i. \end{aligned}$$

Note that this means that for any $h \in \mathbb{R}^n$ we have

$$h = \sum_{i=1}^n \langle v_i, h\rangle v_i, \quad \text{and} \quad D^2 f(x^*) h = \sum_{i=1}^n \langle v_i, h\rangle \sigma_i v_i.$$

Therefore,

$$\begin{aligned} h^{\mathrm{T}} D^2 f(x^*) h &= \left(\sum_{i=1}^n \langle v_i, h\rangle v_i \right)^{\mathrm{T}} \left(\sum_{j=1}^n \langle v_j, h\rangle \sigma_j v_j \right) = \sum_{i,j} \langle v_i, h\rangle \langle v_j, h\rangle \sigma_j \langle v_i, v_j\rangle \\ &= \sum_{i=1}^n \langle v_i, h\rangle^2 \sigma_i \ge \sigma_n \sum_{i=1}^n \langle v_i, h\rangle^2 = \sigma_n \sum_{i=1}^n \big\langle \langle v_i, h\rangle v_i, \langle v_i, h\rangle v_i \big\rangle \\ &= \sigma_n \Big\langle \sum_{i=1}^n \langle v_i, h\rangle v_i, \sum_{j=1}^n \langle v_j, h\rangle v_j \Big\rangle = \sigma_n \|h\|^2. \end{aligned} \tag{2.3}$$

Let $c, \delta > 0$ be as in part (ii). Then (2.3) implies that for all h such that $\|h\| < \min(\delta, \sigma_n/2c)$ we have

$$\begin{aligned} f(x^*+h) &= f(x^*) + \frac{1}{2} h^{\mathrm{T}} D^2 f(x^*) h + O(\|h\|^3) \overset{(2.3)}{\ge} f(x^*) + \frac{1}{2}\|h\|^2 \sigma_n - c\|h\|^3 \\ &\ge f(x^*) + \frac{1}{2}\|h\|^2 \sigma_n - c\frac{\sigma_n}{2c}\|h\|^2 = f(x^*), \end{aligned}$$

which shows that x^* is a local minimiser of f. □

3. Line-Search Descent Methods. The optimality conditions we just derived play an important role in the construction of algorithms: Solving the simultaneous system of nonlinear equations

$$\nabla f(x) = 0$$

by an iterative procedure generating a sequence of points $(x_k)_{\mathbb{N}}$, if we can assure that $f(x_k)$ decreases in each iteration,

$$f(x_{k+1}) \le f(x_k) \quad \forall k,$$

then in practice $(x_k)_{\mathbb{N}}$ can only converge to a *local minimiser* x^* and

$$\|\nabla f(x^*)\| < \epsilon$$

can be used as a stopping criterion. Thus, solving unconstrained optimisation problems is closely related to the problem of solving simultaneous equations with the added feature that progress can be controlled by monitoring a naturally defined *merit function* (i.e., one asks "does f decrease?").

Most competitive algorithms for unconstrained minimisation are based on this idea. There are two main families of such methods: *line-search methods* and *trust region methods*. We start with a description of the former.

Example 3.1 (Steepest descent without line searches). *A simple method is defined as follows: starting from some $x_0 \in \mathbb{R}^n$, compute a sequence of intermediate solutions $(x_k)_{\mathbb{N}}$ as follows,*

$$x_{k+1} = x_k - \nabla f(x_k).$$

The method is motivated by the fact that $-\nabla f(x_k)$ is the direction in which f decreases fastest when moving away from x_k. But is it a descent method? The first order Taylor approximation of f shows that $f(x_k - \alpha\nabla f(x_k)) \leq f(x_k)$ for small $\alpha > 0$. However, it is not necessarily the case that $f(x_{k+1}) \leq f(x_k)$, as the step $-\nabla f(x_k)$ can be too far. To make this a true descent method, we have to use *line-searches*: in each iteration we have to find $\alpha_k > 0$ such that

$$f(x_k - \alpha_k \nabla f(x_k)) < f(x_k),$$

and then we can set

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k).$$

A word of warning: although this method works in principle, it is too primitive to produce any good results in practice! We will later learn why. For now we set out to generalise this example.

Algorithm 3.2 (Descent method).
S0 *Choose a starting point $x_0 \in \mathbb{R}^n$ and a tolerance parameter $\epsilon > 0$. Set $k = 0$.*
S1 *If $\|\nabla f(x_k)\| \leq \epsilon$ then stop and output x_k as an approximate local minimiser.*
S2 *Otherwise choose a search direction $d_k \in \mathbb{R}^n$ such that $\langle \nabla f(x_k), d_k\rangle < 0$.*
S3 *Choose a step size $\alpha_k > 0$ such that $f(x_k + \alpha_k d_k) < f(x_k)$.*
S4 *Set $x_{k+1} := x_k + \alpha_k d_k$, replace k by $k+1$, and go back to S1.*

Below we will see that the minimal assumption we need to make for this algorithm to work is $f \in C^1$ with Lipschitz continuous gradient.

The generality of Algorithm 3.2 leaves flexibility both in the choice of the step length α_k and the search direction d_k. In the remainder of this lecture we discuss the step length selection and treat the choice of good search directions in the next few lectures.

3.1. Step Length Selection. The conceptually simplest method of choosing α_k are *exact line searches*, defined by

$$\alpha_k := \inf\{\alpha \geq 0 : \phi'(\alpha) = 0\},$$

where $\phi(\alpha) = f(x_k + \alpha d_k)$. That is to say, the point $x_k + \alpha_k d_k$ is the first stationary point of f encountered along the half line $\{x_k + \alpha d_k : \alpha \geq 0\}$. Note that if $\{\alpha \geq 0 : \phi'(\alpha) = 0\} = \emptyset$, as is the case for example when $\phi(\alpha) = -\ln\alpha$, then $\{\alpha \geq 0 : \phi'(\alpha) = 0\} = \emptyset$, and hence $\alpha_k := \inf\emptyset = +\infty$ corresponds to an infinitely long step which is still sensible.

Exact line searches are mainly a theoretical tool in the convergence analysis of algorithms. In practice, they are computationally too expensive. We will now derive step length computations that are equally good choices for the purposes of Algorithm 3.2 and much cheaper to compute.

Definition 3.3. *We say that α_k satisfies the* Wolfe conditions *if*

$$\phi(\alpha_k) \leq \phi(0) + c_1\alpha_k\phi'(0), \tag{3.1}$$
$$\phi'(\alpha_k) \geq c_2\phi'(0), \tag{3.2}$$

where $0 < c_1 < 1/2$ and $c_1 < c_2 < 1$ are constants, and where ϕ is the function $\phi(\alpha) = f(x_k + \alpha d_k)$.

The Wolfe conditions represent a sensible choice of step length: Condition (3.1) ensures that the actual objective value decrease $f(x_k) - f(x_k + \alpha_k d_k)$ equals at least a fixed fraction of the change $-\alpha_k\langle\nabla f(x_k), d_k\rangle$ predicted by the first order Taylor approximation

$$f(x_k + \alpha_k d_k) \approx f(x_k) + \alpha_k\langle\nabla f(x_k), d_k\rangle.$$

The restriction $c_1 \leq 1/2$ is desirable because this allows α_k to take the value of the exact minimiser when $\phi(\alpha)$ is a convex quadratic function. Condition (3.2) on the other hand guarantees that the step size is not zero, because $\langle\nabla f(x_k + \alpha_k d_k), d_k\rangle$ is substantially larger than $\langle\nabla f(x_k), d_k\rangle$ (which is a negative number).

Proposition 3.4. *If $f \in C^1(\mathbb{R}^n)$ is bounded below on the half-line $\{x_k + \alpha d_k : \alpha \geq 0\}$ then there exists a step length $\alpha_k \in (0, \infty)$ that satisfies the Wolfe conditions.*

Proof. Since the mapping $x \mapsto \nabla f(x)$ is continuous and $\langle\nabla f(x^*), d_k\rangle < 0$, there exists a $\delta > 0$ such that

$$\phi'(\alpha) = \langle\nabla f(x^* + \alpha d_k), d_k\rangle \leq c_1\langle\nabla f(x^*), d_k\rangle$$

for all $\alpha \in [0, \delta]$. But then

$$\phi(\alpha) = \phi(0) + \int_0^\alpha \phi'(t)dt \leq \phi(0) + c_1\alpha\phi'(0).$$

This shows that the first Wolfe condition (3.1) is satisfied for any $\alpha_k \in [0, \delta]$.

Note that if (3.1) is true for all $\alpha \in [0, \infty)$ then f is unbounded below and $\lim_{\alpha\to\infty}\phi(\alpha) = -\infty$. That is, in this case there exists no global minimiser, and this is revealed by an infinite step length. However, since in the statement of the theorem we assumed that ϕ is bounded below,

$$\bar\alpha := \sup\{\alpha : \text{(3.1) holds for } \alpha_k = \alpha\}$$

is a well-defined number.

Note that then (3.1) holds for $\alpha_k = \bar\alpha$. Moreover, $\phi'(\bar\alpha) \geq c_1\phi'(0)$, for otherwise

$$\phi(\bar\alpha + t) = \phi(\bar\alpha) + t\phi'(\bar\alpha) + O(t^2) < \phi(\bar\alpha) + tc_1\phi'(0) \leq \phi(0) + c_1(\bar\alpha + t)\phi'(0)$$

for all $t > 0$ sufficiently small, contradicting the choice of $\bar\alpha$. But since $\phi'(0) < 0$ and $0 < c_1 < c_2$, we have

$$\phi'(\bar\alpha) \geq c_1\phi'(0) > c_2\phi'(0).$$

Therefore, (3.2) holds true for $\alpha_k = \bar\alpha$ too. Thus, $\alpha_k = \bar\alpha$ is a valid choice for the step size. □

To turn the Wolfe conditions into a practical tool that can be used as an element of an algorithm, we need to devise a method for computing a step length α_k that satisfies the Wolfe conditions under the assumptions of Proposition 3.4. The following algorithm does the job:

ALGORITHM 3.5 (Bisection method for step size).
S0 *Choose $\alpha > 0$ and set $\alpha_{low} = \alpha_{high} = 0$.*
S1 *If α satisfies* (3.1) *(that is, if α is long enough) then goto* **S3**.
S2 *Else (if α does not satisfy* (3.1)*) make the replacements $\alpha_{high} \leftarrow \alpha$ and $\alpha \leftarrow (\alpha_{low} + \alpha_{high})/2$, and then goto* **S1**.
S3 *If α satisfies* (3.2) *(that is, α now satisfies both Wolfe conditions) output $\alpha_k = \alpha$ and stop.*
S4 *Otherwise (if α does not satisfy* (3.2)*), make the replacements $\alpha_{low} \leftarrow \alpha$ and*

$$\alpha \leftarrow \begin{cases} 2\alpha_{low} & \text{if } \alpha_{high} = 0, \\ \frac{1}{2}(\alpha_{low} + \alpha_{high}) & \text{if } \alpha_{high} > 0, \end{cases}$$

and then go back to **S1**.

PROPOSITION 3.6. *Under the assumptions of Proposition 3.4, Algorithm 3.5 terminates in finite time and outputs a choice of α_k that satisfies both Wolfe conditions.*

Proof. Note that the two sets

$$W_1 := \{\alpha \geq 0 : (3.1) \text{ holds}\},$$
$$W_2 := \{\alpha \geq 0 : (3.2) \text{ holds}\}$$

are closed subsets of $\mathbb{R}_+$. Moreover,

$$\phi(\alpha) = \phi(0) + \int_0^\alpha \phi'(\tau)d\tau < \phi(0) + \int_0^\alpha c_1\phi'(0)d\tau$$

for all α sufficiently small, because ϕ' is continuous and $c_1 < 1$, showing that there exists $\delta_1 > 0$ such that $[0, \delta_1] \subset W_1$. Let $\alpha > 0$, $(\alpha_{low}^{[i]})_{\mathbb{N}} \subset W_1$ and $(\alpha_{high}^{[i]})_{\mathbb{N}} \subset W_1^c$ be such that

$$\begin{aligned} \alpha_{low}^{[i]} < \alpha \quad \forall i \in \mathbb{N}, \quad \alpha_{low}^{[i]} \overset{i\to\infty}{\longrightarrow} \alpha, \\ \alpha_{high}^{[i]} > \alpha \quad \forall i \in \mathbb{N}, \quad \alpha_{high}^{[i]} \overset{i\to\infty}{\longrightarrow} \alpha. \end{aligned} \tag{3.3}$$

We claim that this implies $\alpha \in W_2^\circ$ (the topological interior of W_2). In fact, suppose to the contrary that $\alpha \in W_2^c$, and hence that $\phi'(\alpha) \leq c_2\phi'(0)$. Then there exists a value $\delta_2 > 0$ such that

$$\phi'(\alpha + \tau) < c_1\phi'(0) \qquad \forall \tau \in [0, \delta_2],$$

because ϕ' is continuous and $c_2 < c_1$. Therefore,

$$\phi(\alpha + \tau) = \phi(\alpha) + \int_\alpha^{\alpha+\tau} \phi'(\theta)d\theta < \phi(0) + c_1(\alpha + \tau)\phi'(0)$$

for all $\tau \in [0, \delta_2]$. Since $\alpha_{high}^{[i]}$ converges to α from the right there exists an index j large enough so that $\alpha_{high}^{[j]} \in [\alpha, \alpha + \delta_2]$, contradicting the assumption that $\alpha_{high}^{[j]} \in W_1^c$. Therefore, it is indeed the case that $\alpha \in W_2^\circ$.

Let us now start analysing the algorithm. Note that we only need to prove that the algorithm terminates in finite time, because the termination criterion is set such that if the algorithm terminates, then α_k satisfies both Wolfe conditions.

- We say that the algorithm starts iteration i when it visits step **S1** for the i-th time, starting with iteration $i = 0$. Let $\alpha_{low}^{[i]}, \alpha_{high}^{[i]}$ and $\alpha^{[i]}$ denote the values of α_{low}, α_{high} and α respectively just before the algorithm enters iteration i.
- Note that it is impossible that $\alpha_{low}^{[i]} = 0$ for all i, because in that case $\alpha^{[i]} = 2^{-i}\alpha^{[0]}$, and ultimately $\alpha^{[i]} \in [0, \delta_1] \subset W_1$ and α_{low} is updated to $\alpha^{[i]} > 0$.
- $(\alpha_{low}^{[i]})_{\mathbb{N}}$ is an increasing sequence in W_1 such that $\alpha_{low}^{[i]} < \alpha^{[i]}$ for all i. In fact, these properties hold true at $i = 0$, and since α_{low} can only be updated in step **S4** it will increase to the strictly larger value $\alpha_{low}^{[i+1]} = \alpha^{[i]}$ and $\alpha^{[i+1]}$ takes on a strictly larger value than $\alpha^{[i]}$ in the same step.
- Initially, $\alpha_{high}^{[i]} = 0$ for a few iterations, but once it takes on a value $\alpha^{[i_0]} > 0$ in some iteration i_0, then this can only happen in step **S2**. From then on $(\alpha_{high}^i)_{\{i\in\mathbb{N}:i\geq i_0\}}$ is a decreasing sequence of values from W_1^c, because α_{high} is only updated in step **S2** to a value of α that is strictly smaller than α_{high} and not in W_1, and α itself is updated to a strictly smaller value.
- Overall, there are only two possible scenarios: either $\alpha_{high}^{[i]} = 0$ for all i, and then $\alpha_{low}^{[i]} = \alpha^{[0]}2^{i-1}$ for all i, in which case the algorithm detects that f is unbounded below in the direction d_k, a situation we excluded in the assumptions of Proposition 3.4. It is thus the second scenario that takes place, which is that there exists an index $i_0 \in \mathbb{N}$ such that $\alpha_{high}^{[i_0]} > 0$, and from then on $\alpha^{[i]} = (\alpha_{high}^{[i]} + \alpha_{low}^{[i]})/2$, $(\alpha_{low}^{[i]})_{\mathbb{N}}$ is increasing, $(\alpha_{high}^{[i]})_{\mathbb{N}}$ is decreasing, and the interval $[\alpha_{low}^{[i]}, \alpha_{high}^{[i]}]$ is halved in length in every iteration. This shows that $\alpha_{low}^{[i]}$ converges to a point α from within W_1 and $\alpha_{high}^{[i]}$ converges to the same point from within W_1^c. By the arguments above, $\alpha \in W_1 \cap W_2^\circ$. Therefore, $\alpha_{low}^{[i]} \in W_1 \cap W_2$ for i sufficiently large, and the algorithm will detect this and terminate with this value.

□

3.2. Convergence of Descent Methods. It is now possible to give a fairly general convergence theorem for Algorithm 3.2 as long as the step lengths satisfy the Wolfe conditions. We prepare the proof through a lemma that gives a useful bound on the amount of decrease in the objective function that is achieved in every iteration:

LEMMA 3.7. *Let Algorithm 3.2 be applied to a C^1 function f with Λ-Lipschitz continuous gradient and assume that the step length α_k satisfies the Wolfe conditions* (3.1) *and* (3.2). *Then*

$$f(x_{k+1}) \leq f(x_k) - c_1(1 - c_2)\frac{(\cos^2\theta_k)\|\nabla f(x_k)\|^2}{\Lambda},$$

where θ_k is the angle between d_k and $-\nabla f(x_k)$, and where c_1, c_2 are the constants from Definition 3.3.

Proof. The second Wolfe condition implies

$$\begin{aligned} \langle \nabla f(x_k + \alpha_k d_k), d_k\rangle - \langle \nabla f(x_k), d_k\rangle = \phi'(\alpha_k) - \phi'(0) \geq (c_2 - 1)\phi'(0) \\ = (1 - c_2)\left(-\langle \nabla f(x_k), d_k\rangle\right). \end{aligned}$$

The Cauchy–Schwartz inequality and the Lipschitz condition imply that the left hand side of this expression is bounded above by $\alpha_k \Lambda \|d_k\|^2$. Therefore,

$$\alpha_k \geq (1 - c_2) \cdot \frac{-\langle \nabla f(x_k), d_k \rangle}{\Lambda \|d_k\|^2}.$$

Since $\langle \nabla f(x_k), d_k \rangle < 0$, Condition (3.1) yields

$$f(x_{k+1}) = \phi(\alpha_k) \leq \phi(0) + c_1 \alpha_k \phi'(0) \leq f(x_k) - c_1(1 - c_2) \frac{\big(\langle \nabla f(x_k), d_k \rangle\big)^2}{\Lambda \|d_k\|^2},$$

and since

$$\langle \nabla f(x_k), d_k \rangle = -\cos\theta_k \|d_k\| \cdot \|\nabla f(x_k)\|,$$

this proves the result. □

The convergence of the descent method is characterised by the following result which shows that either $\nabla f(x_k)$ converges to the zero vector as $k \to \infty$, that is, asymptotically x_k becomes approximately a stationary point, or else the angle θ_k converges to $\pi/2$, which is to say that the search direction asymptotically looses the property of being a descent direction. Whenever the search direction is designed so that θ_k remains bounded away from $\pi/2$ the former situation occurs, and the convergence to a stationary point is guaranteed.

THEOREM 3.8. *Suppose $f \in C^1(\mathbb{R}^n)$ has Lipschitz continuous gradients on $\mathbb{R}^n$ and is bounded below. When Algorithm 3.2 is applied with step lengths α_k that satisfy the Wolfe conditions then*

$$\sum_{k=0}^{\infty} (\cos^2 \theta_k) \|\nabla f(x_k)\|^2 < \infty,$$

where θ_k is defined as in Lemma 3.7.

Proof. Let b be a lower bound for f, that is $f(x) \geq b$ for all $x \in \mathbb{R}^n$. Lemma 3.7 shows that

$$\begin{aligned} f(x_0) - b &\geq f(x_0) - f(x_{k+1}) \\ &\geq f(x_0) - f(x_k) + c_1(1 - c_2) \frac{(\cos^2 \theta_k) \|\nabla f(x_k)\|^2}{\Lambda} \\ &\geq \cdots \\ &\geq f(x_0) - f(x_0) + \frac{c_1(1 - c_2)}{\Lambda} \sum_{k=0}^{j} (\cos^2 \theta_k) \|\nabla f(x_k)\|^2. \end{aligned}$$

Therefore,

$$0 \leq \sum_{k=0}^{j} (\cos^2 \theta_k) \|\nabla f(x_k)\|^2 \leq \frac{\big(f(x_0) - b\big)\Lambda}{c_1(1 - c_2)},$$

and since this is true for all j, this proves the result. □

Theorem 3.8 is valid under the assumption that the objective function is bounded below. It is interesting to note that when this is not the case, the algorithm fails to terminate in finite time but produces a sequence $(x_k)_{\mathbb{N}}$ such that $\lim_{k\to\infty} f(x_k) = -\infty$, which is a perfectly sensible and desireable behaviour under the circumstances.