Lecture 3

Methods with linear convergence

(Multidimensional Convex Problems: Class Complexity; Cutting Plane Scheme, Center of Gravity Method, Ellipsoid Method)

3.1 Class of general convex problems: description and complexity

A convex programming problem is

\[
\begin{align*}
\text{minimize} & \quad f(x) \text{ subject to } g_i(x) \leq 0, \ i = 1, \ldots, m, \ x \in G \subset \mathbb{R}^n. \\
& \quad (3.1.1)
\end{align*}
\]

Here the domain \( G \) of the problem is a closed convex set in \( \mathbb{R}^n \) with a nonempty interior, the objective \( f \) and the functional constraints \( g_i, \ i = 1, \ldots, m, \) are convex continuous functions on \( G. \)

Let us fix a closed and bounded convex domain \( G \subset \mathbb{R}^n \) and the number \( m \) of functional constraints, and let \( \mathcal{P} = \mathcal{P}_m(G) \) be the family of all feasible convex problems with \( m \) functional constraints and the domain \( G. \) Note that since the domain \( G \) is bounded and all problems from the family are feasible, all of them are solvable, due to the standard compactness reasons.

In what follows we identify a problem instance from the family \( \mathcal{P}_m(G) \) with a vector-valued function

\[
p = (f, g_1, \ldots, g_m)
\]

comprised of the objective and the functional constraints.

What we shall be interested in for a long time are the efficient methods for solving problems from the indicated very wide family. Similarly to the one-dimensional case, we assume that the methods have an access to the first order local oracle \( \mathcal{O} \) which, given an input vector \( x \in \text{int} \ G, \) returns the values and some subgradients of the objective and the functional constraints at \( x, \) so that the oracle computes the mapping

\[
x \mapsto \mathcal{O}(p, x) = (f(x), f'(x); g_1(x), g_1'(x); \ldots; g_m(x), g_m'(x)) : \text{int} \ G \rightarrow \mathbb{R}^{(m+1)\times(n+1)}.
\]
The notions of a method and its complexity at a problem instance and on the whole family are introduced exactly as it was done in Section 1.2 of our first lecture\(^1\).

We define the accuracy of a method at a problem and on the family in the following way. Let us start with the \textit{vector of residuals} of a point \(x \in G\) regarded as an approximate solution to a problem instance \(p\):

\[
\text{Residual}(p, x) = (f(x) - f^*, (g_1(x))_+, \ldots, (g_m(x))_+)
\]

which is comprised of the inaccuracy in the objective and the violations of functional constraints at \(x\). In order to get a convenient scalar accuracy measure, it is reasonable to pass from this vector to the \textit{relative accuracy}

\[
\varepsilon(p, x) = \max \left\{ \frac{f(x) - f^*}{\max_G f - f^*}, \frac{(g_1(x))_+}{(\max_G g_1)_+}, \ldots, \frac{(g_m(x))_+}{(\max_G g_m)_+} \right\};
\]

to get the relative accuracy, we normalize each of the components of the vector of residuals by its maximal, over all \(x \in G\), value and take the maximum of the resulting quantities. It is clear that the relative accuracy takes its values in \([0, 1]\) and is zero if and only if \(x\) is an optimal solution to \(p\), as it should be for a reasonable accuracy measure.

After we have agreed how to measure accuracy of tentative approximate solutions, we define the accuracy of a method \(\mathcal{M}\) at a problem instance as the accuracy of the approximate solution found by the method when applied to the instance:

\[
\text{Accuracy}(\mathcal{M}, p) = \varepsilon(p, \bar{x}(p, \mathcal{M})).
\]

The accuracy of the method on the family is its worse-case accuracy at the problems of the family:

\[
\text{Accuracy}(\mathcal{M}) = \sup_{p \in \mathcal{P}_m(G)} \text{Accuracy}(\mathcal{M}, p).
\]

Last, the complexity of the family is defined in the manner we already are acquainted with, namely, as the best complexity of a method solving all problems from the family to a given accuracy:

\[
\mathcal{A}(\varepsilon) = \min \{ \mathcal{A}(\mathcal{M}) \mid \text{Accuracy}(\mathcal{M}) \leq \varepsilon \}.
\]

What we are about to do is to establish the following main result:

\textbf{Theorem 3.1.1} The complexity \(\mathcal{A}(\varepsilon)\) of the family \(\mathcal{P}_m(G)\) of general-type convex problems on an \(n\)-dimensional closed and bounded convex domain \(G\) satisfies the inequalities

\[
n \left\lfloor \frac{\ln(\frac{1}{\varepsilon})}{6 \ln 2} \right\rfloor - 1 \leq \mathcal{A}(\varepsilon) \leq 2.181 n \ln \left(\frac{1}{\varepsilon} \right) \left(\frac{1}{\varepsilon} \right).
\]

Here the upper bound is valid for all \(\varepsilon < 1\). The lower bound is valid for all \(\varepsilon < \varepsilon(G)\), where

\[
\varepsilon(G) \geq \frac{1}{n^d}
\]

\(^1\)that is a set of rules for forming sequential search points, the moment of termination and the result as functions of the information on the problem; this information is comprised by the answers of the oracle obtained to the moment when a rule is to be applied.
3.2. CUTTING PLANE SCHEME AND CENTER OF GRAVITY METHOD

for all $G \subset \mathbb{R}^n$; for an ellipsoid $G$ one has

$$\varepsilon(G) = \frac{1}{n},$$

and for a parallelootope $G$

$$\varepsilon(G) = 1.$$  

Same as in the one-dimensional case, to prove the theorem means to establish the lower complexity bound and to present a method associated with the upper complexity bound (and thus optimal in complexity, up to an absolute constant factor, for small enough $\varepsilon$, namely, for $0 < \varepsilon < \varepsilon(G)$). We shall start with this latter task, i.e., with constructing an optimal method.

### 3.2 Cutting Plane scheme and Center of Gravity Method

The method we are about to present is based on a very natural extension of the bisection - the cutting plane scheme.

#### 3.2.1 Case of the problem without functional constraints

To explain the cutting plane scheme, let me start with the case when there are no functional constraints at all, so that $m = 0$ and the family is comprised of problems

$$\text{minimize } f(x) \text{ s.t. } x \in G$$

(3.2.3)

of minimizing convex continuous objectives over a given closed and bounded convex domain $G \subset \mathbb{R}^n$.

To solve such a problem, we can use the same basic idea as in the one-dimensional bisection. Namely, choosing somehow the first search point $x_1$, we get from the oracle the value $f(x_1)$ and a subgradient $f'(x_1)$ of $f$; thus, we obtain a linear function

$$f_1(x) = f(x_1) + (x - x_1)^T f'(x_1)$$

which, due to the convexity of $f$, underestimates $f$ everywhere on $G$ and coincides with $f$ at $x_1$:

$$f_1(x) \leq f(x), \; x \in G; \quad f_1(x_1) = f(x_1).$$

If the subgradient is zero, we are done - $x_1$ is an optimal solution. Otherwise we can point out a proper part of $G$ which localizes the optimal set of the problem, namely, the set

$$G_1 = \{x \in G \mid (x - x_1)^T f'(x_1) \leq 0\};$$

indeed, outside this new localizer our linear lower bound $f_1$ for the objective, and therefore the objective itself, is greater than the value of the objective at $x_1$. 
Now, our new localizer of the optimal set, i.e., $G_1$, is, same as $G$, a closed and bounded convex domain, and we may iterate the process by choosing the second search point $x_2$ inside $G_1$ and generating the next localizer

$$G_2 = \{ x \in G_1 \mid (x - x_2)^T f'(x_2) \leq 0 \},$$

and so on. We come to the following generic cutting plane scheme:

starting with the localizer $G_0 \equiv G$, choose $x_i$ in the interior of the current localizer $G_{i-1}$ and check whether $f'(x_i) = 0$; if it is the case, terminate, $x_i$ being the result, otherwise define the new localizer

$$G_i = \{ x \in G_{i-1} \mid (x - x_i)^T f'(x_i) \leq 0 \}$$

and loop.

The approximate solution found after $i$ steps of the routine is, by definition, the best point found so far, i.e., the point

$$\tilde{x}_i \in \{ x_1, ..., x_i \} \text{ such that } f(\tilde{x}_i) = \min_{1 \leq j \leq i} f(x_j).$$

A cutting plane method, i.e., a method associated with the scheme, is governed by the rules for choosing the sequential search points in the localizers. In the one-dimensional case there is, basically, the only natural possibility for this choice - the midpoint of the current localizer (the localizer always is a segment). This choice results exactly in the bisection and enforces the lengths of the localizers to go to 0 at the rate $2^{-i}$, $i$ being the step number. In the multidimensional case the situation is not so simple. Of course, we would like to decrease a reasonably defined size of localizer at the highest possible rate; the problem is, anyhow, which size to choose and how to ensure its decreasing. When choosing a size, we should take care of two things

1. we should have a possibility to conclude that if the size of a current localizer $G_i$ is small, then the inaccuracy of the current approximate solution also is small;
2. we should be able to decrease at certain rate the size of sequential localizers by appropriate choice of the search points in the localizers.

Let us start with a wide enough family of sizes which satisfy the first of our requirements.

**Definition 3.2.1** A real-valued function $\text{Size}(Q)$ defined on the family $Q$ of all closed and bounded convex subsets $Q \subset \mathbb{R}^n$ with a nonempty interior is called a size, if it possesses the following properties:

(Size.1) Positivity: $\text{Size}(Q) > 0$ for any $Q \in Q$;
(Size.2) Monotonicity with respect to inclusion: $\text{Size}(Q) \geq \text{Size}(Q')$ whenever $Q' \subset Q$,
(Q, $Q' \in Q$):
(Size.3) Homogeneity with respect to homotheties: if $Q \in Q$, $\lambda > 0$, $a \in \mathbb{R}^n$ and

$$Q' = a + \lambda(Q - a) = \{ a + \lambda(x - a) \mid x \in Q \}$$
3.2. CUTTING PLANE SCHEME AND CENTER OF GRAVITY METHOD

is the image of \( Q \) under the homothety with the center at the point \( a \) and the coefficient \( \lambda \), then

\[
\text{Size}(Q') = \lambda \text{Size}(Q).
\]

**Example 1.** The diameter

\[
\text{Diam}(Q) = \max \{|x - x'| \mid x, x' \in Q\}
\]
is a size;

**Example 2.** The average diameter

\[
\text{AvDiam}(Q) = (\text{Vol}_n(Q))^{1/n},
\]

\( \text{Vol}_n \) being the usual \( n \)-dimensional volume, is a size.

To the moment these examples are sufficient for us.

Let us prove that any size of the indicated type satisfies requirement (1), i.e., if the size of a localizer is small, then the problem is ”almost” solved.

**Lemma 3.2.1** Let \( \text{Size}(\cdot) \) be a size. Assume that we are solving a convex problem (3.2.3) by a cutting plane method, and let \( G_i \) and \( \bar{x}_i \) be the localizers and approximate solutions generated by the method. Then we have for all \( i \geq 1 \)

\[
\varepsilon(p, \bar{x}_i) \leq \varepsilon_i \equiv \frac{\text{Size}(G_i)}{\text{Size}(G)}, \tag{3.2.4}
\]

\( p \) denoting the problem in question.

**Proof** looks completely similar to that one used for the bisection method. Indeed, let us fix \( i \geq 1 \). If \( \varepsilon_i \geq 1 \), then (3.2.4) is evident - recall that the relative accuracy always is \( \leq 1 \).

Now assume that \( \varepsilon_i < 1 \) for our \( i \). Let us choose \( \alpha \in (\varepsilon_i, 1] \), let \( x^* \) be a minimizer of our objective \( f \) over \( G \) and let

\[
G^\alpha = x^* + \alpha(G - x^*).
\]

Then

\[
\text{Size}(G^\alpha) = \alpha \text{Size}(G) > \varepsilon_i \text{Size}(G) = \text{Size}(G_i)
\]

(we have used the homogeneity of \( \text{Size}(\cdot) \) with respect to homotheties). Thus, the size of \( G^\alpha \) is greater than that on of \( G_i \), and therefore, due to the monotonicity of \( \text{Size} \), \( G^\alpha \) cannot be a subset of \( G_i \). In other words, there exists a point

\[
y \in G^\alpha \setminus G_i.
\]

Since \( G^\alpha \) clearly is contained in the domain of the problem and does not belong to the \( i \)-th localizer \( G_i \), we have

\[
f(y) > f(\bar{x}_i);
\]

indeed, at each step \( j, j \leq i \), of the method we remove from the previous localizer (which initially is the whole domain \( G \) of the problem) only those points where the objective is
greater than at the current search point $x_j$ and is therefore greater than at the best point $\bar{x}_i$ found during the first $i$ steps; since $y$ was removed at one of these steps, we conclude that $f(y) > f(\bar{x}_i)$, as claimed.

On the other hand, $y \in G^\alpha$, so that
\[
y = (1 - \alpha)x^* + \alpha z
\]
with some $z \in G$. From convexity of $f$ it follows that
\[
f(y) \leq (1 - \alpha)f(x^*) + \alpha f(z) \leq (1 - \alpha)\min_G f + \alpha \max_G f,
\]
whence
\[
f(y) - \min_G f \leq \alpha(\max_G f - \min_G f).
\]
As we know, $f(y) > f(\bar{x}_i)$, and we come to
\[
f(\bar{x}_i) - \min_G f \leq \alpha(\max_G f - \min_G f),
\]
or, which is exactly the same, to
\[
\varepsilon(p, \bar{x}_i) < \alpha.
\]
Since $\alpha$ is an arbitrary real $> \varepsilon_i$, we conclude that (3.2.4) holds.

Thus, we realize now what could be the sizes we are interested in, and the problem is how to ensure certain rate of their decreasing along the sequence of localizers generated by a cutting plane method. The difficulty here is that when choosing the next search point in the current localizer, we do not know what will be the next cutting plane; the only thing we know is that it will pass through the search point. Thus, we are interested in the choice of the search point which guarantees certain reasonable, not too close to 1, ratio of the size of the new localizer to that one of the previous localizer independently of what will be the cutting plane. Whether such a choice of the search point is possible, it depends on the size we are using. For example, the diameter of a localizer, which is a very natural measure of it and which was successively used in the one-dimensional case, would be a very bad choice in the multidimensional case. To realize this, imagine that we are minimizing over the unit square on the two-dimensional plane, and our objective in fact depends on the first coordinate only. All our cutting planes (in our example they are lines) will be parallel to the second coordinate axis, and the localizers will be stripes of certain horizontal size (which we may enforce to tend to 0) and of some fixed vertical size (equal to 1). The diameters of the localizers here although decrease but do not tend to zero. Thus, the first of the particular sizes we have looked at does not fit the second requirement. In contrast to this, the second particular size, the average diameter $AvDiam$, is quite appropriate, due to the following geometric fact which we present without proof:

**Proposition 3.2.1 (Grunbaum)** Let $Q$ be a closed and bounded convex domain in $\mathbb{R}^n$, let
\[
x^*(G) = \frac{1}{\text{Vol}_n(G)} \int_G x dx
\]
be the center of gravity of $Q$, and let $\Gamma$ be an affine hyperplane passing through the center of gravity. Then the volumes of the parts $Q', Q''$ in which $Q$ is partitioned by $\Gamma$ satisfy the inequality

$$\text{Vol}_n(Q'), \text{Vol}_n(Q'') \leq \{1 - \left(\frac{n}{n+1}\right)^n\} \text{Vol}_n(Q) \leq \exp\{-\kappa\} \text{Vol}_n(Q),$$

where

$$\kappa = \ln(1 - 1/e) = 0.45867...$$

in other words,

$$\text{AvDiam}(Q'), \text{AvDiam}(Q'') \leq \exp\{-\frac{\kappa}{n}\} \text{AvDiam}(Q). \quad (3.2.5)$$

Note that the proposition states exactly that the smallest (in terms of the volume) fraction you can cut off a $n$-dimensional convex body by a hyperplane passing through the center of gravity of the body is the fraction you get when the body is a simplex, the plane passes parallel to a facet of the simplex and you cut off the part not containing the facet.

**Corollary 3.2.1** Consider the Center of Gravity method, i.e., the cutting plane method with the search points being the centers of gravity of the corresponding localizers:

$$x_i = x^*(G_{i-1}) \equiv \frac{1}{\text{Vol}_n(G_{i-1})} \int_{G_{i-1}} x \, dx.$$

For the method in question one has

$$\text{AvDiam}(G_i) \leq \exp\{-\frac{\kappa}{n}\} \text{AvDiam}(G_{i-1}), \ i \geq 1;$$

consequently (see Lemma 3.2.4) the relative accuracy of $i$-th approximate solution generated by the method as applied to any problem $p$ of minimizing a convex objective over $G$ satisfies the inequality

$$\varepsilon(p, \bar{x}_i) \leq \exp\{-\frac{\kappa}{n}\}, \ i \geq 1.$$

In particular, to solve the problem within relative accuracy $\varepsilon \in (0, 1)$ it suffices to perform no more than

$$N = \lfloor \frac{1}{\kappa} n \ln \left(\frac{1}{\varepsilon}\right) \rfloor \leq \lfloor 2.181 n \ln \left(\frac{1}{\varepsilon}\right) \rfloor \quad (3.2.6)$$

steps of the method.

**Remark 3.2.1** The Center of Gravity method for convex problems without functional constraints was invented in 1965 independently by A.Yu.Levin in the USSR and J. Newman in the USA.
3.3 The general case: problems with functional constraints

The Center of Gravity method, as it was presented, results in the upper complexity bound stated in our Main Theorem, but only for problems without functional constraints. In order to establish the upper bound in full generality, we should modify the cutting plane scheme in a manner which enables us to deal with these constraints. The very first idea is to act as follows: after current search point is chosen and we have received the values and subgradients of the objective and the constraints at the point, let us check whether the point is feasible; if it is the case, then let us use the subgradient of the objective in order to cut off the part of the current localizer where the objective is greater that at the current search point, same as it was done in the case when there were no functional constraints. Now, if there is a functional constraint violated at the point, we can use its subgradient to cut off points where the constraint is for sure greater than it is at the search point; the removed points for sure are not feasible: This straightforward approach cannot be used as it is, since the feasible set may have empty interior, and in this case our process, normally, will never find a feasible search point and, consequently, will never look at the objective. In this case the localizers will shrink to the feasible set of the problem, and this is fine, but if the set is not a singleton, we would not have any idea how to extract from our sequence of search points a point where the constraints are ”almost satisfied” and at the same time the objective is close to the optimal value - recall that we simply did not look at the objective when solving the problem!

There is, anyhow, a simple way to overcome the difficulty - we should use for the cut the subgradient of the objective at the steps when the constraints are ”almost satisfied” at the current search point, not only at the steps when the point is feasible. Let us consider the method, proposed by Nemirovski and Yudin in 1976:

Cutting plane scheme for problems with functional constraints:

Given in advance the desired relative accuracy $\varepsilon \in (0,1)$, generate, starting with $G_0 = G$, the sequence of localizers $G_i$, as follows:

given $G_{i-1}$ (which is a closed and bounded convex subset of $G$ with a nonempty interior), act as follows:

1) choose $i$-th search point

\[ x_i \in \text{int } G_{i-1} \]

and ask the oracle about the values

\[ f(x_i), g_1(x_i), ..., g_m(x_i) \]

and subgradients

\[ f'(x_i), g'_1(x_i), ..., g'_m(x_i) \]

of the objective and the constraints at $x_i$. 
2) Form the affine lower bounds
\[ g_j^{(i)}(x) = g_j(x_i) + (x - x_i)^T g'_j(x_i) \]
for the functional constraints (these actually are lower bounds since
the constraints are convex) and compute the quantities
\[ g_j^{i,*} = \left( \max_{G_j} g_j^{(i)}(x) \right)_{+}, \quad i = 1, \ldots, m. \]

3) If, for all \( j = 1, \ldots, m \),
\[ g_j(x_i) \leq \varepsilon g_j^{i,*}, \quad (3.3.7) \]
claim that \( i \) is a productive step and go to 4.a), otherwise go to 4.b).

4.a) [productive step]
If \( f'(x_i) \neq 0 \), define the new localizer \( G_i \) as
\[ G_i = \{ x \in G_{i-1} \mid (x - x_i)^T f'(x_i) \leq 0 \} \]
and go to 5); otherwise terminate, \( x_i \) being the result formed by the
method.

4.b) [non-productive step]
Choose \( j \) such that
\[ g_j(x_i) > \varepsilon g_j^{i,*}, \]
define the new localizer \( G_i \) as
\[ G_i = \{ x \in G_{i-1} \mid (x - x_i)^T g'_j(x_i) \leq 0 \} \]
and go to 5).

5) Define \( i \)-th approximate solution as the best (with the smallest
value of the objective) of the search points \( x_j \) corresponding to the
productive steps \( j \leq i \). Replace \( i \) by \( i + 1 \) and go to 1).

Note that the approximate solution \( \bar{x}_i \) is defined only if a productive step already has been
performed.

The rate of convergence of the above method is given by the following analogue of Lemma
3.2.4:

**Proposition 3.3.1** Let Size be a size, and let a cutting plane method be applied to a convex
problem \( p \) with \( m \) functional constraints. Assume that for a given \( N \) the method either
terminates in course of the first \( N \) steps, or this is not the case and the relation
\[ \frac{\text{Size}(G_N)}{\text{Size}(G)} < \varepsilon \]  
(3.3.8)
is satisfied. In the first case the result \( \bar{x} \) found by the method is a solution to the problem of
relative accuracy \( \leq \varepsilon \); in the second case the \( N \)-th approximate solution is well-defined and
solves the problem to the relative accuracy \( \varepsilon \).
Proof. Let us first note that for any \( i \) and \( j \) one has
\[
g_{j}^{*,i} \leq \left( \max_{G} g_{j} \right)_{+}; \tag{3.3.9}
\]
this is an immediate consequence of the fact that \( g_{j}^{(i)}(x) \) is a lower bound for \( g_{j}(x) \) (an immediate consequence of the convexity of \( g_{j} \)), so that the maximum of this lower bound over \( x \in G \), i.e., \( g_{j}^{*,i} \), is at most the similar quantity for the constraint \( g_{j} \) itself.

Now, assume that the method terminates at certain step \( i \leq N \). According to the description of the method, it means that \( i \) is a productive step and \( 0 \) is a subgradient of the objective at \( x_{i} \); the latter means that \( x_{i} \) is a minimizer of \( f \) over the whole \( G \), so that
\[
f(x_{i}) \leq f^{*}.
\]
Besides this, \( i \) is a productive step, so that
\[
g_{j}(x_{i}) \leq \varepsilon g_{j}^{*,i} \leq \varepsilon \left( \max_{G} g_{j} \right)_{+}, \quad j = 1, \ldots, m
\]
(we have used (3.3.9)); these inequalities, combined with the definition of the relative accuracy, state exactly that \( x_{i} \) (i.e., the result obtained by the method in the case in question) solves the problem within the relative accuracy \( \varepsilon \), as claimed.

Now assume that the method does not terminate in course of the first \( N \) steps. In view of our premise, here we have
\[
\text{Size}(G_{N}) < \varepsilon \text{Size}(G). \tag{3.3.10}
\]
Let \( x^{*} \) be an optimal solution to the problem, and let
\[
G^{\varepsilon} = x^{*} + \varepsilon(G - x^{*}).
\]
\( G^{\varepsilon} \) is a closed and bounded convex subset of \( G \) with a nonempty interior; due to homogeneity of \( \text{Size} \) with respect to homotheties, we have
\[
\text{Size}(G^{\varepsilon}) = \varepsilon \text{Size}(G) > \text{Size}(G_{N})
\]
(the second inequality here is (3.3.10)). From this inequality and the monotonicity of the size it follows that \( G^{\varepsilon} \) cannot be a subset of \( G_{N} \):
\[
\exists y \in G^{\varepsilon} \setminus G_{N}.
\]
Now, \( y \) is a point of \( G \) (since the whole \( G^{\varepsilon} \) is contained in \( G \)), and since it does not belong to \( G_{N} \), it was cut off at some step of the method, i.e., there is an \( i \leq N \) such that
\[
e_{i}^{T}(y - x_{i}) > 0, \tag{3.3.11}
\]
\( e_{i} \) being the linear functional defining \( i \)-th cut.
Note also that since \( y \in G^\varepsilon \), we have a representation

\[
y = (1 - \varepsilon)x^* + \varepsilon z
\]

(3.3.12)

with certain \( z \in G \).

Let us prove that in fact \( i \)-th step is productive. Indeed, assume it is not the case. Then

\[
e_i = g_j'(x_i)
\]

(3.3.13)

for some \( j \) such that \( g_j'^i(x_i) > \varepsilon g_j^{*,i} \). From this latter inequality combined with (3.3.13) and (3.3.11) it follows that

\[
g_j^i(y) > g_j(x_i) = g_j^i(x_i) > \varepsilon g_j^{*,i}.
\]

(3.3.14)

On the other hand, we have

\[
g_j^i(y) = (1 - \varepsilon)g_j^i(x^*) + \varepsilon g_j^i(z) \leq \varepsilon g_j^{*,i}
\]

(we have taken into account that \( g_j^i \) is a lower bound for \( g_j(\cdot) \) and therefore this bound is nonpositive at the optimal solution to the problem); the resulting inequality contradicts (3.3.14), and thus the step \( i \) indeed is productive.

Now, since \( i \) is a productive step, we have \( e_i = f'(x_i) \), and (3.3.11) implies therefore that

\[
f(y) > f(x_i);
\]

from this latter inequality and (3.3.12), exactly as in the case of problems with no functional constraints, it follows that

\[
f(x_i) - f^* < f(y) - f^* \leq \varepsilon (\max \limits_G f - f^*).
\]

(3.3.15)

Now let us summarize our considerations. We have proved that in the case in question (i.e., when the method does not terminate during first \( N \) steps and (3.3.8) is satisfied) there exist a productive step \( i \leq N \) such that (3.3.15) holds. Since the \( N \)-th approximate solution is the best (in terms of the values of the objective) of the search points generated at the productive steps with step numbers \( \leq N \), it follows that \( \bar{x}_N \) is well-defined and

\[
f(\bar{x}_N) - f^* \leq f(x_i) - f^* \leq \varepsilon (\max \limits_G f - f^*);
\]

(3.3.16)

since \( \bar{x}_N \) is, by construction, the search point generated at certain productive step \( i' \), we have also

\[
g_j(\bar{x}_N) = g_j^{i'}(x_{i'}) \leq \varepsilon g_j^{*,i'} \leq \varepsilon \left( \max \limits_G g_j \right)_+, \ j = 1, ..., m;
\]

these inequalities combined with (3.3.16) results in

\[
\varepsilon(p, \bar{x}_N) \leq \varepsilon,
\]

as claimed. \( \blacksquare \)
Combining Proposition 3.3.1 and the Grunbaum Theorem, we come to the Center of Gravity method for problems with functional constraints. The method is obtained from our general cutting plane scheme for constrained problems by the following specifications:

first, we use, as a current search point, the center of gravity of the previous localizer:

\[ x_i = \frac{1}{\text{Vol}_n(Q_{i-1})} \int_{Q_{i-1}} x \, dx; \]

second, we terminate the method after \( N \)-th step, \( N \) being given by the relation

\[ N = \lfloor 2.181n \ln \left( \frac{1}{\varepsilon} \right) \rfloor. \]

With these specifications the average diameter of \( i \)-th localizer at every step, due to the Grunbaum Theorem, decreases with \( i \) at least as

\[ \exp \left\{ -\frac{\kappa}{n} i \right\} \text{AvDiam}(G), \quad \kappa = 0.45867..., \]

and since \( \frac{1}{\kappa} < 2.181 \), we come to

\[ \text{AvDiam}(G_N) < \varepsilon \text{AvDiam}(G); \]

this latter inequality, in view of Proposition 3.3.1, implies that the method does find an \( \varepsilon \)-solution to every problem from the family, thus justifying the upper complexity bound we are proving.

### 3.4 Lower complexity bound

To complete the proof of Theorem 3.1.1, it remains to establish the lower complexity bound. This is done, basically, in the same way as in the one-dimensional case. In order to avoid things difficult for verbal presentation, we show here a slightly worse lower bound

\[ \mathcal{A}(\varepsilon) \geq O(1) \frac{n \ln \left( \frac{1}{\varepsilon} \right)}{\ln \left( n \ln \left( \frac{1}{\varepsilon} \right) \right)}, \quad 0 < \varepsilon < \varepsilon^*(G), \tag{3.4.17} \]

\( O(1) \) being a positive absolute constant and \( \varepsilon^*(G) \) being certain positive quantity depending on the geometry \( G \) only (we shall see that this quantity measures how \( G \) differs from a parallelepiped). \(^2\)

The ”spoiled” bound (3.4.17) (which is by logarithmic denominator worse than the estimate announced in the Theorem) is more or less immediate consequence of our one-dimensional considerations. Of course, it is sufficient to establish the lower bound

for the case of problems without functional constraints, since the constrained ones form a wider family (indeed, a problem without functional constraints can be thought of as a problem with a given number $m$ of trivial, identically zero functional constraints). Thus, in what follows the number of constraints $m$ is set to 0.

Let us start with the following simple observation. Let, for a given $\varepsilon > 0$ and a convex objective $f$, the set $G_\varepsilon(f)$ be comprised of all approximate solutions to $f$ of relative accuracy not worse than $\varepsilon$:

$$G_\varepsilon(f) = \{ x \in G \mid f(x) - \min_G f \leq \varepsilon (\max_G f - \min_G f) \}.$$

Assume that, for a given $\varepsilon > 0$, we are able to point out a finite set $\mathcal{F}$ of objectives with the following two properties:

(I) no different problems from $\mathcal{F}$ admit a common $\varepsilon$-solution:

$$G_\varepsilon(f) \cap G_\varepsilon(\bar{f}) = \emptyset$$

whenever $f, \bar{f} \in \mathcal{F}$ and $f \neq \bar{f}$;

(II) given in advance that the problem in question belongs to $\mathcal{F}$, one can compress an answer of the first order local oracle to be a $(\log_2 K)$-bit word. It means the following. For certain positive integer $K$ one can indicate a function $I(f, x)$ taking values in a $K$-element set and a function $R(i, x)$ such that

$$O(f, x) = R(I(f, x), x), \quad f \in \mathcal{F}, x \in \text{int} G.$$

In other words, given in advance that the problem we are interested in belongs to $\mathcal{F}$, a method can imitate the first-order oracle $O$ via another oracle $I$ which returns log$_2$ $K$ bits of information rather than infinitely many bits contained in the answer of the first order oracle; given the "compressed" answer $I(f, x)$, a method can substitute this answer, along with $x$ itself, into a universal (defined by $\mathcal{F}$ only) function in order to get the "complete" first-order information on the problem.

E.g., consider the family $\mathcal{F}_n$ comprised of $2^n$ convex functions

$$f(x) = \max_{i=1,\ldots,n} \epsilon_i x_i,$$

where all $\epsilon_i$ are $\pm 1$. At every point $x$ a function from the family admits a subgradient of the form $I(f, x) = \pm \epsilon_i$ ($\epsilon_i$ are the orths of the axes), with $i$, same as the sign at $\epsilon_i$, depending on $f$ and $x$. Assume that the first order oracle in question when asked about $f \in \mathcal{F}_n$ reports a subgradient of exactly this form. Since all functions from the family are homogeneous, given $x$ and $I(f, x)$ we know not only a subgradient of $f$ at $x$, but also the value of $f$ at the point:

$$f(x) = x^T I(f, x).$$

Thus, our particular first-order oracle as restricted onto $\mathcal{F}_n$ can be compressed to log$_2$$(2n)$ bits.

Now let us make the following observation:
(*)
under assumptions (I) and (II) the $\varepsilon$-complexity of the family $F$, and
therefore of every larger family, is at least

\[
\frac{\log_2 |F|}{\log_2 K}.
\]

Indeed, let $M$ be a method which solves all problems from $F$ within accuracy $\varepsilon$ in
no more than $N$ steps. We may assume (since informationally this is the same) that
the method uses the oracle $I$, rather than the first-order oracle. Now, the behavior of
the method is uniquely defined by the sequence of answers of $I$ in course of $N$ steps;
therefore there are at most $K^N$ different sequences of answers and, consequently, no
more than $K^N$ different trajectories of $M$. In particular, the set $\bar{X}$ formed by the
results produced by $M$ as applied to problems from $F$ is comprised of at most $K^N$
points. On the other hand, since $M$ solves every of $|F|$ problems of the family within
accuracy $\varepsilon$, and no two different problems from the family admit a common $\varepsilon$-solution,
$\bar{X}$ should contain at least $|F|$ points. Thus,

\[
K^N \geq |F|,
\]
as claimed.

As an immediate consequence of what was said, we come to the following result:

\textit{the complexity of minimizing a convex function over an $n$-dimensional parallelootope}
\textit{G within relative accuracy $\varepsilon < 1/2$ is at least $n/(1 + \log_2 n)$}.

Indeed, all our problem classes and complexity-related notions are affine invariant, so
that we always may assume the parallelootope $G$ mentioned in the assertion to be the
unit cube

\[
\{x \in \mathbb{R}^n \mid |x|_\infty = \max_i |x_i| \leq 1\}.
\]

For any $\varepsilon < \frac{1}{2}$ the aforementioned family

\[
F_n = \{f(x) = \max_i \varepsilon_i x_i\}
\]
clearly possesses property (I) and, as we have seen, at least for certain first-order oracle
possesses also property (II) with $K = 2n$. We immediately conclude that the complexity
of finding an $\varepsilon$-minimizer, $\varepsilon < 1/2$, of a convex function over an $n$-dimensional
parallelootope is, at least for some first order oracle, no less than

\[
\frac{\log_2 |F|}{\log_2 (2n)},
\]
as claimed. In fact, of course, the complexity is at least $n$ for any first order oracle,
but to prove the latter statement it requires more detailed considerations.

Now let us use the above scheme to derive the lower bound (3.4.17). Recall that
when studying the one-dimensional case, we have introduced certain family of univariate
convex functions which was as follows. The functions of the family form a tree,
with the root ("generation 0") being the function

\[
f^{\text{root}}(x) = |x|;
\]
when subject to the left and to the right modifications, the function produces two "children", let them be called $f_r$ and $f_l$: each of these functions, in turn, may be subject to the right and to the left modification, producing two new functions, so that at the level of "grandchildren" there are four functions $f_{rr}, f_{rl}, f_{lr}, f_{ll}$, and so on. Now, every of the functions $f$ of a "generation" $k > 0$ possesses its own active segment $\delta(f)$ of the length $2^{1-2k}$, and at this segment the function is modulus-like:

$$ f(x) = a^{(k)} + 8^{-k}|x - c(f)|, $$

$c(f)$ being the midpoint of $\delta(f)$. Note that $a^{(k)}$ depends only on the generation $f$. belongs to, not on the particular representative of the generation; note also that the active segments of the $2^k$ functions belonging to the generation $k$ are mutually disjoint and that a function from our "population" coincides with its "parent" outside the active segment of the parent. In what follows it is convenient also to define the active segment of the root function $f_{\text{root}}$ as the whole axis.

Now, let $\mathcal{F}_k$ be the set of $2^k$ functions comprising $k$-th generation in our population. Let us demonstrate that any first order oracle, restricted on to this family of functions, admits compression to $\log_2(2k)$ bits. Indeed, it is clear from our construction that in order to restore, given an $x$, $f(x)$ and a subgradient $f^'(x)$, it suffices to trace the path of predecessors of $f$ - its father, its grandfather, ... - and to find the "youngest" of them, let it be $\tilde{f}$, such that $x$ belongs to the active segment of $\tilde{f}$ (let us call this predecessor the active at $x$ predecessor of $f$). The active at $x$ predecessor of $f$ does exist, since the active segment of the "common predecessor" $f_{\text{root}}$ is the whole axis.

Now, $f$ is obtained from $\tilde{f}$ by a number of modifications; the first of them possibly varies $\tilde{f}$ in a neighborhood of $x$ ($x$ is in the active segment of $\tilde{f}$), but the subsequent modifications do not, since $x$ is outside the corresponding active segments. Thus, in a neighborhood of $x$ $\tilde{f}$ coincides with the function $\hat{f}$ - the modification of $\tilde{f}$ which leads from $\tilde{f}$ to $f$. Now, to identify the local behavior of $\hat{f}$ (i.e., that one of $f$) at $x$, it suffices to indicate the "age" of $\hat{f}$, i.e., the number of the generation it belongs to, and the type of the modification - left or right - which transforms $\tilde{f}$ into $\hat{f}$.

Indeed, given $x$ and the age $\hat{k}$ of $\hat{f}$, we may uniquely identify the active segment of $\hat{f}$ (since the segments for different members of the same generation $\hat{k} - 1$ have no common points): given the age of $\hat{f}$, its active segment and the type of modification leading from $\tilde{f}$ to $\hat{f}$, we, of course, know $\hat{f}$ in a neighborhood of the active segment of $\hat{f}$ and consequently at a neighborhood of $x$.

Thus, to identify the behavior of $f$ at $x$ and therefore to imitate the answer of any given local oracle on the input $x$, it suffices to know the age $\hat{k}$ of the active at $x$ predecessor of $f$ and the type - left or right - of modification which "moves" the predecessor towards $f$, i.e., to know a point from certain $(2k)$-element set, as claimed.

Now let us act as follows. Let us start with the case when our domain $G$ is a parallelopotope; due to affine invariance of our considerations, we may assume $G$ to be the unit $n$-dimensional cube:

$$ G = \{x \in \mathbb{R}^n \mid |x|_{\infty} \leq 1\}. $$

Given a positive integer $k$, consider the family $\mathcal{F}^k$ comprised of the objectives

$$ f_{i_1, \ldots, i_n}(x) = \max\{f_{i_1}(x_1), \ldots, f_{i_n}(x_n)\}, \quad f_{i_s} \in \mathcal{F}_k, \ s = 1, \ldots, n. $$
This family contains $|\mathcal{F}|^n = 2^{nk}$ objectives, all of them clearly being convex and Lipschitz continuous with constant 1 with respect to the uniform norm $| \cdot |_{\infty}$. Let us demonstrate that there exists a first order oracle such that the family, equipped with this oracle, possesses properties (I), and (II), where one should set

$$\varepsilon = 2^{-6k}, \quad K = 2nk. \quad (3.4.18)$$

Indeed, a function $f_{i_1,\ldots,i_n}$ attains its minimum $a^{(k)}$ exactly at the point $x_{i_1,\ldots,i_n}$ with the coordinates comprised of the minimizers of $f_{i_s}(x_s)$. It is clear that within the cube

$$C_{i_1,\ldots,i_n} = \{ x \mid |x - x_{i_1,\ldots,i_n}|_{\infty} \leq 2 \times 2^{-3k} \}$$

(i.e., within the direct product of the active segments of $f_{i_s}$, $s = 1,\ldots,n$) the function is simply

$$a^{(k)} + 2^{-3k} |x - x_{i_1,\ldots,i_n}|_{\infty},$$

therefore outside this cube one has

$$f_{i_1,\ldots,i_n}(x) - \min_G f_{i_1,\ldots,i_n} > 2^{-6k} = \varepsilon.$$

Taking into account that all our functions $f_{i_1,\ldots,i_n}$, being restricted onto the unit cube $G$, take their values in $[0,1]$, so that for these functions absolute inaccuracy in terms of the objective is majorated by the relative accuracy, we come to

$$G_{\varepsilon}(f_{i_1,\ldots,i_n}) \subset C_{i_1,\ldots,i_n}.$$

It remains to note that the cubes $C_{i_1,\ldots,i_n}$ corresponding to various functions from the family are mutually disjoint (since the active segments of different elements of the generation $\mathcal{F}_k$ are disjoint). Thus, (I) is verified.

In order to establish (II), let us note that to find the value and a subgradient of $f_{i_1,\ldots,i_n}$ at a point $x$ it suffices to know the value and a subgradient at $x_{i_s}$ of any function $f_{i_s}$ which is "active" at $x$, i.e., is $\geq$ all other functions participating in the expression for $f_{i_1,\ldots,i_n}$. In turn, as we know, to indicate the value and a subgradient of $f_{i_s}$ it suffices to report a point from a $(2k)$-element set. Thus, one can imitate certain (not any) first order oracle for the family $\mathcal{F}_k$ via a "compressed" oracle reporting $\log_2(2nk)$-bit word (it suffices to indicate the number $s$, $1 \leq s \leq n$ of a component $f_{i_s}$ active at $x$ and a point of a $(2k)$-element set to identify $f_{i_s}$ at $x_{i_s}$).

Thus, we may imitate certain first order oracle for the family $\mathcal{F}_k$ (comprised of $2^{kn}$ functions), given a "compressed" oracle with $K = 2nk$; it follows from (*) that the $\varepsilon$-complexity of $\mathcal{F}$ for $\varepsilon = 2^{-6k}$ (see (3.4.18)) is at least

$$\frac{\log_2(2^{nk})}{\log_2(2nk)},$$

expressing $k$ in terms of $\varepsilon$, we come to

$$\mathcal{A}(\varepsilon) \geq \frac{n\log_2(1/\varepsilon)}{6\log_2(n\log_2(2/\varepsilon))}, \quad \varepsilon = 2^{-6k}, k = 1,2,\ldots;$$
taking into account that $A(\varepsilon)$ is a non-increasing function, we come to an estimate of the type (3.4.17) with $\varepsilon < \varepsilon^*(G) = \frac{1}{128}$ (recall that we were dealing with the unit cube $G$).

Let me stress what was in fact was proved. We have demonstrated that it is possible to equip the family of all convex continuous functions on the unit cube with a first order oracle in such a way that the complexity of the family would satisfy lower bound (3.4.17), provided that $\varepsilon < \frac{1}{128}$.

Now, what happens in the case when $G$ is not a parallelotope? In this case we can find a pair of homothetic parallelotopes, $p$ and $P$, such that $p \subset G \subset P$. Let us choose among these pairs $p, P$ that one with the smallest possible similarity ratio, let it be called $\alpha(G)$. Those who have solved problems given at the previous lecture know, and other should believe that $\alpha(G) \leq \frac{n}{3}$ for any closed and bounded convex body $G \subset \mathbb{R}^n$. After appropriate affine transformation of the space (the transformation does not influence the complexity - once again, all our complexity-related notions are affine invariant) we may assume that $p$ is the unit cube.

Thus, we come to the situation as follows:

$$C_n \equiv \{ x \in \mathbb{R}^n \mid |x|_\infty \leq 1 \} \subset G \subset \{ x \in \mathbb{R}^n \mid |x|_\infty \leq \alpha(G) \}. \quad (3.4.19)$$

Now let us look at the family of problems

$$\text{minimize } f(x) \text{ s.t. } x \in G$$

associated with $f \in F^k$. It is easily seen that all the objectives from the family attain their global (i.e., over the whole space) minima within the unit cube and the sets of approximate solutions of absolute accuracy $2^{-6k}$ to problems of the family are mutually disjoint (these sets are exactly the small cubes already known to us). Therefore our previous reasoning states that it is possible to equip the family with a first order oracle in such a way that the worst-case, with respect to the family $F^k$, complexity of finding an approximate solution of absolute accuracy $2^{-5k}$ is at least $\frac{n \log_2 k}{\log_2(2nk)}$. On the other hand, the Lipschitz constant of every function from the family $F^k$ taken with respect to the uniform norm is, as we know, at most 1, so that the variation

$$\max_G f - \min_G f$$

of such a function on the domain $G$ is at most the diameter of $G$ with respect to the uniform norm; the latter diameter, due to (3.4.19), is at most $2\alpha(G)$. It follows that any method which solves all problems from $F^k$ within relative accuracy $2^{-6k-1}/\alpha(G)$ solves all these problems within absolute accuracy $2^{-6k}$ as well; thus, the complexity of minimizing convex function over $G$ within relative accuracy $2^{-6k-1}/\alpha(G)$ is at least $\frac{n \log_2 k}{\log_2(2nk)}$:

$$A(2^{-6k-1}/\alpha(G)) \geq \frac{n \log_2 k}{\log_2(2nk)}, \quad k = 1, 2, ...$$

This lower bound immediately implies that

$$A(\varepsilon) \geq O(1) \frac{n \log_2(\frac{1}{\alpha(G)\varepsilon})}{\log_2(\frac{n \log_2(\frac{1}{\alpha(G)\varepsilon})}{\log_2(\frac{1}{\alpha(G)\varepsilon})})}, \quad \alpha(G)\varepsilon < \frac{1}{128}.$$
whence, in turn,

\[ A(\varepsilon) \geq O(1) \frac{n \ln(1/\varepsilon)}{\ln(n \ln(1/\varepsilon))}, \quad \varepsilon \leq \varepsilon^*(G) \equiv \frac{1}{128a^2(G)} (\geq \frac{1}{128n^3}); \]

this is exactly what is required in (3.4.17).

Note that our reasoning results in a lower bound which is worse than that one indicated in the Theorem not only by the logarithmic denominator, but also due to the fact that this is a lower bound for a particular first order oracle, not for an arbitrary one. In fact both these shortcomings, i.e. the presence of the denominator and the "oracle-dependent" type of the lower bound, may be overcome by more careful reasoning, but we are not going to reproduce it here.

3.5 The Ellipsoid method

We have presented the Center of Gravity method which is optimal in complexity, up to an absolute constant factor, at least when the required accuracy is small enough. The rate of convergence of the method, which is the best possible theoretically, looks fine also from the practical viewpoint. The upper complexity bound \( O(1)n \ln(1/\varepsilon) \) associated with the method means that in order to improve inaccuracy by an absolute constant factor (say, reduce it by factor 10) it suffices to perform \( O(1)n \) iterations more, which looks not too bad, taking into account how wide is the family of problems the method can be applied to. Thus, one may ask: what were the computational consequences of the method invented as early as in 1965? The answer is: no consequences at all, since the method cannot be used in practice, provided that the dimension of the problem is, say \( > 4 \). The reason is that the auxiliary problems arising at the steps of the method, I mean those of finding centers of gravity, are computationally extremely hard; the source of the difficulty is that the localizers arising in the method may be almost arbitrary solids; indeed, all we can say is that if the domain \( G \) of the problem is a polytope given by \( k_0 \) linear inequalities, then the localizer \( G_i \) formed after \( i \) steps of the method is a polytope given by \( k_0 + i \) linear inequalities; and all known deterministic methods for computing the center of gravity of a polytope in \( R^n \) given by \( k > 2n \) linear inequalities take an exponential in \( n \) time to find the center. In our definition of complexity we ignore the computational effort required to implement the search rules; but in practice this effort, of course, should be taken into account, this is why the Center of Gravity method, for which this latter effort is tremendous, cannot be used at all.

The situation, nevertheless, is not as bad as one could think: to the moment we have not exploit all abilities of the cutting plane scheme.

Let us note that the cutting plane scheme can be "spoiled" as follows. Given previous localizer \( G_{i-1} \), we apply our basic scheme to produce a new localizer, \( \tilde{G}_i \), but now this is something intermediate, not the localizer we are forwarding to the next step (this is why we denote by \( \tilde{G}_i \) the solid which in the basic scheme was designated \( G_i \)); the localizer we do use at the next step is certain larger solid \( G_i \supset \tilde{G}_i \). Thus, at a step of the modified cutting plane scheme we perform a cut, exactly as in the basic scheme, and enlarge the resulting localizer to obtain \( G_i \).
At this point one could ask: what for should we add to an actual localizer something which for sure does not contain optimal solutions? The answer is: acting in this manner, we may stabilize geometry of our localizers and enforce them to be convenient for numerical implementation of the search rules. This is the idea underlying the Ellipsoid method we are about to present.

### 3.5.1 Ellipsoids

Recall that an ellipsoid in $\mathbb{R}^n$ is defined as a level set of a nondegenerate convex quadratic form, i.e., as a set of the type

$$W = \{ x \in \mathbb{R}^n \mid (x - c)^T A (x - c) \leq 1 \},$$

(3.5.20)

where $A$ is an $n \times n$ symmetric positive definite matrix and $c \in \mathbb{R}^n$ is the center of the ellipsoid. An equivalent definition, which will be more convenient for us, says that an ellipsoid is the image of the unit Euclidean ball under a one-to-one affine transformation:

$$W = W(B, c) = \{ x = Bu + c \mid u^T u \leq 1 \},$$

(3.5.21)

where $B$ is an $n \times n$ nonsingular matrix. It is immediately seen that one can pass from representation (3.5.21) to (3.5.20) by setting

$$A = (B^T)^{-1} B^{-1};$$

(3.5.22)

since any symmetric positive definite matrix $A$ admits a representation of the type (3.5.22) (e.g., with $B = A^{-1/2}$), the above definitions indeed are equivalent.

From (3.5.21) it follows immediately that

$$\text{Vol}_n(W(B, c)) = |\text{Det} B| \text{Vol}_n(V),$$

(3.5.23)

where $V$ denotes the unit Euclidean ball in $\mathbb{R}^n$.

Now, by compactness reasons it is easily seen that for any $n$-dimensional solid $Q$ there exist ellipsoids containing $Q$ and among these ellipsoids there is at least one with the smallest volume (in fact this extremal outer ellipsoid of $Q$ is uniquely defined, but we are not interested in the uniqueness issues). The average diameter of this extremal outer ellipsoid is certain function of $Q$, let it be denoted by $\text{EllOut}(Q)$ and called the outer ellipsoidal size:

$$\text{EllOut}(Q) = (\min \{ \text{Vol}_n(W) \mid W \text{ is an ellipsoid containing } Q \})^{1/n}.$$ 

It is immediately seen that the introduced function is a size, i.e., it is positive, monotone with respect to inclusions and homogeneous with respect to similarity transformations of the homogeneity degree 1.

We need the following result.
Lemma 3.5.1 Let $n > 1$, let

\[ W = \{ x = Bu + c \mid u^T u \leq 1 \} \]

be an ellipsoid in $\mathbb{R}^n$, and let

\[ \bar{W} = \{ x \in W \mid (x - c)^T q \leq 0 \} \]

be a "half-ellipsoid" - the intersection of $W$ and a half-space with the boundary hyperplane passing through the center of $W$ (here $q \neq 0$). Then $\bar{W}$ can be covered by an ellipsoid $W^+$ of the volume

\[ \text{Vol}_n(W^+) = \kappa_n(n) \text{Vol}_n(W), \]

where

\[ \kappa_n(n) = \frac{n^2}{n^2 - 1} \sqrt{\frac{n - 1}{n + 1}} \leq \exp\{-\frac{1}{2(n-1)}\}; \]  \hspace{1cm} (3.5.24)

in particular,

\[ \text{EllOut}(W^+) \leq \kappa(n) \text{EllOut}(W) \leq \exp\{-\frac{1}{2n(n-1)}\} \text{EllOut}(W). \]  \hspace{1cm} (3.5.25)

The ellipsoid $W^+$ is given by

\[ W^+ = \{ x = B^+ u + c^+ \mid u^T u \leq 1 \}, \]

where

\[ B^+ = \alpha(n)B - \gamma(n)(Bp)p^T, \quad c^+ = c - \frac{1}{n+1}Bp, \]

and

\[ \alpha(n) = \left\{ \frac{n^2}{n^2 - 1} \right\}^{1/2}, \quad \gamma(n) = \alpha(n) \left( 1 - \sqrt{\frac{n - 1}{n + 1}} \right), \quad p = \frac{B^T q}{q^T BB^T q}. \]

To prove the lemma, it suffices to reduce the situation to the similar one with $W$ being the unit Euclidean ball $V$; indeed, since $W$ is the image of $V$ under the affine transformation $u \mapsto Bu + c$, the half-ellipsoid $\bar{W}$ is the image, under this transformation, of the half-ball

\[ \bar{V} = \{ u \in V \mid (B^T q)^T u \leq 0 \} = \{ u \in V \mid p^T u \leq 0 \}. \]

Now, it is quite straightforward to verify that a half-ball indeed can be covered by an ellipsoid $V^+$ with the volume being the required fraction of the volume of $V$; to verify this, it was one of the exercises of the previous lecture (cf. Exercise 2.3.8), and in the formulation of the exercise you were given the explicit representation of $V^+$. It remains to note that the image of $V^+$ under the affine transformation which maps the unit ball $V$ onto the ellipsoid $W$ is an ellipsoid which clearly contains the half-ellipsoid $\bar{W}$ and is in the same ratio of volumes with respect to $W$ as $V^+$ is with respect to the unit ball $V$ (since the ratio of volumes remains invariant under affine transformations). The ellipsoid $W^+$ given in formulation of the lemma is nothing but the image of $V^+$ under our affine transformation.
3.5.2 The Ellipsoid method

Bearing in mind our "spoiled" cutting plane scheme, we may interpret the statement of Lemma 3.5.1 as follows: assume that at certain step of a cutting plane method the localizer \( G_{i-1} \) to be updated is an ellipsoid. Let us choose the current search point as the center of the ellipsoid; then the cut will result in the intermediate localizer \( \bar{G}_i \) which is a half-ellipsoid. Let us cover this intermediate localizer by an ellipsoid given by our lemma and choose the latter ellipsoid as the new localizer \( G_i \). Now we are in the same position as we were - the new localizer is an ellipsoid, and we may proceed in the same manner. And due to our lemma, we do decrease in certain ratio certain size of localizers - namely, the outer ellipsoidal size \( \text{EllOut} \).

Two issues should be thought of. First, how to initialize our procedure - i.e., how to enforce the initial localizer to be an ellipsoid (recall that in our basic scheme the initial localizer was the domain \( G \) of the problem). The answer is immediate - let us take as \( G_0 \) an arbitrary ellipsoid containing \( G \). The second difficulty is as follows: in our "spoiled" cutting plane scheme the localizers, generally speaking, are not subsets of the domain of the problem; it may, consequently, happen that the center of a localizer is outside the domain; how to perform a cut in the latter case? Here again the difficulty can be immediately overcome. If the center \( x_i \) of the current localizer \( G_{i-1} \) is outside the interior of \( G \), then, due to the Separation Theorem for convex sets, we may find a nonzero linear functional \( e^T x \) which separates \( x_i \) and \( \text{int} \ G \):

\[(x - x_i)^T e \leq 0, \ x \in G.\]

Using \( e \) for the cut, i.e., setting

\[G_i = \{x \in G_{i-1} \mid (x - x_i)^T e \leq 0\},\]

we remove from the previous localizer \( G_{i-1} \) only those points which do not belong to the domain of the problem, so that \( \bar{G}_i \) indeed can be thought of as a new intermediate localizer.

Thus, we come to the Ellipsoid method, due to Nemirovski and Yudin (1979), which, as applied to a convex programming problem

\[
\text{minimize} \ f(x) \ \text{s.t.} \ g_j(x) \leq 0, \ j = 1, \ldots, m, \ x \in G \subset \mathbb{R}^n
\]

works as follows:

**Initialization.** Choose \( n \times n \) nonsingular matrix \( B_0 \) and a point \( x_1 \) such that the ellipsoid

\[G_0 = \{x = B_0 u + x_1 \mid u^T u \leq 1\}\]

contains \( G \). Choose \( \beta > 0 \) such that

\[
\beta \leq \frac{\text{EllOut}(G)}{\text{EllOut}(G_0)}.
\]

**i-th step, \( i \geq 1 \).** Given \( B_{i-1}, x_i \), act as follows:
1) Check whether \(x_i \in \text{int } G\). If it is not the case, then call step \(i\) non-productive, find a nonzero \(e_i\) such that
\[(x - x_i)^T e_i \leq 0 \ \forall x \in G\]
and go to 3), otherwise go to 2).

2) Call the oracle to compute the quantities
\[f(x_i), f'(x_i), g_1(x_i), g'_1(x_i), ..., g_m(x_i), g'_m(x_i)\].

If one of the inequalities
\[g_j(x_i) \leq \varepsilon \left( \max_G \{g_j(x_i) + (x - x_i)^T g'_j(x_i)\} \right)_+, j = 1, ..., m \quad (3.5.26)\]
is violated, say, \(k\)-th of them, call \(i\)-th step non-productive, set
\[e_i = g'_k(x_i)\]
and go to 3).
If all inequalities (3.5.26) are satisfied, call \(i\)-th step productive and set
\[e_i = f'(x_i)\].

If \(e_i = 0\), terminate, \(x_i\) being the result found by the method, otherwise go to 3).

3) Set
\[p = \frac{B^T_{i-1} e_i}{\sqrt{e_i^T B_{i-1} B^T_{i-1} e_i}} , \quad B_i = \alpha(n)B_{i-1} - \gamma(n)(B_{i-1}p)p^T , \quad x_{i+1} = x_i - \frac{1}{n+1}B_{i-1}p, \quad (3.5.27)\]
\(\alpha(n)\) and \(\gamma(n)\) being the quantities from Lemma 3.5.1.
If
\[\kappa^i(n) < \varepsilon \beta, \quad (3.5.28)\]
terminate with the result \(\bar{x}\) being the best, with respect to the objective, of the search points associated with the productive steps:
\[\bar{x} \in \text{Argmin}\{f(x) \mid x \text{ is one of } x_j \text{ with productive } j \leq i\}\]
otherwise go to the next step.

The main result on the Ellipsoid method is as follows:

**Theorem 3.5.1** The associated with a given relative accuracy \(\varepsilon \in (0,1)\) Ellipsoid method \(\text{Ell}(\varepsilon)\), as applied to a problem instance \(p \in \mathcal{P}_m(G)\), terminates in no more than
\[A(\text{Ell}(\varepsilon)) = \left| \ln \left( \frac{1}{\varepsilon \beta} \right) \right| \leq 2n(n-1) \ln \left( \frac{1}{\beta \varepsilon} \right) \]

steps and solves $p$ within relative accuracy $\varepsilon$: the result $\bar{x}$ is well defined and

$$\varepsilon(p, \bar{x}) \leq \varepsilon.$$ 

Given the direction $e_i$ defining $i$-th cut, it takes $O(n^2)$ arithmetic operations to update $(B_{i-1}, x_i)$ into $(B_i, x_{i+1})$.

**Proof.** The complexity bound is an immediate corollary of the termination test (3.5.28). To prove that the method solves $p$ within relative accuracy $\varepsilon$, note that from Lemma 3.5.1 it follows that

$$\text{EllOut}(G_i) \leq \kappa^i(n) \text{EllOut}(G_0) \leq \kappa^i(n)\beta^{-1} \text{EllOut}(G)$$

(the latter inequality comes from the origin of $\beta$). It follows that if the method terminates at a step $N$ due to (3.5.28), then

$$\text{EllOut}(G_N) < \varepsilon \text{EllOut}(G).$$

Due to this latter inequality, we immediately obtain the accuracy estimate as a corollary of our general convergence statement on the cutting plane scheme (Proposition 3.3.1). Although the latter statement was formulated and proved for the basic cutting plane scheme rather than for the ”spoiled” one, the reasoning can be literally repeated in the case of the ”spoiled” scheme. ■

Note that the complexity of the Ellipsoid method depends on $\beta$, i.e., on how good is the initial ellipsoidal localizer we start with. Theoretically, we could choose as $G_0$ the ellipsoid of the smallest volume containing the domain $G$ of the problem, thus ensuring $\beta = 1$; for ”simple” domains, like a box, a simplex or a Euclidean ball, we may start with this optimal ellipsoid not only in theory, but also in practice. Even with this good start, the Ellipsoid method has $O(n)$ times worse theoretical complexity than the Center of Gravity method (here it takes $O(n^2)$ steps to improve inaccuracy by an absolute constant factor). As a compensation of this theoretical drawback, the Ellipsoid method is not only of theoretical interest, it can be used for practical computations as well. Indeed, if $G$ is a simple domain from the above list, then all actions prescribed by rules 1)-3) cost only $O(n(m+n))$ arithmetic operations. Here the term $mn$ comes from the necessity to check whether the current search point is in the interior of $G$ and, if it is not the case, to separate the point from $G$, and also from the necessity to maximize the linear approximations of the constraints over $G$; the term $n^2$ reflects complexity of updating $B_{i-1} \mapsto B_i$ after $e_i$ is found. Thus, the arithmetic cost of a step is quite moderate, incomparably to the tremendous one for the Center of Gravity method.
3.6 Exercises

3.6.1 Some extensions of the Cutting Plane scheme

To the moment we have applied the Cutting Plane scheme to convex optimization problems in the standard form

$$\text{minimize } f(x) \text{ s.t. } g_j(x) \leq 0, \ j = 1, \ldots, m, \ x \in G \subset \mathbb{R}^n$$ (3.6.29)

($G$ is a solid, i.e., a closed and bounded convex set with a nonempty interior, $f$ and $g_j$ are convex and continuous on $G$). In fact the scheme has a wider field of applications. Namely, consider a ”generic” problem as follows:

$$(f) : \text{minimize } f(x) \text{ s.t. } x \in G_f \subset \mathbb{R}^n;$$ (3.6.30)

here $G_f$ is certain (specific for the problem instance) solid and $f$ is a function taking values in the extended real axis $\mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$ and finite on the interior of $G$.

Let us make the following assumption on our abilities to get information on $(f)$:

(A): we have an access to an oracle $O_A$ which, given on input a point $x \in \mathbb{R}^n$, informs us whether $x$ belongs to the interior of $G_f$; if it is not the case, the oracle reports a nonzero functional $e_x$ which separates $x$ and $G_f$, i.e., is such that

$$(y - x)^T e_x \leq 0, \ y \in G_f;$$

if $x \in \text{int } G_f$, then the oracle reports $f(x)$ and a functional $e_x$ such that the level set

$$\text{lev}_f(x) = \{y \in G_f \mid f(y) < f(x)\}$$

is contained in the open half-space

$$\{y \in \mathbb{R}^n \mid (y - x)^T e_x < 0\}.$$

In the mean time we shall see that under assumption (A) we can efficiently solve $(f)$ by cutting plane methods; but before coming to this main issue let me indicate some interesting
particular cases of the "generic" problem $(f)$. 

![Figure 3. A quasiconvex function $h(x)$](image)

**Example 1. Convex optimization in the standard setting.** Consider the usual convex problem (3.6.29) and assume that the feasible set of the problem has a nonempty interior where the constraints are negative. We can rewrite (3.6.29) as (3.6.30) by setting

$$G_f = \{ x \in G \mid g_j(x) \leq 0, \quad j = 1, \ldots, m \}.$$ 

Now, given in advance the domain $G$ of the problem (3.6.29) and being allowed to use a first-order oracle $O$ for (3.6.29), we clearly can imitate the oracle $O_a$ required by (A).

Example 1 is not so interesting - we simply have expressed something we already know in a slightly different way. The next examples are less trivial.

**Example 2. Quasi-convex optimization.** Assume that the objective $f$ involved into (3.6.29) and the functional constraints $g_j$ are quasi-convex rather than convex. Recall that a function $h$ defined on a convex set $Q$ is called *quasi-convex*, if the sets $\text{lev}_{h(x)}h = \{ y \in Q \mid h(y) \leq h(x) \}$ are convex whenever $x \in Q$ (note the difference between $L_h(x)$ and $\text{lev}_{h(x)}h$; the latter set is defined via strict inequality $<$, the former one via $\leq$). Besides quasi-convexity, assume that the functions $f$, $g_j$, $j = 1, \ldots, m$, are regular, i.e., continuous on $G$, differentiable on the interior of $G$ with a nonzero gradient at any point which is not a minimizer of the function over $G$. Last, assume that the feasible set $G_f$ of the problem possesses a nonempty interior and that the constraints are negative on the interior of $G_f$.

If $h$ is a regular quasi-convex function on $G$ and $x \in \text{int} G$, then the set $\text{lev}_{h(x)}h = \{ y \in G \mid h(y) < h(x) \}$ belongs to the half-space

$$\Pi_h(x) = \{ y \in G \mid (y - x)^T h'(x) < 0 \}.$$
Exercise 3.6.1 #+ Prove the latter statement.

Thus, in the case in question the sets \( \{ y \in G \mid f(y) < f(x) \} \), \( \{ y \in G \mid g_j(y) < g_j(x) \} \) are contained in the half-spaces \( \{ y \mid (y-x)^T f'(x) < 0 \} \), \( \{ y \mid (y-x)^T g_j'(x) < 0 \} \), respectively. It follows that in the case in question, same as in the convex case, we, given access to a first-order oracle for (3.6.29), can imitate a required by (A) oracle \( O_A \) for the induced problem (3.6.30).

Example 3. Linear-fractional programming. Consider the problem

\[
\text{minimize } f(x) = \max_{\omega \in \Omega} \left\{ \frac{a_\omega(x)}{b_\omega(x)} \right\} \quad \text{s.t. } g_j(x) \leq 0, \ j = 1, \ldots, m, \ b_\omega(x) > 0, \ \omega \in \Omega, \ x \in G; \tag{3.6.31}
\]

here \( G \) is a solid in \( \mathbb{R}^n \) and \( \Omega \) is a finite set of indices, \( a_\omega \) and \( b_\omega \) are affine functions and \( g_j \) are, say, convex and continuous on \( G \). The problem is, as we see, to minimize the maximum of ratios of given linear forms over the convex set defined by the inclusion \( x \in G \), convex functional constraints \( g_j(x) \leq 0 \) and additional linear constraints expressing positivity of the denominators.

Let us set

\[
G_f = \{ x \in G \mid g_j(x) \leq 0, b_\omega(x) \geq 0, \omega \in \Omega \};
\]

we assume that the (closed and convex) set \( G_f \) possesses a nonempty interior and that the functions \( g_j \) are negative, while \( b_\omega \) are positive on the interior of \( G_f \).

By setting

\[
f(x) = \begin{cases} 
\max_\omega \{ a_\omega(x)/b_\omega(x) \} & \text{if } x \in \text{int } G_f \\
+\infty & \text{otherwise}
\end{cases} \tag{3.6.32}
\]

we can rewrite our problem as

\[
\text{minimize } f(x) \quad \text{s.t. } x \in G_f. \tag{3.6.33}
\]

Now, assume that we are given \( G \) in advance and have an access to a first-order oracle \( O \) which, given on input a point \( x \in \text{int } G \), reports the values and subgradients of functional constraints at \( x \), same as reports all \( a_\omega(\cdot), b_\omega(\cdot) \).

Under this assumptions we can imitate for (3.6.33) the oracle \( O_A \) required by the assumption (A). Indeed, given \( x \in \mathbb{R}^n \), we first check whether \( x \in \text{int } G \), and if it is not the case, find a nonzero functional \( e_x \) which separates \( x \) and \( G \) (we can do it, since \( G \) is known in advance); of course, this functional separates also \( x \) and \( G_f \), as required in (A). Now, if \( x \in \text{int } G \), we ask the first-order oracle \( O \) about the values and subgradients of \( g_j, a_\omega \) and \( b_\omega \) at \( x \) and check whether all \( g_j \) are negative at \( x \) and all \( b_\omega(x) \) are positive. If it is not the case and, say, \( g_k(x) \geq 0 \), we claim that \( x \not\in \text{int } G_f \) and set \( e_x = g_k(x) \); this functional is nonzero (since otherwise \( g_k \) would attain a nonnegative minimum at \( x \), which contradicts our assumptions about the problem) and clearly separates \( x \) and \( G_f \) (due to the convexity of \( g_k \)). Similarly, if one of the denominators \( b_\omega \) is nonpositive at \( x \), we claim that \( x \not\in \text{int } G_f \) and set

\[
e_x = \nabla b_\omega;
\]
here again $e_x$ is nonzero and separates $x$ and $G_f$ (why?)

Last, if $x \in \text{int } G$, all $g_j$ are negative at $x$ and all $b_j$ are positive at the point, we claim that $x \in \text{int } G_f$ (as it actually is the case), find the index $\omega = \omega(x)$ associated with the largest at $x$ of the fractions $a_{\omega}(\cdot)/b_{\omega}(\cdot)$, compute the corresponding fraction at $x$ (this is nothing but $f(x)$) and set

$$e_x = \nabla_y a_{\omega(x)}(y) - f(x)\nabla_y b_{\omega}(x)(y).$$

Since in the latter case we have

$$\text{lev } f(x) \subset \{ y \in G_f \mid \frac{a_{\omega(x)}(y)}{b_{\omega(x)}(y)} < f(x) \} \subset \{ y \in G_f \mid (y-x)^T e_x < 0 \},$$

we do fit the requirements imposed by (A).

Note that the problem of the type (3.6.31) arises, e.g., in the famous von Neumann Model of Economy Growth which is as follows. Consider an economy where $m$ kinds of goods are circulating. The economy is described by a pair of $m \times n$ matrices $A$ and $B$ with positive entries, where row index $i$ stands for goods and column index $j$ stands for "processes". A process $j$ takes, as input, $a_{ij}$ units of good $i$ and produces, as output, $b_{ij}$ units of the same good, per year. Now, let $x^t$ be an $n$-dimensional vector with coordinates $x^t_j$ being the "intensities" by which we let $j$-th process work in year $t$. Then the amount of goods consumed by all the processes run in year $t$ is represented by the vector $Ax^t$, and the amount of goods produced in the same year is given by the vector $Bx^t$. If we have no external sources of goods, then the "trajectory" of our economy should satisfy the inequalities

$$Ax^{t+1} \leq Bx^t, \ t = 0, 1, ...$$

(for $t = 0$ the right hand side should be replaced by a positive vector representing the "starting" amount of goods). Now, in the von Neumann Economic Growth problem it is asked what is the largest growth factor, $\gamma^*$, for which there exists a "semi-stationary growth trajectory", i.e., a trajectory of the type $x^t = \gamma^t x^0$. In other words, we should solve the problem

$$\text{maximize } \gamma \text{ s.t. } \gamma Ax \leq Bx \text{ for some nonzero } x \geq 0.$$  

Without loss of generality, $x$ in the above formulation can be taken as a point form the standard simplex

$$G = \{ x \in \mathbb{R}^n \mid x \geq 0, \sum_j x_j = 1 \}$$

(which should be regarded as a solid in its affine hull). It is clearly seen that the problem in question can be rewritten as follows:

$$\text{minimize } \max \frac{\sum_{i,j} a_{ij}x_j}{\sum_{i,j} b_{ij}x_j} \text{ s.t. } x \in G; \quad (3.6.34)$$

this is a problem of the type (3.6.33).
It is worthy to note that the von Neumann growth factor $\gamma^*$ describes, in a sense, the highest rate of growth of our economy (this is far from being clear in advance: why the "Soviet" proportional growth is the best one? Why could we not get something better along an oscillating trajectory?) One exact statement on optimality of the von Neumann semi-stationary trajectory (or, better to say, the simplest of these statements) is as follows:

**Proposition 3.6.1** Let $\{x^t\}_{t=1}^T$ be a trajectory of our economy, so that $x^t$ are nonnegative, $x^0 \neq 0$ and

$$Ax^{t+1} \leq Bx^t, \quad t = 0, 1, ..., T - 1.$$ 

Assume that $x^T \geq \lambda^T x^0$ for some positive $\lambda$ (so that our trajectory results, for some $T$, in growth of the amount of goods in $\lambda^T$ times in $T$ years). Then $\lambda \leq \gamma^*$.

Note that following the von Neumann trajectory

$$x^t = (\gamma^*)^T x^0,$$

$x^0$ being the $x$-component of an optimal solution to (3.6.34), does ensure growth by factor $(\gamma^*)^T$ each $T$ years.

**Exercise 3.6.2** * Prove Proposition 3.6.1.

**Example 4. Generalized Eigenvalue problem.** Let us again look at problem (3.6.31). What happens if the index set $\Omega$ becomes infinite? To avoid minor elaborations, let us assume (as it actually is the case in most applications) that $\Omega$ is a compact set and that the functions $a_\omega(x)$, $b_\omega(x)$ are continuous in $(x, \omega)$ (and, as above, are affine in $x$). As far as reformulation (3.6.32) - (3.6.33) is concerned, no difficulties occur. The possibility to imitate the oracle $O_A$, this is another story (it hardly would be realistic to ask $O$ to report infinitely many numerators and denominators). Note, anyhow, that from the discussion accompanying the previous example it is immediately seen that at a given $x$ we are not interested in all $a_\omega$, $b_\omega$; what in fact we are interested in are the active at $x$ numerator and denominator, i.e., either (any) $b_\omega(\cdot)$ such that $b_\omega(x) \leq 0$, if such a denominator exists, or those $a_\omega(\cdot)$, $b_\omega(\cdot)$ with the largest ratio at the point $x$, if all the denominators at $x$ are positive. Assume therefore that we know $G$ in advance and have an access to an oracle $O$ which, given on input $x \in \text{int} G$, reports the values and subgradients of the functional constraints $g_j$ at $x$ and, besides this, tells us (at least in the case when all $g_j$ are negative at $x$) whether all the denominators $b_\omega(x)$, $\omega \in \Omega$, are positive; if it is the case, then the oracle returns the numerator and the denominator with the largest ratio at $x$, otherwise returns the denominator which is nonpositive at $x$. Looking at the construction of $O_A$ given in the previous example we immediately conclude that in our now situation we again can imitate a compatible with (A) oracle $O_A$ for problem (3.6.33).

In fact the "semidefinite fractional problem" we are discussing possesses interesting applications; let me introduce one of them which is of extreme importance for modern Control Theory - the Generalized Eigenvalue problem. The problem is as follows: given two affine functions, $A(x)$ and $B(x)$, taking values in the space of symmetric $m \times m$ matrices ("affine"
means that the entries of the matrices are affine functions of \( x \), minimize, with respect to \( x \), the Rayleigh ratio

\[
\max_{\omega \in \mathbb{R}^m \setminus \{0\}} \frac{\omega^T A(x) \omega}{\omega^T B(x) \omega}
\]

of the quadratic forms associated with these matrices under the constraints that \( B(x) \) is positive definite (and, possibly, under additional convex constraints on \( x \)). In other words, we are looking for a pair \((x, \lambda)\) satisfying the constraints

\[ B(x) \text{ is positive definite} \quad , \lambda B(x) - A(x) \text{ is positive semidefinite} \]

and additional constraints

\[ g_j(x) \leq 0, \ j = 1, ..., m, \ x \in G \subset \mathbb{R}^n \]

\((g_j \text{ are convex and continuous on the solid } G)\) and are interested in the pair of this type with the smallest possible \( \lambda \).

The Generalized Eigenvalue problem (the origin of the name is that in the particular case when \( B(x) \equiv I \) is the unit matrix we come to the problem of minimizing, with respect to \( x \), the largest eigenvalue of \( A(x) \)) can be immediately written down as a "semidefinite fractional problem"

\[
\text{minimize} \ \max_{\omega \in \Omega} \frac{\omega^T A(x) \omega}{\omega^T B(x) \omega} \quad \text{s.t.} \ g_j(x) \leq 0, \ j = 1, ..., m, \ \omega^T B(x) \omega > 0, \ \omega \in \Omega, \ x \in G; \ (3.6.35)
\]

here \( \Omega \) is the unit sphere in \( \mathbb{R}^m \). Note that the numerators and denominators in our "objective fractions" are affine in \( x \), as required by our general assumptions on fractional problems.

Assume that we are given \( G \) in advance, same as the data identifying the affine in \( x \) matrix-valued functions \( A(x) \) and \( B(x) \), and let we have an access to a first-order oracle providing us with local information on the "general type" convex constraints \( g_j \). Then it is not difficult to decide, for a given \( x \), whether \( B(x) \) is positive definite, and if it is not the case, to find \( \omega \in \Omega \) such that the denominator \( \omega^T B(x) \omega \) is nonpositive at \( x \). Indeed, it suffices to compute \( B(x) \) and to subject the matrix to the Cholesky factorization (I hope you know what it means). If factorization is successful, we find a lower-triangular matrix \( Q \) with nonzero diagonal such that

\[ B(x) = QQ^T, \]

and \( B(x) \) is positive definite; if the factorization fails, then in course of it we automatically meet a unit vector \( \omega \) which "proves" that \( B(x) \) is not a positive semidefinite, i.e., is such that \( \omega^T B(x) \omega \leq 0 \). Now, if \( B(x) \), for a given \( x \), is positive semidefinite, then to find \( \omega \) associated with the largest at \( x \) of the fractions

\[
\frac{\omega^T A(x) \omega}{\omega^T B(x) \omega}
\]

is the same as to find the eigenvector of the (symmetric) matrix \( Q^{-1} A(x)(Q^T)^{-1} \) associated with the largest eigenvalue of the matrix, \( Q \) being the above Cholesky factor of \( B(x) \) (why?); to find this eigenvector, this is a standard Linear Algebra routine.
Thus, any technique which allows to solve \((f)\) under assumption (A) immediately implies a numerical method for solving the Generalized Eigenvalue problem.

It is worthy to explain what is the ”control source” of Generalized Eigenvalue problems. Let me start with the well-known issue - stability of a linear differential equation

\[ z'(t) = \alpha z(t) \]

\((z \in \mathbb{R}^s)\). As you for sure know, the maximal ”growth” of the trajectories of the equation as \(t \to \infty\) is predetermined by the eigenvalue of \(\alpha\) with the largest real part, let this part be \(\lambda\); namely, all the trajectories admit, for any \(\epsilon > 0\), the estimate

\[ |z(t)|_2 \leq C \epsilon \exp\{(\lambda + \epsilon)t\}|z(0)|_2, \]

and vice versa: from the fact that all the trajectories admit an estimate

\[ |z(t)|_2 \leq C \exp\{at\}|z(0)|_2 \tag{3.6.36} \]

it follows that \(a \geq \lambda\).

There are different ways to prove the above fundamental Lyapunov Theorem, and one of the simplest is via quadratic Lyapunov functions. Let us say that a quadratic function \(z^T Lz\) \((L\) is a symmetric positive definite \(s \times s\) matrix) proves that the decay rate of the trajectories is at most \(a\), if for any trajectory of the equation one has

\[ z^T(t)Lz'(t) \leq az^T(t)Lz(t), \quad t \geq 0; \tag{3.6.37} \]

if it is the case, then of course

\[ \frac{d}{dt} \ln \left( z^T(t)Lz(t) \right) \leq 2a \]

and, consequently,

\[ \left( z^T(t)Lz(t) \right)^{1/2} \leq \exp\{at\} \left( z^T(0)Lz(0) \right)^{1/2}, \]

which immediately results in an estimate of the type \((3.6.36)\). Thus, any positive definite symmetric matrix \(L\) which satisfies, for some \(a\), relation \((3.6.37)\) implies an upper bound \((3.6.36)\) on the trajectories of the equation, the upper bound involving just this \(a\). Now, what does it mean that \(L\) satisfies \((3.6.37)\)? Since \(z'(t) = \alpha z(t)\), it means exactly that

\[ z^T(t)Lz(t) = \frac{1}{2} z^T(t)(L\alpha + \alpha^T L)z(t) \leq az^T(t)Lz(t) \]

for all \(t\) and all trajectories of the equation; since \(z(t)\) can be an arbitrary vector of \(\mathbb{R}^s\), the latter inequality means that

\[ 2aL - (\alpha^T L + L\alpha) \text{ is positive semidefinite}. \tag{3.6.38} \]
Thus, any pair comprised of a real \( a \) and a positive definite symmetric \( L \) satisfying \((3.6.38)\) results in upper bound \((3.6.36)\); the best (with the smallest possible \( a \)) bound \((3.6.36)\) which can be obtained on this way is given by the solution to the problem

\[
\text{minimize } a \text{ s.t. } L \text{ is positive definite, } 2aL - (\alpha^T L - L\alpha) \text{ is positive definite;}
\]

this is nothing but the Generalized Eigenvalue problem with \( B(L) = 2L, A(L) = \alpha^T L + L\alpha \) and no additional constraints on \( x \equiv L \). And it can be proved that the best \( a \) given by this construction is nothing but the largest of the real parts of the eigenvalues of \( \alpha \), so that in the case in question the approach based on quadratic Lyapunov functions and Generalized Eigenvalue problems results in complete description of the equipped of the trajectories as \( t \to \infty \).

In fact, of course, what was said is of no "literal" significance: what for should we solve a Generalized Eigenvalue problem in order to find something which can be found by a direct computation of the eigenvalues of \( \alpha \)? The indicated approach becomes meaningful when we come from our simple case of a linear differential equation with constant coefficients to a much more difficult (and more important for practice) case of a differential inclusion. Namely, assume that we are given a multivalued mapping \( z \mapsto Q(z) \subset \mathbb{R}^s \) and are interested in bounding the trajectories of the differential inclusion

\[
z'(t) \in Q(z(t)), \ t \geq 0; \tag{3.6.39}
\]

such an inclusion may model, e.g., a time-varying dynamic system

\[
z'(t) = \alpha(t)z(t)
\]

with certain unknown \( \alpha(\cdot) \). Assume that we know finitely many matrices \( \alpha_1, \ldots, \alpha_M \) such that

\[
Q(z) = \text{Conv} \{ \alpha_1 z, \ldots, \alpha_M z \}
\]

(e.g., we know bounds on entries of \( \alpha(t) \) in the above time-varying system). In order to obtain an estimate of the type \((3.6.36)\), we again may use a quadratic Lyapunov function \( z^T Lz \): if for all trajectories of the inclusion one has

\[
z(t)Lz'(t) \leq az^T(t)Lz(t),
\]

or, which is the same, if

\[
z^T Lz' \leq az^T Lz \ \forall (z \in \mathbb{R}^s, z' \in Q(z)) \tag{3.6.40}
\]

then, same as above,

\[
(z^T(t)Lz(t))^{1/2} \leq \exp\{at\}(z^T(0)Lz(0))^{1/2}
\]

for all trajectories.

Now, the left hand side in \((3.6.40)\) is linear in \( z' \), and in order for the inequality in \((3.6.40)\) to be satisfied for all \( z' \in Q(z) \) it is necessary and sufficient to have \( z^T L\alpha_i z \equiv
\[ \frac{1}{2} \left( z^T (\alpha_i^T L + L \alpha_i) z \right) \leq z^T L z \text{ for all } z \text{ and all } i, \text{ i.e., to ensure positive semidefiniteness of the matrices } 2aL - (\alpha_i^T L + L \alpha_i), \text{ } i = 1, ..., M. \] By setting

\[ B(L) = \text{Diag}(2L, ..., 2L), \quad A(L) = \text{Diag}(\alpha_1^T L + L \alpha_1, ..., \alpha_M^T L + L \alpha_M) \]

we convert the problem of finding the best quadratic Lyapunov function (i.e., that one with the best associated decay rate \( a \)) into the Generalized Eigenvalue problem

\[
\text{minimize } a \quad \text{s.t. } B(L) \text{ positive definite, } aB(L) - A(L) \text{ positive semidefinite}
\]

with no additional constraints on \( x \equiv L \).

Note that in the case of differential inclusions (in contrast to that one of equations with constant coefficients) the best decay rate which can be demonstrated by a quadratic Lyapunov function is not necessarily the actually best decay rate possessed by the trajectories of the inclusion; the trajectories may behave themselves better than it can be demonstrated by a quadratic Lyapunov function. This shortcoming, anyhow, is compensated by the fact that the indicated scheme is quite tractable computationally; this is why it becomes now one of the standard tools for stability analysis and synthesis.\(^3\)

It is worthy to add that the von Neumann problem is a very specific case of a Generalized Eigenvalue problem (make the entries of \( Ax \) and \( Bx \) the diagonal entries of diagonal matrices).

I have indicated a number of important applications of (\( f \)); it is time now to think how to solve the problem. There is no difficulty in applying to the problem the cutting plane scheme.

The Cutting Plane scheme for problem (\( f \)):

\begin{itemize}
  \item **Initialization.** Choose a solid \( G_0 \) which covers \( G_f \).
  \item \( i \)-th step, \( i \geq 1 \). Given a solid \( G_{i-1} \) (the previous localizer), choose \( x_i \in \text{int} \ G_i \);
  \item and call the oracle \( O_A, x_i \) being the input.
  \item Given the answer \( e_i \equiv e_{x_i} \) of the oracle,
  \item - call step \( i \) productive if the oracle says that \( x_i \in \text{int} \ G_f \), and call the step non-productive otherwise;
  \item - check whether \( e_i = 0 \) (due to (A), this may happen only at a productive step); if it is the case, terminate, \( x_i \) being the result found by the method. Otherwise
    \item - define \( i \)-th approximate solution to (\( f \)) as the best, in terms of the values of \( f \), of the search points \( x_j \) associated with the productive steps \( j \leq i \);
    \item - set
      \[ \tilde{G}_i = \{ x \in G_{i-1} \mid (x - x_i)^T e_i \leq 0 \}; \]
    \item - embed the intermediate localizer \( \tilde{G}_i \) into a solid \( G_i \) and loop.
\end{itemize}

\(^3\)For “user” information look at the MATLAB LMI Control Toolbox manual.
The presented scheme defines, of course, a family of methods rather than a single method. The basic implementation issues, as always, are how to choose $x_i$ in the interior of $G_{i-1}$ and how to extend $G_i$ to $G_i$; here one may use the same tactics as in the Center of Gravity or in the Ellipsoid methods. An additional problem is how to start the process (i.e., how to choose $G_0$); this issue heavily depends on a priori information on the problem, and here we hardly could do any universal recommendations.

Now, what can be said about the rate of convergence of the method? First of all, we should say how we measure inaccuracy. A convenient general approach here is as follows. Let $x \in G_f$ and let, for a given $\varepsilon \in (0, 1)$,

$$G_f^\varepsilon = x + \varepsilon (G_f - x) = \{ y = (1 - \varepsilon)x + \varepsilon z \mid z \in G_f \}$$

be the image of $G_f$ under the similarity transformation which shrinks $G_f$ to $x$ in $1/\varepsilon$ times. Let

$$f_x(\varepsilon) = \sup_{y \in G_f^\varepsilon} f(y).$$

The introduced quantity depends on $x$; let us take the infimum of it over $x \in G_f$:

$$f^*(\varepsilon) = \inf_{x \in G_f} f_x(\varepsilon).$$

By definition, an $\varepsilon$-solution to $f$ is any point $x \in G_f$ such that

$$f(x) \leq f^*(\varepsilon).$$

Let us motivate the introduced notion. The actual motivation is, of course, that the notion works, but let us start with a kind of speculation. Assume for a moment that the problem is solvable, and let $x^*$ be an optimal solution to it. One hardly could argue that a point $\bar{x} \in G_f$ which is at the distance of order of $\varepsilon$ of $x^*$ is a natural candidate on the role of an $\varepsilon$-solution; since all points from $G_f^{x^*}$ are at the distance at most $\varepsilon \text{Diam}(G_f)$ from $x^*$, all these points can be regarded as $\varepsilon$-solutions, in particular, the worst of them (i.e., with the largest value of $f$) point $x^*(\varepsilon)$. Now, what we actually are interested in are the values of the objective; if we agree to think of $x^*(\varepsilon)$ as of an $\varepsilon$-solution, we should agree that any point $x \in G_f$ with $f(x) \leq f(x^*(\varepsilon))$ also is an $\varepsilon$-solution. But this latter property is shared by any point which is an $\varepsilon$-solution in the sense of the above definition (look, $f(x^*(\varepsilon))$ is nothing but $f_{x^*}(\varepsilon)$), and we are done - our definition is justified!

Of course, this is nothing but a speculation. What might, and what might not be called a good approximate solution, this cannot be decided in advance; the definition should come from the real-world interpretation of the problem, not from inside the Optimization Theory. What could happen with our definition in the case of a "bad" problem, it can be seen from the following example:

$$\begin{align*}
\text{minimize} & \quad \frac{x}{10^{-20} + x}, \\
& \quad x \in G_f = [0, 1].
\end{align*}$$

Here in order to find a solution with the value of the objective better than, say, $1/2$ (note that the optimal value is $0$) we should be at the distance of order $10^{-20}$ of the exact solution.
LECTURE 3. METHODS WITH LINEAR CONVERGENCE

For our toy problem it is immediate, of course, to indicate the solution exactly, but think what happens if the same effect is met in the case of a multidimensional and nonpolyhedral \( G_f \). We should note, anyhow, that the problems like that one just presented are "intrinsically bad" (what is the problem?); in ”good” situations our definition does work:

Exercise 3.6.3 # Prove that
1) if the function \( f \) involved into \((f)\) is convex and continuous on \( G_f \), then any \( \varepsilon \)-solution \( x \) to the problem satisfies the inequality

\[ f(x) - \min_{G_f} f \leq \varepsilon (\max_{G_f} f - \min_{G_f} f), \]

i.e., solves the problem within relative accuracy \( \varepsilon \);
2) if the function \( f \) involved into \((f)\) is Lipschitz continuous on \( G_f \) with respect to certain norm \( |\cdot| \), \( L_f \) being the corresponding Lipschitz constant, and if \( x \) is a \( \varepsilon \)-solution to \((f)\), then

\[ f(x) - \min_{G_f} f \leq \text{Diam}(G_f)L_f\varepsilon, \]

where \( \text{Diam} \) is the diameter of \( G_f \) with respect to the norm in question.

Now - the end of the story.

Exercise 3.6.4 #* Prove the following statement:
let a cutting plane method be applied to \((f)\), and let \( \text{Size}(\cdot) \) be a size. Assume that for certain \( N \) the method either terminates in course of the first \( N \) steps, or this is not the case, but \( \text{Size}(G_N) \) is smaller than \( \text{Size}(G_f) \). In the first of the cases the result produced by the method is a minimizer of \( f \) over \( G_f \), and in the second case the \( N \)-th approximate solution \( \bar{x}_N \) is well-defined an is an \( \varepsilon' \)-solution to \((f)\) for any

\[ \varepsilon' > \frac{\text{Size}(G_N)}{\text{Size}(G_f)}. \]

Exercise 3.6.5 # Write a code implementing the Ellipsoid version of the Cutting Plane scheme for \((f)\). Use the code to find the best decay rate for the differential inclusion

\[ z'(t) \in Q(z) \subset \mathbb{R}^3, \]

where

\[ Q(z) = \text{Conv} \{\alpha_1z, ..., \alpha_Mz\} \]

and \( \alpha_i, i = 1, 2, ..., M = 2^6 = 64 \), are the vertices of the polytope

\[ P = \left\{ \begin{pmatrix} 1 & p_{12} & p_{13} \\ p_{21} & 1 & p_{23} \\ p_{31} & p_{32} & 1 \end{pmatrix} \mid |p_{ij}| \leq 0.1 \right\}. \]
3.6.2 The method of outer simplex

Looking at the Ellipsoid method, one may ask: why should we restrict ourselves to ellipsoids and not to use localizers of some other "simple" shape? This is a reasonable question. In fact all properties of the "ellipsoidal" shape which were important in the Ellipsoid version of the cutting plane scheme were as follows:

(a) all ellipsoids in \( \mathbb{R}^n \) are affine equivalent to each other, and, in particular, are affine equivalent to certain "standard" ellipsoid, the unit Euclidean ball; therefore in order to look what happens after a cut passing through the center of an ellipsoid, it suffices to study the simplest case when the ellipsoid is the unit Euclidean ball;

(b) the part of the unit Euclidean ball which is cut off the ball by a hyperplane passing through the center of the ball can be covered by an ellipsoid of volume less than that one of the ball.

Now, can we point out another "simple shape" which satisfies the above requirements? The natural candidate is, of course, a simplex. All \( n \)-dimensional simplices are affine equivalent to each other, so that we have no problems with (a); the only bottleneck for simplices could be (b). The below exercises demonstrate that everything is ok with (b) as well.

Let us start with considering the standard simplex (which now plays the role of the unit Euclidean ball).

**Exercise 3.6.6** Let
\[
\Delta = \{ x \in \mathbb{R}^n \mid x \geq 0, \sum_{i=1}^{n} x_i \leq 1 \}
\]
be the standard \( n \)-dimensional simplex in \( \mathbb{R}^n \), let
\[
c = \left( \frac{1}{n+1}, \ldots, \frac{1}{n+1} \right)^T
\]
be the barycenter (\( \equiv \) the center of gravity, see exercise 1.5) of the simplex, let
\[
g = (\gamma_1, \ldots, \gamma_n)^T \geq 0
\]
be a nonzero nonnegative vector such that
\[
g^T c = 1,
\]
and let
\[
\tilde{\Delta} = \{ x \in \Delta \mid g^T x \leq g^T c = 1 \}.
\]
Prove that if \( \vartheta \in [0, 1] \), then the simplex \( \Delta_{\vartheta} \) with the vertices \( v_0 = 0 \),
\[
v_i = (1 - \vartheta + \vartheta \gamma_i)^{-1} e_i, \quad i = 1, \ldots, n,
\]
\((e_i \text{ are the standard orths of the axes})\) contains \( \tilde{\Delta} \).

Prove that under appropriate choice of \( \vartheta \) one can ensure that
\[
\text{Vol}_n(\Delta_{\vartheta}) \leq \chi^n(n)\text{Vol}_n(\Delta), \quad \chi^n(n) = \left( 1 + \frac{1}{n^2 - 1} \right)^{n-1} \left( 1 - \frac{1}{n+1} \right) \leq 1 - O(n^{-2}).
\]
(3.6.41)
Exercise 3.6.7 Let $D$ be a simplex in $\mathbb{R}^n$, $v_0, ..., v_n$ be the vertices of $D$,

$$w = \frac{1}{n+1}(v_0 + ... + v_n)$$

be the barycenter of $D$ and $g$ be a nonzero linear functional. Prove that the set

$$\tilde{D} = \{x \in D \mid (x-w)^T g \leq 0\}$$

can be covered by a simplex $D'$ such that

$$\text{Vol}_n(D') \leq \chi(n)\text{Vol}_n(D),$$

with $\chi(n)$ given by (3.6.41). Based on this observation, construct a cutting plane method for convex problems with functional constraints where all localizers are simplices. What should be the associated size? What is the complexity of the method?

Hint: without loss of generality we may assume that the linear form $g^T x$ attains its minimum over $D$ at the vertex $v_0$ and that $g^T (w - v_0) = 1$. Choosing $v_0$ as our new origin and $v_1 - v_0, ..., v_n - v_0$ as the orths of our new coordinate axes, we come to the situation studied in exercise 3.6.6.

Note that the progress in volumes of the subsequent localizers in the method of outer simplex (i.e., the quantity $\chi(n) = 1 - O(n^{-2})$) is worse than that one $\kappa(n) = 1 - O(n^{-1})$ in the Ellipsoid method. It does not, anyhow, mean that the former method is for sure worse than the latter one: in the Ellipsoid method, the actual progress in volumes always equals to $\kappa(n)$, while in the method of outer simplex the progress depends on what are the cutting planes; the quantity $\chi(n)$ is nothing but the worst case bound for the progress, and the latter, for a given problem, may happen to be more significant.