STABILIZATION AND ACCELERATION OF THE ELLIPSOID METHOD FOR THE MINIMIZATION OF CONVEX NONSMOOTH FUNCTIONS

GYÖRGY SONNEVEND

Dept. of Numerical Mathematics, Eötvös L. University,
1088 Budapest, Múzeum krt. 6–8.

(Received October 4, 1984 – revised August 27, 1986)

Abstract. New algorithms are proposed with corresponding error estimations for finding the minimum f^* of an arbitrary convex function $f: \mathbf{R}^n \to \mathbf{R}^1$ (having a known Lipschitz constant over a ball of $\mathbf{R}^n$ which is known to contain at least one minimum point of f). It is supposed that – at each step of the algorithm – the value of f and one of its subgradients can be evaluated (at an arbitrary point of $\mathbf{R}^n$). These algorithms represent ways of acceleration and stabilization of the "ellipsoid method" (due to Shor, Yudin-Nemirowskii, 1977). The main part of this paper is devoted to the construction and error estimation of a *stabilized* version of this method which turns out to share all the positive features of the original method and allows to reduce significantly the accuracy required for the computations and function (gradient) evaluations. For $n = 1$ an algorithm is obtained whose convergence rate is $9^{-1/3}$, i. e. less than 1/2.

1. Introduction

The ellipsoid method has been proposed by Shor (see [2]) for the solution of the problem of unconstrained minimization of convex nonsmooth functions. The same method has been proposed also in [11] as an "implementable" version of the method of centers of gravity, see e. g. [8] and our remark at the end of Section 3.

Later L. G. Khachian [4] used this method in order to prove that linear programming problems

$$\inf\{\langle c, x\rangle \mid Ax \leq b\},\ x, c \in \mathbf{R}^n,\ A \in \mathbf{R}^{mxn},\ b \in \mathbf{R}^m \tag{1.1}$$

with rational coefficients can be solved in "polynomial" time (in the number of digits in the coefficients). Roughly speaking, Khachian showed, see. e. g. [10], that if the rational coefficients are transformed into integer ones and

$$L_0 := \sum_i \sum_j \log_2(|A_{ij}|+1) + \sum_i \log_2(|b_i|+1) + \log_2(nm+1),$$

then $O(n^3(n+m)L_0)$ arithmetical operations $(+, -, \times, \div, \sqrt{\ })$ suffice to find the exact solution, if it exists (it is assumed that all such arithmetical operations are kept exact in using no more than $23L_0$ binary digits before and $38nL_0$ after the decimal point). This opened a way of constructing "polynomial" algorithms for a number of combinatorial optimization problems.

A serious drawback of the (originally proposed) algorithm lies in its *instability* in the following sense. The volumes of the succesively constructed ellipsoids – which serve as sets of localization of possible minimum points – tend to zero, their diameters however may tend to infinity which leads to bad conditioning (oscillations) long before the minimum is reached within sufficient accuracy (for large values of L_0), see (2.15) below. This instability has been noticed, i. e. observed in computer tests by many researchers, and has lead – after a very enthusiastic beginning – to an exaggerated discarding of the original idea on which the method is based. Implicitly this instability is behind the enormous accuracy, $\text{const} \cdot \exp(-nL_0)$, required for the arithmetical operations in the original method.

We shall see that by finding a remedy for this "internal" instability other, "external" instabilities can also be cured (mildened) and a realistic algorithm with error estimation can be obtained.

In fact there are other reasons for not proposing the ellipsoid method for the solution of large ($n \gg 1$) linear programming problems: it is slowly converging (when compared e. g. with the simplex method, for problems where m is not too large); it is not clear whether the method can easily incorporate additional (e. g. "sparsity") structures (decomposability) being often present in such problems, or not.

For unstructured problems with many constraints, $m \gg n$, or more generally for the solution of the general convex programming problem,

$$\inf \{f_0(x) \mid f_i(x) \leq 0, \quad i = 1, 2, \ldots, m\} = f_0^*, \tag{1.2}$$

where the functions $f_0, f_1, \ldots, f_m$ are assumed only to be convex over $\mathbf{R}^n$, the stabilized versions of the ellipsoid method (see below) seem to be competitive (for not too small values of n). More precisely this can be expected for such classes of problems where the values and at least one of the subgradients of each function $f_0, f_1, \ldots, f_m$ can be computed exactly at an arbitrary point of $\mathbf{R}^n$. One can use the method of exact penalty functions (see Section 2) to transform (1.2) into a similar problem without constraints, i. e. when we have only one function $f_0 = f$, $m = 0$.

In Section 2 we present the description and an error estimation for a new, externally stabilized version of the original method (a new method of acceleration will be given in Section 5). The internal stabilization method and its estimation are presented in Sections 3 and 4. We show that, by the introduction of suitably constructed stabilization steps, the instability of the original method can be removed so that essentially the same number of arithmetical operations (and function evaluations) are needed as for the original method to reach a prescribed (small) uncertainty ε in the value of the minimum. The accuracy required when performing these, as well as the measure-

ments of $g(x) \in \partial f(x)$ is not "greater" (for ε small enough) than $\varepsilon^{10} n^{-4} \cdot \text{const}$ (thus is "polynomial" in ε, n and independent of m, the number of constraints in (1.1)), this is one of the main results of this paper, see the end of Section 2.

The second (numerical) problem – the solution of which yields the basis of our internal stabilization method – consists in constructing (approximating) an ellipsoid of (minimal) small volume and *small* diameter containing the intersection of a ball (ellipsoid) and an (other) ellipsoid. The latter problem has independent interest in other areas e. g. for the approximation of reachable sets (or sets of localization) in control systems with bounded gontrols (resp. disturbances). In the paper [13] test results demonstrate the gain from stabilization.

Remark. If the values of f can be measured only within a given accuracy ε_0 (and the values of grad f are not accessible directly) then the straightforward generalizations of the ellipsoid method, see [11, Ch. VIII], exhibit an other instability: one with respect to ε_0 (this instability is different in its nature from the two instabilities studied here). For the construction of methods with stability constants (the ratio of the minimal achievable uncertainty in f^* to ε_0 for $N \to \infty$) arbitrarily near to one see [1]. In Section 5 we present a new algorithm for the solution of the one-dimensional problem whose rate of convergence is $9^{-1/3}$, i. e. less than 2^{-1}, This algorithm shows the way (by the more full use of the available information, i. e. measured values of f and $g \in \partial f$ and the convexity of f) for the construction of "ellipsoid" algorithms essentially faster than the original ellipsoid algorithm for arbitrary values of the dimension n.

2. Description of an accelerated version of the original method and an error estimation for it

We shall deal with the problem of finding the minimal value f^* of a convex function $f: \mathbf{R}^n \to \mathbf{R}^1$ under the assumption (initial information) that a ball of (finite) radius R contains at least one extremal point of f, i. e. that

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y), \text{ for all } x, y \in \mathbf{R}^n, \ 0 \leq \lambda \leq 1 \tag{2.1}$$

$$G_R \cap X^*(f) \neq \emptyset, \text{ where } G_R: \{x \mid \|x\| \leq R, \ x \in \mathbf{R}^n\}, \tag{2.2}$$

$$X^*(f) = \{x \mid f(x) = f^*, \ x \in \mathbf{R}^n\}, \ f^* = \inf \{f(x) \mid x \in \mathbf{R}^n\}. \tag{2.3}$$

We assume also that a Lipschitz constant L is known for f over the ball G_{Rb_n}, where $b_n \geq 1$ is a constant defined below, see (2.18)

$$|f(x) - f(y)| \leq L\|x - y\|, \text{ for all } x, y \in G_{Rb_n}. \tag{2.4}$$

The class of convex functions satisfying (2.2) and (2.4) will be denoted by $F(G_R, L)$. When not stated otherwise, norms of vectors resp. symmetric matrices will be always the Euclidean resp. spectral ones.

The problem is to estimate the value of f^* based on the values of $f(x_i)$ and $g(x_i) \in \partial f(x_i)$, $i = 1, \ldots, N$, which should be computed sequentially.

By definition, the knowledge of $f(x)$ and $g(x)$ at a point $x = x_j, j = 1, \ldots, N$ tell us that

$$f(z) \geq f(x) + \langle g(x).\ z - x\rangle, \text{ for all } z \in \mathbf{R}^n. \tag{2.5}$$

The aim is to construct an algorithm for the sequential choice of the x_i's, $i = 1, \ldots, N$, so that – for a prefixed accuracy ε – the inequality

$$\varepsilon \geq \varepsilon(N, f) := \inf \{|h^* - \min_{1 \leq i \leq N} f(x_i)| h \in F(G_R, L),\ h(x_j) = f(x_j),$$

$$g(x_j) \in \partial h(x_j),\ j = 1, \ldots, N\}, \tag{2.6}$$

is satisfied with a possibly small value of N.

For a more detailed definition of algorithms for function minimization we refer to [8], [11]. Here we shall allow to use (i. e. know) – at each step k of the algorithm, when x_{k+1} is chosen – the values of R, L and ε; what is however more important: we shall not need to keep in memory all the computed values $f(x_j)$, $g(x_j)$, $1 \leq j \leq k$, and we shall have to perform only $O(n^2)$ arithmetical operations in each step.

It is known that the problem (1.2) can be "reduced" to the problem of unconstrained minimization of the convex function (an exact penalty function)

$$f(x) := \Phi_N(x) = f_0(x) + N' \max \{0, f_1(x), \ldots, f_m(x))\},\ x \in \mathbf{R}^n, \tag{2.7}$$

when N' is large enough: $N' \geq K$ (if $K < \infty$, which is a condition of well-posedness for the problem (1.2)), where

$$-K := \inf_{\lambda \geq 0} \frac{V(\lambda) - V(0)}{\lambda},\ V(\lambda) := \inf \{f_0(x) | f_i(x) \leq \lambda,\ i = 1, \ldots, m\},$$

see e. g. [5, § 8]. We are not interested in studying here the above reduction (which amounts to the proper, sequential guessing of K) we remark only that the problems (1.1), (1.2), (1.3) and (1.7) are – to some extent – equivalent (because each convex function is the supremum of a family of linear functions).

The idea of the ellipsoid method is to confine the set $G_R \cap X^*(f)$ into successively constructed ellipsoids E_i of geometrically decreasing volumes $E_0 = G_R, x_0 = 0, E_i = E(x_i, A_i), i = 1, \ldots, N$, where

$$E(x, A) := \{z | \langle A^{-1}(z - x), z - x\rangle \leq 1\},\ x \in \mathbf{R}^n,\ 0 < A^* = A \in \mathbf{R}^{n \times n}.$$

Note that – denoting by $A^{1/2}$ the positive square root of A –

$$E(x, A) = x + A^{1/2} G_1,\ \det A^{1/2} = \text{vol}\,(E(x, A))/k_n, k_n = \frac{\pi^{\frac{n}{2}}}{\Gamma\left(\frac{n}{2} + 1\right)},$$

where G_1 is the unit ball in $\mathbf{R}^n$, vol (K) is the volume of K.

Here – in order to provide an accelerated version of the original method – we shall construct E_{k+1} (i. e. x_{k+1} and A_{k+1}) based on the knowledge of E_k

$$g_k = g(x_k) \in \partial f(x_k), \text{ and } f(x_{i(k)}),$$

where

$$f(x_{i(k)}) = \min \{f(x_j) \mid 0 \leq j \leq k\}, \tag{2.8}$$

so that either

$$\frac{\operatorname{vol}(E_{k+1})}{\operatorname{vol} E_k} \leq h_n := \frac{n^n}{n+1}(n^2-1)^{-\frac{n-1}{2}} < e^{-\frac{1}{2(n+1)}} \tag{2.9}$$

or

$$\langle A_k g(x_k), g(x_k)\rangle \leq \varepsilon_1 \tag{2.10}$$

(in the latter case the algorithm stops at x_k).

Below we shall choose ε_1 in a special way depending on ε_0, ε, L and R, (for $\varepsilon_0 \neq 0$, we shall introduce an additional stopping criterion), and the analysis of the implications of (2.10) will be crucial.

$$\text{Let } D_k := \Big(f(x_k) - \min_{1 \leq j \leq k} f(x_j)\Big)\langle g_k, A_k g_k\rangle^{-1/2} \tag{2.11}$$

(note that $D_k \leq 1$ by the assumption (2.2), see below),

$$x_{k+1} := x_k - \frac{D_k n+1}{n+1} A_k g_k \langle A_k g_k, g_k\rangle^{-1/2}, \ (x_0 = 0, A_0 = R^2 I) \tag{2.12}$$

$$A_{k+1} := \frac{n^2(1-D_k^2)}{n^2-1}\left(A_k - \frac{2(D_k n+1)}{(n+1)(D_k+1)} \frac{A_k g_k (A_k g_k)^*}{\langle A_k g_k, g_k\rangle}\right). \tag{2.13}$$

The following lemma shows that $E_{k+1} = E(x_{k+1}, A_{k+1})$ will be a set of localization for $X^*(f) \cap G_R$ (corresponding to the informations gathered till step $k+1$) and that

$$\frac{\operatorname{vol}(E_{k+1})}{\operatorname{vol}(E_k)} = h_n(1-D_k)(1-D_k^2)^{\frac{n-1}{2}}.$$

We note that the original method, see [2], [3], [11], is obtained when we set $D_k \equiv 0$ in the formulas (2.12), (2.13). The appearence of the expression $\langle A_k g_k, g_k\rangle$ in the denominators of the update formulas dictates the introduction of a stopping criterion like (2.10) in any "stable" realization of the algorithm: we shall see that the choice (2.10) has special advantages.

Lemma 1. *Let x, g be arbitrary vectors in $\mathbf{R}^n$, $0 < A = A^* \in \mathbf{R}^{n \times n}$ and $0 \leq d \leq 1$*

$$E' = \{z \mid \langle g, z\rangle \leq \langle g, x\rangle - d\langle g, Ag\rangle^{1/2}\} \cap E(x, A),$$

then E' can be included into the ellipsoid $E(x, A')$

$$x' := x - \frac{dn+1}{n+1}\frac{Ag}{\sqrt{\langle Ag, g\rangle}},$$

$$A' := \frac{n^2(1-d^2)}{n^2-1}\left(A - \frac{2(dn+1)}{(n+1)(d+1)}\frac{Ag(Ag)^*}{\langle g, Ag\rangle}\right)$$

so that

$$\text{vol}\,(E(x', A')) = \text{vol}\,E(x, A)h_n(1-d)(1-d^2)^{\frac{n-1}{2}}.$$

In fact $E(x', A')$ is the ellipsoid of smallest volume containing the set E'. For completeness we reproduce here the simple proof given e. g. in [10].

Proof. One can assume that A is the identity matrix, i. e. $E(x, A)$ is the unit ball, then one has to compute the minimum of $v^{-1}w^{-(n-1)}$, i. e. the maximum K^* of the convex function $v^2w^{2n-2} = k(v, w)$ under the constraints (where h corresponds to the location of the centre of the new ellipsoid)

$$v^2(h-t)^2 + w^2(1-t^2) \leq 1, \text{ for all } 1 \geq t \geq d.$$

One has to choose $0 < h < 1$ so that $k^* = k^*(h, d)$ becomes maximal. This results in

$$k^*(h, d) = (1-h)^{-2}(1-d^2)^{1-n}\left(1 - \left(\frac{h-d}{1-h}\right)^2\right)^{n-1}$$

$$h^* = \frac{dn+1}{n+1},\ v^* = \frac{n+1}{n(1-d)},\ w^* = \left(\frac{n^2-1}{n^2(1-d^2)}\right)^{1/2}. \quad \square$$

It is interesting to note that Lemma 1 is true for all values $1 \geq d \geq -n^{-1}$, and it is true for $d = -n^{-1}$ that the ellipsoid $E(x', A')$ is identical with $E(x, A)$.

Now Lemma 1 is used in the derivation of (2.11) – (2.13) based on the observation that the computation of $f(x_k)$ and $g(x_k)$ allows – by the convexity of f, see (2.5) – to localize the set of possible minimumpoints (within G_R) into the intersection

$$E(x_k, A_k) \cap \{z \mid \langle g(x_k), z - x_k\rangle \leq -(f(x_k) - \min_{0 \leq j \leq k} f(x_j))\}.$$

Notice that the original method is not modified in steps k where $f(x_k) = \min\{f(x_j),\ 1 \leq j \leq k\}$. One could, however, propose an acceleration for this case also by using Lemma 1, r-times, for the intersections

$$E(x_k^{i-1}, A_k^{i-1}) \cap \{z \mid \langle g(x_k^{i-1}), z - x_k^{i-1}\rangle \leq f(x_k^i) - \min_{\substack{0 \leq j \leq k \\ i=1, \ldots, r}} f(x_j), \tag{2.14}$$

where $x_k^0 = x_k = x_{k-1}^r$, $A_k = A_k^0 = A_{k-1}^r$, and x_k^i $i = 1, \ldots, r$ are points from among $x_1, \ldots, x_k$.

For example one could set $r = 1$ and define $x_k^1 = x_{j*}$, where

$$f(x_{j*}) = \min_j \{f(x_j) \mid f(x_j) \neq f(x_{i(k)})\}.$$

We shall return to these possibilities in Section 5.

The instability phenomenon referred to in the introduction can be observed now easily. Suppose that for all values of k, $g(x_k)$ has the same direction (this can happen even for a strongly convex quadratic function). Then the matrixes A_k have linearly growing (spectral) norms

$$\|A_k\| = \left(\frac{n^2}{n^2-1}\right)^k, \quad k = 1, 2, \ldots, \tag{2.15}$$

provided that $D_k = 0$ in all steps (which can happen e. g. for a function $f = \max(l_1, l_2)$, where l_i, $i = 1, 2$ are linear functions which are equal to a constant, f^*, on an $(n-1)$ dimensional subspace of $\mathbf{R}^n$).

In order to provide an estimate for the error of the algorithm, i. e. for $\varepsilon(N, f)$, see (2.6), we need the following lemma.

Lemma 2. *Suppose that a (nondegenerate) ellipsoid $E = E(x, A)$ is known to contain the set $X^*(f) \cap G_R$ and let $z \in E$ be a point where $f(z)$ is known and provides a lower bound for the values of f in $G_R \setminus E$, then*

$$f(z) - f^* \leq 2L\sqrt{\lambda_1(A)} \leq 2L(k_n^{-1} \operatorname{vol} E)^{\frac{1}{n}}, \quad k_n = \frac{\pi^{n/2}}{\Gamma\left(\frac{n}{2}+1\right)}, \tag{2.16}$$

holds if $\lambda_1(A)$, the smallest eigenvalue of A, is smaller than $R^2/4$ (here k_n denotes the volume of the unit ball in $\mathbf{R}^n$).

Before the proof we note that $E = E(x_N, A_N)$, $z = x_{i(N)}$ provide – for all values of N – instances for which the conditions of Lemma 2 are fulfilled.

Proof. Let x^* be a point in $X^*(f) \cap G_R$. Since $x^* \in E$, there exist two points w_1 and w_2 on the boundary of E such that $x^* \in [w_1, w_2]$, a segment in the direction of an eigenvector corresponding to the eigenvalue $\lambda_1(A)$ and $\|w_1 - w_2\| \leq d = 2\sqrt{\lambda_1(A)}$.

If at least one of these two points, say w_1, belongs to G_R, then – because of $f(w_1) \geq f(z)$ –

$$f(z) - f^* \leq f(w_1) - f(x^*) \leq L\|w_1 - x^*\| \leq Ld. \tag{2.17}$$

(2.16) is proved. Now suppose that w_1 and w_2 do not belong to G_R, we shall construct two points in G_R, w_1^* and w_2^* such that $\|x^* - w_i^*\| \leq d$ and either w_1^* or w_2^* do not belong to the interior of E. For this let – if $\|x^*\| \geq \frac{d}{2}$ –

$$x' := x^*\left(1 - \frac{d}{2\|x^*\|}\right), \quad w_1^* := x' + \frac{1}{2}d\frac{w_1 - w_2}{\|w_1 - w_2\|}, \quad w_2^* =$$

$$= x' + \frac{1}{2} d \frac{w_2 - w_1}{\|w_1 - w_2\|}.$$

If $\|x^*\| \leq d/2$, then – whenever $d < R$ – already either w_1 or w_2 belongs to G_R. This ends the proof, because either $f(w_1^*)$ or $f(w_2^*)$ is then not smaller than $f(z)$. □

In order to give an error estimation for the algorithm (2.10)–(2.13) we have to formulate a condition which is in close connection with the instability phenomenon mentioned above.

Condition B. Let us suppose that – for some constant b_n –

$$\|A_k\| \leq b_n^2 R^2, \; k = 1, 2, \ldots, (A_0 = R^2 I) \tag{2.18}$$

holds.

In fact the internal stabilization procedure described in Section 3 guarantees that this condition will be satisfied. We prove that $b_n \leq 500$ can be achieved for all values of $n \geq 6$, as well as $b_n < 130$ for sufficiently large values of n. In fact it seems to be true that the value of b_n can be set not greater than 20 without changing the main order relations for the complexities (i. e. the effectivity) for all value of n. We have chosen $b_n = 500$ in order to get – in a simple way – almost the same estimations for the "complexity" of the modified method as for the original method.

A consequence of this condition and (2.2) is that

$$\|x_k\| \leq (b_n + 1)R, \text{ for } k = 0, 1, \ldots . \tag{2.19}$$

Now if for some value of k the inequality (2.10) holds then $\lambda_1(A_k)\|g_k\|^2 \leq \varepsilon_1$, where $\lambda_1(A)$ denotes the smallest eigenvalue of the positive definite matrix A, therefore either

$$\lambda_1(A_k) \leq \frac{R}{L}\sqrt{\varepsilon_1} \text{ or } \|g_k\|^2 \leq \frac{L}{R}\sqrt{\varepsilon_1}. \tag{2.20}$$

The proof of Lemma 2 shows that $\lambda_1(A_N) \leq \frac{R}{L}\sqrt{\varepsilon_1}$ implies that

$$\varepsilon(N, f) \leq 2L\sqrt{\lambda_1(A_N)} \leq 2\sqrt{LR}\varepsilon_1^{1/4}$$

(one can take $\|w_1 - w_2\|^2 = 4\lambda_1(A_k)$. Similarly $\|g_N\|^2 \leq \frac{L}{R}\sqrt{\varepsilon_1}$ implies – if (2.19) holds – that

$$|f(x_N) - f^*| \leq \sqrt{LR}(2 + b_n)\varepsilon_1^{1/4}.$$

Thus we have proved the following theorem.

Theorem 1. *If Condition B is fulfilled and* $\varepsilon_0 = 0$, *then the error of the algorithm* (2.10)–(2.13) *can be estimated by*

$$\varepsilon(N, f) \leq LR \max \{2e^{-\frac{N}{n2(n+1)}}, \; \varepsilon_1^{1/4}(2 + b_n)(LR)^{-1/2}\}. \tag{2.21}$$

If the desired (final) accuracy ε is prescribed i. e. we have to guarantee $\varepsilon(N, f) < \varepsilon$, (with a possibly small value of N), then we choose ε_1 and N so that the two values in the bracket $\{i\}$ in (2.21) be equal to $\varepsilon(LR)^{-1}$, i. e.

$$\varepsilon_1 := \varepsilon^4 (LR)^{-2} (2+b_n)^{-4}. \tag{2.22}$$

It is however not true that the estimation (2.21) remains valid (even only in its essential features) if all the storages, calculations and subgradient evaluations in (2.11)–(2.12) are made within accuracy $\varepsilon_0 = \varepsilon_1$ only. In order to analyse the "external stability problem" we assume that the components of the vectors $g(x) \in \partial f(x)$ can be computed only within (absolute) accuracy ε_0. We also assume that the components of A_k, x_k $(k = 1, 2, \ldots)$ are stored within the same (absolute) accuracy; this in fact will not mean an essentially more severe restriction for ε_0 (as a function of (ε, R, L, n)). The arithmetical operations $+$, $-$, $\times$ over pairs of numbers a, b both not greater than $2b_n^2 R^2 L$ should be performed within the same (absolute) accuracy ε_0, while for divisions a/b this is required only if $a \leq b_n^4 R^4 L^2$ and $b \geq 0{,}5\sqrt{\varepsilon_1}$. Finally we assume that for $b_n^2 R^2 L^2 > a \geq 0{,}5\sqrt{\varepsilon_1}$ the value of $\sqrt{a}$ is computed within absolute accuracy $4^{-1}\varepsilon_1^{-1/2}(2b_n^2 R^2 L^2 n + L\sqrt{n})\varepsilon_0$. One could "normalize" the problem by the transformation $f^*(x) := (LR)^{-1} f(xR)$, then $L^* = R^* = 1$, $\varepsilon^* = (LR)^{-1}\varepsilon$ and a different, simplifying assumption about rounding errors is that the arithmetical operations can be fulfilled as given above (now with $L = R = 1$) and the components of $g(x)$ can be computed within (absolute) accuracy $\varepsilon_0 L^{-1}$. We have kept the values of R, L unnormalized because this better suits the applications.

Theorem 2. *Suppose that we apply the algorithm* (2.10)–(2.13) *in the modified form given by* (2.30), (2.31), (2.35) below. *Then under the rounding assumptions made above and Condition* B *the validity of*

$$\varepsilon_0 \leq \varepsilon^{10} b_n^{-18} n^{-4} k \tag{2.23}$$

with a suitable positive value of k assures – for ε small enough – that

$$\varepsilon(N, f) \leq \varepsilon \text{ for } N \leq N(\varepsilon, n, LR) := \left[2n(n+1)\lg\frac{2RL}{\varepsilon}\right] + 1.$$

Here – for arbitrarily given $n \geq 6$, L, R, one can set $b_n \leq 500$ and

$$k = k(L, R, n, b_n, \varepsilon) \geq k(L, R) > 0,$$

if $\varepsilon \leq \varepsilon(n, L, R, b_n)$. Specially for $L = R = 1$,

$$k(1, 1, n, b_n, \varepsilon) \geq k_0, \text{ if } \varepsilon \leq k_1,$$

for some universal constants k_0, $k_1 > 0$.

From the assumptions made in connection with the normalization $f^* \to f^*$ we obtain that for $\varepsilon \leq LRk_1$ it is enough to assume that

$$\varepsilon_0 \leq \varepsilon^{10}\, b_n^{-18}\, n^{-4}\, L^{-9}\, R^{-10}\, k_0.$$

The remarkable in the estimation (2.23) is the weak, "polynomial" dependence of ε_0 on ε and n.

Let's compare this result with theorem (5.16) in [6] where in essentially the same situation only an estimation with a term ε^n on the right hand side of (2.23) is obtained.

The estimation (2.23) "explains" that "overflow" is caused not directly by the growth of $\|A_k\|$ but by the extreme sensitivity of the arithmetical expressions involved in (2.12)–(2.13) with respect to this growth.

The algorithm of external stabilization

The problem is caused by the inevitable circumstance that – due to the necessary diminishment of $\lambda_1(A_k)$ as k grows – the matrices A_k may become increasingly badly conditioned. We see thus that the condition B requires (when we try to minimize the value of b_n) the maximum that can be required when we dot not impose any condition of strong convexity on f (since then $X^*(f)$ may be, say, a line segment). Formally the appearance of $\langle A_k\, g_k, g_k\rangle$ in the denominators of (2.12)–(2.13) causes the problem.

Here we have to assume that f has a known Lipschitz constant – which, without loss of generality can again be denoted by L – over the ball of radius b_nR around the origin, especially that

$$\|g(x)\| \leq L, \text{ for } x \text{ in } G_{b_nR}. \tag{2.24}$$

We suppose that the value ε of the required final accuracy for the computation of f^* is given (i. e. we have to guarantee that $\varepsilon(N, f) \leq \varepsilon$, for some N) and try to determine the accuracy of computations minimally needed for this purpose.

We shall need several lemmas, which will lead to the proof of Theorem 2.

Lemma 3. *Suppose that A and A' are symmetric positive definite $(n \times n)$ matrices and x and x' are vectors in $\mathbf{R}^n$ such that*

$$|A_{i,j} - A'_{i,j}| \leq \lambda_1(A)\varphi(n), \ (i, j = 1, \ldots, n), \tag{2.25}$$

$$\|x - x'\| \leq \sqrt{\lambda_1(A)}\Psi(n), \tag{2.26}$$

then

$$x' + \frac{1+\Psi(n)}{1-\varphi(n)n} E(0, A') \supseteq E(x, A) \supseteq x + \frac{1}{1+\varphi(n)n} E(0, A'), \tag{2.27}$$

whenever

$$s(n) = \varphi(n)n < 1. \tag{2.28}$$

Proof. First we note that for the Hausdorff distance

$$d(E(x, A), E(x', A')) \leq \|x - x'\| + \|\sqrt{A} - \sqrt{A'}\|$$

holds. Now (2.25) implies that

$$\|\sqrt{A} - \sqrt{A}\| \leq \sqrt{\lambda_1(A)}\ n\varphi(n). \tag{2.29}$$

Indeed – for arbitrary symmetric positive definite matrices

$$4\|D^2 - B^2\|_F^2 = \|(D+B)(D-B) + (D-B)(D+B)\|_F^2 \geq 4\lambda_1^2(D)\|D - B\|_F^2,$$

where $\|C\|_F$ denotes the Frobenius norm of a matrix C

$$\|C\|_F^2 = \sum_{k,j=n}^{n} C_{ij}^2,$$

and we have used the inequality $\|z_1 + z_2\|^2 \geq 4\langle z_1, z_2\rangle$ and used a coordinate system (when computing the F norm) where $D+B$ is diagonal. Since (2.25) implies that

$$\|A - A'\|_F^2 \leq n^2\lambda_1^2(A)\varphi^2(n) \text{ and } \|C\|^2 \leq \|C\|_F^2,$$

for an arbitrary symmetric matrix C, we obtain (2.29), if we set $D = \sqrt{A}$, $B = \sqrt{A'}$.

Now the inclusions (2.27) follow easily from (2.29) and from

$$E(A, x) \supseteq \sqrt{\lambda_1(A)}G_1 + x. \quad \square$$

We shall define the updates of x_k^* and A_k^* by (see (2.35))

$$x_{k+1}^* := x_{k+1}', \ (x_0 = 0), \ s(n) = \frac{1}{32n^2(n+1)}, \tag{2.30}$$

$$A_{k+1}^* := \left(\frac{1+s(n)}{1-s(n)}\right)^2 A_{k+1}', \ A_0 = R^2 I, \tag{2.31}$$

where x_{k+1}' resp. A_{k+1}' denote the result of the error contaminated computations (2.12), (2.13) started with $x_k = x_k^*$, $A_k = A_k^*$, (for an other, possible choice of A_{k+1}^* see (2.50). In these formulas we shall set $D_k = 0$ (because the error analysis for the general case $D_k \geq 0$ is somewhat more complicated and apparently we have to pay for the acceleration – say when $0 \leq D_k < 1 - \delta$, for a fixed $\delta > 0$ – by a corresponding decrease of stability depending on δ).

Now we shall choose the values of $\psi(n)$ and $\varphi(n)$ so that

$$E(x_k^*, A_k^*) \supseteq X^*(f) \cap G_R$$

for all k. This holds if

$$E(x_{k+1}^*, A_{k+1}^*) \supseteq E(x_{k+1}, A_{k+1}), \tag{2.32}$$

where x_{k+1}, A_{k+1} denote the "exact" updates corresponding to the starting values $x_k = x_k^*$, $A_k = A_k^*$ according to (2.12)–(2.13). Lemma 3 and the definitions (2.30), (2.31) assure that (2.32) holds, for all k, if (2.25) and (2.26) is satisfied for $A = A_{k+1}$, $A' = A'_{k+1}$, $x = x_{k+1}$, $x' = x'_{k+1}$. (Note that in (2.32) the notation does not mean that the sequence of pairs A_k, X_k are identical with the sequence generated by (2.11) and (2.12) in the error free case!) Moreover we shall guarantee that – for a suitable choice of $s(n) = s(n, \gamma)$ – and with $\gamma = \frac{1}{2}$, $\mu = 1{,}1$,

$$\text{vol } E(x_{k+1}^*, A_{k+1}^*) \leq e^{-\frac{\gamma}{(n+1)}} \text{ vol } E(x_k^*, A_k^*) \tag{2.33}$$

$$\|A_{k+1}^*\| \leq \frac{n^2}{n^2 - \mu} \|A_k^*\|. \tag{2.34}$$

In other words: the stabilized algorithm retains the validity of the two estimations which are essential for the estimations established in Section 3. 4, and which imply that (2.18) can be satisfied – even if only (2.33) and (2.34) hold – with a constant b_n depending on γ and μ, ($b_n \leq 500$ is claimed here only for $\gamma = 1/2$, $\mu = 1{,}1$. When not stated otherwise – as in the next remark – we shall always assume that $\gamma = 1/2$, $\mu = 1{,}1$ (of course if we change $N/2$ in (2.21) to $N\gamma$, then the value ε_1 should be defined correspondingly).

Lemma 4. *The requirements* (2.33) *and* (2.34) *will be satisfied if we choose* (ε_0 *so small that* (2.25), (2.26) *holds with*) – *for* $n \geq 4$ –

$$\Psi(n) = n\varphi(n) = s(n) = \frac{1}{32n^2(n+1)}. \tag{2.35}$$

Proof. This will follow from Lemma 3. From the well-known asymptotics

$$\log(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \ldots, \; x \text{ small},$$

we get for h_n, see (2.9) – using a simple majoration technique –

$$\lg h_n = \frac{n-1}{2} \lg \frac{n}{n-1} - \frac{n+1}{2} \lg \frac{n+1}{n} \leq -\frac{1}{2(n+1)} -$$

$$-\frac{1}{4(n+1)} + \frac{4n^3 - 6n^3(n-1) + 3(n+1)(n-1)^2}{24n^4(n-1)^2},$$

therefore (2.33) will be satisfied – in virtue of (2.27) and (2.31) – if for $s = s(n)$

$$\frac{\text{vol } E_{k+1}^*}{\text{vol } E_k^*} \leq \left(\frac{(1+s)^2}{1-s}\right)^n \leq \left(\frac{1+s}{1-s}\right)^{2n} \leq (1+2s)^{2n} \leq$$

$$\leq \exp(4sn) \leq \exp \frac{1}{8n(n+1)}$$

and

$$\frac{1}{8n(n+1)} \geq \frac{4n^3-6n^2(n-1)+3(n+1)(n-1)^2}{24n^4(n-1)^2}.$$

The condition (2.34) will be satisfied also, since

$$(2.36)\quad \left(\frac{1+s}{1-s}\right)^2 \frac{n^2}{n^2-1} \leq \frac{n^2}{n^2-1,1} \text{ holds if } \left(\frac{1+s}{1-s}\right)^2 \leq 1+0,1n^2$$

and the latter inequality is satisfied for the choice made in (2.35). □

Remark. It is easy to see that for any value of $0<\gamma<\frac{1}{2}$ the validity of (2.33) and (2.34) with some (smallest possible) $\mu = \mu(\gamma) > 1$ can be guarantee whenever

$$(2.37)\qquad s(n) = \varrho(\gamma)n^{-2}$$

for a suitable (largest possible) positive value $\varrho(\gamma)$. The value of b_n in (2.18) whose existence is proved in Section 4 will then depend on γ so that $b_n(\gamma_1) > b_n(\gamma_2)$, if $\gamma_1<\gamma_2$, see (4.23). Therefore the gain of a factor n^{-1} in (2.37) with respect to (2.35) leading to the same gain in (2.23) is largely upset by the corresponding growth of b_n and the diminishment of the convergence rate constant $\frac{1}{2}$ to γ. This is why we restricted ourselves to the case $\gamma = 1/2$, when the analysis of the proof of Lemma 4 shows that – for $n\to\infty$ – we can choose $\mu_n \to 1$, and then $\lim_{n\to\infty} b_n < 130$ can be ascertained.

The proof of Theorem 2. The basic, simple observation is that – see our analysis in (2.20) of the consequences of the stopping rule (2.10) – at moments k, when $\lambda_1(A_k)$ or $|g_k|^2$ (thus the denominators in the second term of the right hand side of (2.13) or (2.12)) become small, the value $f(x_k)$ must be already near to f^*. In fact in our stabilized algorithm it may happen that the calculations will be stopped not by (2.10) but by the equality $k = N(\varepsilon, n, R, L)$ (which is adopted thus as an additional stopping rule), nevertheless the smallest value $k = k'$ for which $f(x_k) \leq f^* + \varepsilon$ will be much smaller than $N(\varepsilon, n, R, L)$ and then the values of A_j' and x_j' for $j > k'$ may (are allowed to) behave very irregularly. As this – together with Lemma 2 and 3 – indicates, it will be convenient to bound the errors of the computation of A_{k+1} and x_{k+1} in terms of the smallest eigenvalue of A_{k+1}. In order to do this we need the following simple estimation.

Let $\varepsilon_2, \varepsilon_3, \beta > 0$ be real numbers, a, z vectors in a space $\mathbf{R}^m$, $\|z\| \leq 1$, then

$$(2.38)\quad \left\| \frac{a+\varepsilon 2z}{\beta-\varepsilon_3} - \frac{a}{\beta} \right\| \leq \frac{4\max\{\|a\|, \beta\}}{\beta^2} \max\{\varepsilon_2, \varepsilon_3\}, \text{ if } \varepsilon_3 \leq \frac{\beta}{2}.$$

We shall apply this lemma for two cases, first for (a, z scalars)

$$(2.39)\qquad a = (A_k g_k)_i (A_k g_k)_j, \ (i, j = 1, \ldots, n), \ \beta = \langle A_k g_k, g_k \rangle,$$

when we analyse (2.13), and for the cases (a, z vectors in $\mathbf{R}^n$)

$$a = A_k g_k,\ \beta = \langle A_k g_k, g_k\rangle^{1/2} \tag{2.40}$$

when we analyse (2.12). By the meaning of these variables there is no loss of generality in assuming that

$$\max\{\|a\|, \beta\} = \|a\|.$$

In the case (2.39) the values of ε_2 and ε_3 can be estimated by the following inequalities (under the assumption that the values of $(A_k)_{i,j}$ and $g_k \in \mathbf{R}^n$ are known within accuracy ε_0 and $\|g_k(x)\| \le L$ for $x \in G_{b_n R}$) – provided that condition B holds – using repeatedly the Cauchy-Schwartz inequality for vectors in $\mathbf{R}^n$

$$\varepsilon_2^* \le 2b_n^2 R^2 L(nL + n + R^2 b_n^2 \sqrt{n}) + 1)\varepsilon_0 + 0(\varepsilon_0^2) \tag{2.41}$$

$$\varepsilon_3^* \le (L^2 n + 2R^2 b_n^2 L\sqrt{n} + 1)\varepsilon_0 + 0(\varepsilon_0^2).$$

In the case of (2.40) these values can be estimated by

$$\varepsilon_2' \le (nL + n^{3/2} + R^2 b_n^2 \sqrt{n})\varepsilon_0 + 0(\varepsilon_0^2) \tag{2.42}$$

$$\varepsilon_3' \le \frac{2}{\sqrt{\varepsilon_1}}((L^2 n + R^2 b_n^2 L\sqrt{n} + 1)\varepsilon_0 + 0(\varepsilon_0^2).$$

(One can prove that here everywhere $O(\varepsilon_0^2) \le (n^2 L + b_n^2 R^2 L + n)^2\, \varepsilon_0^2$, if $\varepsilon_0 < 1$.) In the last inequality we already used the validity of $\beta \ge \sqrt{\varepsilon_1}$, (by the stopping rule (2.10)), and the inequality (where $u = \beta^2$)

$$\varepsilon_3' = \sqrt{u} - \sqrt{v} \le \frac{1}{\sqrt{v}}(u - v),\ \text{for } 0 < v \le u,$$

together with the assumption needed in (2.38) (and to be guaranteed below by the proper choice of ε_0)

$$\varepsilon_3 \le \frac{\sqrt{\varepsilon_1}}{2},\ \left(i.e.\ \sqrt{v} = \sqrt{u} - \varepsilon_3 \ge \frac{2}{\sqrt{\varepsilon_1}}\right). \tag{2.43}$$

Now in order to that the assumptions of Lemma 3 be fulfilled (for the corresponding data at step k) – according to (2.38) – we have to impose the following two conditions, where we use again the condition (2.10) and the inequality $\langle A_k, g_k, g_k\rangle \ge \lambda_1(A_k)\|g_k\|^2$

$$\frac{18\, n^2 R^4 b_n^4 L^2 \max\{\varepsilon_2^*, \varepsilon_3^*\}}{(n^2-1)(n+1)\lambda_1^2(A_k)\|g_k\|^4} \le \frac{n}{n+1}\lambda_1(A_k)\varphi(n), \tag{2.44}$$

$$\frac{4R^2 b_n^2 L \max\{\varepsilon_2', \varepsilon_3'\}}{(n+1)\lambda_1(A_k)|g_k|^2} \le \left[\frac{n}{n+1}\lambda_1(A_k)\right]^{1/2}\Psi(n) \tag{2.45}$$

Here we have used the equality $\lambda_1(A_{k+1}) \geq \frac{n}{n+1}\lambda_1(A_k)$ which follows from (2.13) (if $D_k = 0$ as assumed).

The final, important observation is now that – since (because of (2.1.), we shall obtain the same alternative as in (2.20) – we can assume that in (2.44), resp. (2.45)

$$\lambda_1^3(A_k)\|g_k\|^4 \geq \varepsilon_1^{5/2}\frac{R}{L}, \tag{2.46}$$

$$\lambda_1^{3/2}\|g_k\|^2 \geq \varepsilon_1^{5/4}\left(\frac{R}{L}\right)^{1/2}. \tag{2.47}$$

Based on (2.41) and (2.42). the inequalities (2.44)–(2.47) lead to the desired connection between ε_0 and $(\varepsilon, n, R, L, b_n)$. It turns out that the restrictions for ε_0, coming from the error analysis of the updates of A_k are more severe – for fixed the variables – (n, b_n, L, R) than those arising from the error analysis of the updates of x_k, if ε is small enough. This follows by noting that because of (2.41)–(2.47) it is enough to require – say for $R = L = 1$ – that

$$\text{const}\,\frac{b^8(n+\text{const})}{n}\,\varepsilon_0 \leq \varepsilon_1^{5/2}\varphi(n) \tag{2.48}$$

and

$$\text{const}\,\frac{b^4(n^{3/2}+\text{const})}{n}\,\varepsilon_0\varepsilon_1^{-1/2} \leq \varepsilon_1^{5/4}\Psi(n).$$

Here the second inequality is a consequence of the first one if $\varepsilon_1^{7/4}\Psi(n) \geq \geq \varepsilon_1^{5/2}\varphi(n)\sqrt{n}$ const, (here const = const (L, R, b_n).

Now since $\varepsilon_1^{7/4-5/2} = \varepsilon_1^{3/4} = \varepsilon^3$ const, $\psi(n) = n\varphi(n)$, the last inequality will be satisfied if $\varepsilon \leq$ const. It is easy to check that (2.23) assures that the terms $O(\varepsilon_0^2)$ in (2.41) and (2.42) are not greater than constant (not depending on (n, ε, L, R)) multiples of the corresponding main parts if $\varepsilon <$ const (L, R, n); the condition $\varepsilon_3 \leq \beta/2$ in (2.38) (see also (2.43)) will be satisfied also for all cases of interest).

Finally (2.23) in obtained from (2.44), (2.46) and the definitions of ε_1 and $\varphi(n)$ in (2.22) resp. (2.35).

Summarizing, the result obtained means that if we apply the algorithm described by (2.9)–(2.13), (2.30), (2.31) (with $D_k = 0$) and the prescription (2.22), then – for any prefixed $\varepsilon <$ const(R, L) and the rounding assumptions made above –

$$\varepsilon(N, f) \leq \varepsilon, \text{ holds for } N \geq N(\varepsilon, n, LR) := \left[n2(n+1)\log\frac{2LR}{\varepsilon}\right] + 1. \tag{2.49}$$

Notice that – since there is no simple way to check the validity of the conditions (2.20), (2.46) and (2.47): it is possible that these conditions are violated already for some $k < N(\varepsilon)$, the point is that then necessarily $f(x_k) - f^* \leq \varepsilon$. and (2.32) need not be satisfied any more.

Since $\lambda_1(A_k) \geq \frac{R}{L}\sqrt{\varepsilon_1}$ can be assumed, the previous proofs show that we can change the formula (2.31) to

$$\bar{A}_{k+1} := A_{k+1} + \frac{R}{L}\sqrt{\varepsilon_1}\left(\frac{1+s(n)}{1-s(n)}\right)^2 I, \tag{2.50}$$

where I is the $(n \times n)$ indentity matrix.

We have found convenient to write the update formulas (2.12)–(2.13) in terms of the matrices A_k. Instead of this, one could directly update the matrices $A_1^{1/2} Q_k$ for a specially chosen orthogonal matrix Q_k, and e. g. by Cholesky factorization of A_k, return to them after the stabilizing steps. Then one can prove that *instead of* ε^{10} *we can set* ε^7. The corresponding formulae and estimations are given in [12]. Speaking about the standard steps of the (ellipsoid) algorithm, we shall – for brevity – usually refer to only (2.12)–(2.13), yet everything applies as well for the modifications (2.31) or (2.50) (because (2.33) and (2.34) hold).

3. The (internal) stabilization method

In this section we present the description of the stabilization method which consists of several numerical subalgorithms. The tuning of their parameters (degrees of approximation) – necessary to obtain an overall error estimation for the method proposed – will be specified in Section 4.

First we choose the constant b_n (whose value will be specified later). The modified algorithm now will consist of megasteps $s \to s+1$, during each of such steps $L_{s+1} - L_s$ evaluations of $f(x)$ and $g(x) \in \partial f(x)$ will take place and the algorithm proceeds as described in (2.10)–(2.13), till $k = L_{s+1}$, when

$$\|A_k\|_F \geq b_n^2 R^2 \frac{n^2 - \mu}{n^2} \tag{3.1}$$

is satisfied for the first time after $k = L_s$.

In fact it will not be necessary to compute the Frobenius norms at each step k because one can give a lower bound for $L_{s+1} - L_s$. More precisely: we shall prescribe the introduction (i. e. application) of a "stabilization step" automatically after each $[(2n^2+n)3{,}105]+1$ steps (2.11)–(2.13), see below (3.9), (4.21), so that then $\|A_p\| \leq b_n^2 R^2$ holds for all p, and here for all n, where $b_n \leq c_0$ is an universal constant, see (3.10). In the stabilization steps when $k = L_s$, $s = 0, 1, \ldots$, instead of defining E_{k+1} by the formulas (2.11)–(2.13) we shall construct E_{k+1} to be an ellipsoid of possibly small volume and diameter which contains the intersection $E_k \cap G_R$. Thus

$$E_{L_{s+1}+1} \supseteq G_R \cap E_{L_{s+1}} \tag{3.2}$$

should hold for all values of s, $L_0 = 0$. (Notice that to obtain E_{k+1} from E_k we shall thus need no function evaluations when $k = L_s$, $s = 1, 2, \ldots$).

By this construction it remains true that all sets E_k, $k \geq 0$, will be sets of localization for $X^*(f) \cap G_R$. If E_k has the same centre as G_R i. e. $x_k = 0$, then we shall choose E_{k+1}, $(k = L_{s+1})$ to be an approximation of

$$E^*_{k+1} := E(x^*_{k+1}, A^*_{k+1}), \; x^*_{k+1} := 0, \tag{3.3}$$

$$A^*_{k+1} := 2\big(A_k P_1(A_k - R^2 I) + R^2 P_2(A_k - R^2 I)\big) = 2B(A_k). \tag{3.4}$$

Here $P_1(S)$ resp. $P_2(S)$ are the orthogonal projectors into the subspaces generated by the "nonpositive" resp. "positive" eigenvectors of the symmetric matrix S. (We say that $z \neq 0$ is a "positive" eigenvector of S iff $Sz = \lambda z$, has a solution $\lambda > 0$.)

We shall numerate the eigenvalues of the positive symmetrical matrices A according to

$$\lambda_1(A) \leq \lambda_2(A) \leq \ldots \leq \lambda_n(A) \tag{3.5}$$

and assume that the corresponding eigenvectors are orthonormal.

Lemma 5. *For arbitrary* $0 < A_k = A^*_k \in \mathbf{R}^{n \times n}$ *the ellipsoid defined by* (3.4) *contains the set* $G_R \cap E(0, A_k)$ *and has a diameter (volume) not greater than* $\sqrt{2}R$ *(resp.* $\sqrt{2^n}$ vol $E(0, A_k)$*).*

Proof. Obviously we can drop the index k. Now let x be in $G_R \cap E(0, A)$ and let

$$I_1 = \{j \mid \lambda_j(A) \leq R^2, \;\; I_2 = \{j \mid \lambda_j(A) > R^2\}.$$

Then using coordinates with respect to the eigenvectors of A:

$$\frac{1}{2} \sum_{j \in I_1} \lambda_j^{-1}(A) x_j^2 + \frac{1}{2R^2} \sum_{j \in I_2} x_j^2 \leq \frac{1}{2} + \frac{1}{2} = 1,$$

which finishes the proof. □

Remark. If we replace G_R by an arbitrary (nondegenerated) ellipsoid $E(0, C)$, then the analogous problem can be solved by transforming $E(0, C)$ into a ball.

In the case when $x_k \neq 0$, we construct x_{k+1} and A_{k+1} to be (approximations of)

$$x^*_{k+1} := x_k - z_{k+1}, \; z_{k+1} := P_1(A_k - R^2 I) x_k, \tag{3.6}$$

$$A^*_{k+1} := 18B(A_k), \text{ see (3.4)} \tag{3.7}$$

(The factor 18 could be probably reduced to 2 by a proper choice of x^*_{k+1}).

Lemma 6. *The ellipsoid* E^*_{k+1}, defined by (3.6), (3.7) *satisfies the relation* (3.2) *for* $k = L_{s+1}$.

Proof. The statement of Lemma 6 follows from Lemma 5 using the following simple observation. If a point, ω belongs to an ellipsoid $E(u, 4V)$, then $E(u, V)$ is contained in $E(\omega, 9V)$. The previous conditions hold for $\omega = x^*_{k+1}$ and $(u, V) = (0, R^2 I)$ resp. $(u, V) = (x_k, A_k)$. □

What remains to do is choosing the number b_n and the numerical method to approximate the objects $P_i(A_k - R^2 I)$, $i = 1, 2$ (thus A^*_{k+1} and x^*_{k+1}) appropriately.

It turns out that a proper choice of these approximations assures that one van achieve – for all s and $n \geq 6$ – that

$$\text{vol}\,(E_{L_{s+1}}) \leq e^{-n}\,\text{vol}\,E_{L_s} \tag{3.8}$$

with

$$L_{s+1} - L_s \leq [(2n^2 + n)\,3{,}105] + 1 \tag{3.9}$$

so that

$$b_n \leq 500. \tag{3.10}$$

Here [ ] denotes the entire part and the assumption $n \geq 6$ is made only for simplicity, for $n \leq 6$ the same kind of inequalities hold with different constants.

Remark. We could replace the function $B(A)$ by a *more easily computable* one (i. e. compute an outer ellipsoid containing $(E(0, A) \cap G_R)$ by the following formula
$B'(A) = (A^{-1} + R^{-2} I)^{-1}$, then $E(0, B'(A)) \subseteq E(0, B(A)) \subseteq \sqrt{2} E(0, B'(A))$. This, however leads to a larger increase of volume, see also [14].

Notice that in the original algorithm one needs – by (2.9) and (2.33) – approximately $2n^2 + n$ steps (2.12)–(2.13) in order to reduce the volumes by the factor e^{-n} (which implies a reduction of the uncertainty in f^* by the factor e^{-1}). Thus (3.9) shows that in the stabilized version we need to perform only about three times more steps (2.12)–(2.13) than in the original version. (The constants in (3.9) and (3.10) are not the best possible ones).

The error estimation (2.21) remains true if we replace N by $N \cdot 3{,}105^{-1}$ and set $b_n = 500$ in (2.22). The number of arithmetical operations needed in the stabilization steps – during the whole algorithm with N function evaluations and steps (2.12)–(2.13) – will not exceed the sum of $\frac{N}{6n^2}\left(\frac{n^3}{3} + 2n^2\right) \times$ $\times (n+1)$ – which arises by the preconditioning (4.3)–(4.7) – and of

$$\frac{n^2 N c_2}{2n+1} \max\left\{c_3 \log\left(n(1+c_5 R^2)\left(\min\left\{\left|\log\left|\frac{2n^2\sqrt{n} - R^2}{2n^2\sqrt{n} - R^2}\right|\right|, \left|\log\left|\frac{R^2 500^2 + 1}{R^2 500^2 + 1}\right|\right|\right\}^{-1}\right)\right) \times$$

$$\times c_4 \log\left(\frac{N c_1}{n(2n+1)}\left(\log(1+c_5 R^2)\left(\min\left\{\left|\log\left|\frac{1 - 500^2 R^2}{1 + 500^2 R^2}\right|\right|, \left|\log\left|\frac{3/4\, R^2 - 1}{3/4\, R^2 + 1}\right|\right|\right\}^{-1}\right)\right)\right), \tag{3.11}$$

where $c_1, c_2, \ldots$ are positive constants not depending on R, n, and L. The last number is "significantly less" than the remaining number of arithmeti-

cal operations, $O(Nn^2)$ connected with the steps (2.12)–(2.13) if $\frac{1}{n} \lg Nn^{-2}$ is "small". Note that $R \geq 1$ can be assumed without loss of generality – and then the singularity of the expression in (3.11) at $R = 0$ disappears, yet enlarging R the error estimation (2.21) becomes worse.

The assumption that $\frac{1}{n} \lg \frac{N}{n^2}$ is small will be satisfied for large n because $N = 7kn^2$ means that – after N function evaluations – the uncertainty in the values of f^* is already diminished (at least) by the factor e^{-k}.

Unfortunately we could not avoid the dependence on N of the expression in the maximum bracket of (3.11). This is so because the values of x^*_{k+1} and A^*_{k+1} in (3.6), (3.7) should be computed (in our method) with an increasing accuracy as s grows in order to ascertain (3.2) when $\lambda_1(A_k) \to 0$ as $k \to \infty$). Our method of doing this consists in providing such approximations for x^*_{k+1} and $A^{*1/2}_{k+1} G_1$ whose errors can be included into constant multiples of $B(A_k)^{1/2} G_1$.

Remark. It is interesting to note that a similar idea as the one used here allows to construct other "easily implementable" versions of the method of centers of gravity, see e. g. [8], [11]. In order to explain this let us suppose that the initial set of localization – for the whole set $X^*(f)$ – is a simplex P_0 in $\mathbf{R}^n$.

Then the exact set of localization for the possible minimumpoints after k function and subgradient evaluations will be a polyhedron P_k with at most $k+n+1$ faces. If k is small (with respect to n) one can easily compute the center of gravity of P_k e. g. by simply updated simplicial decomposition of P_k, where the operation of updating requires only the easily standardized simplicial decomposition of the intersection of a halfspace (corresponding to $(g(P_k), f(P_k))$ with simplexes (elements of the previous decomposition) into a small (minimal) number of simplexes. The decrease of volumes will then be independent of n, it is known that

$$\operatorname{vol} P_{k+1} \leq \left(1 - \frac{1}{e}\right) \operatorname{vol} P_k.$$

Now in order to keep the sets P_k to have a finite complexity we can set a fixed upper bound $k_n \geq k$ and include the set P_{k_n} into a simplex P^1_0 of small volume. There is a simple, fast algorithm for computing $P^1_0 = P^1_0(P_{k_n})$ such that

$$\operatorname{vol} P^1_0 \leq n^n \operatorname{vol} P_0.$$

Thus, in order to obtain a geometrically convergent method it is enough to set $k_n = O(n \log n)$. Now starting from P^1_0 in the same way as from P_0 the k_n step long iteration can be repeated. For the case $n = 2$ allowing $k_n = 4$, (i. e. poligons with seven vertices) we already get a method with a faster convergence than the elllipsoid method. It seems that – for small values of n – these "simplex-polyhedral" methods are superior to the ellipsoid method in

some respects. Here also we have to prescribe analogons of the above stabilization steps in order to include the intersection $P_0^{L_s} \cap P_0$ into a simplex $P_0^{L_s+1}$.

The main difficulty here is that the bound $k_n = O(n \log n)$ allows P_{k_n} to have exponentially (in n) many vertices. Also from the point of view of (the analysis of) the external stability problem these methods seem to be less favourable (more complicated) than the ellipsoid method for large values of n. One can prove that for any N-step algorithm (i. e. one using N (exact) evaluations of $f(x)$ and $g(x)$) it is true that

$$\sup \{\varepsilon(N, f) | f \in F(G_R, L)\} \geq p_0 LR\, 8^{-\frac{N}{n}}, \tag{3.12}$$

holds with an universal constant p_0 [11]. While the above described method of the center's of "gravity" is "optimal" – since by (3.12) and a simple analogue of Lemma 2 it converges like $\alpha^{N/n}$, for $\alpha \leq \left(1 - \frac{1}{e}\right)$ – nevertheless (in the case of identical memory, arithmetical complexity and stability requirements) the stabilized ellipsoid method seems to be competitive with any of the known methods, see howerer [14] for a promising, new tool.

4. Realization of the numerical subalgorithms

For the approximation of $P_i(S)$, $s = 1, 2$ the following method is used. Because of $P_1 + P_2 = I$ it will be enough to compute the difference $Q = P_2 - P_1$. Consider the iteration

$$Q_0 = S,\ Q_{i+1} := \frac{1}{2}(Q_i + Q_i^{-1}),\ i = 0, 1, 2, \ldots, r. \tag{4.1}$$

Let $\lambda_i(S) \neq 0$, $i = 1, \ldots, n$, it is easy to see that $Q_k \to Q$ quadratically for $k \to \infty$. This method of dividing the spectrum of S is proposed in [9], the author thanks T. Fiala for providing this reference. An other method (with essentially the same properties) can be obtained from the observation: $(P_2(S) - P_1(S))\sqrt{S^2} = S$; one can apply the Newton method for the computation of the positive square root $\sqrt{S^2}$.

In the iteration (4.1) we can use an exact i. e. noniterative method for the computation of the inverses Q_i^{-1}, (e. g. Gaussian elimination). In view of the statement in (3.11) this does not lead to a relatively large augmentation of the number of arithmetical operations needed for the algorithm as a whole. Since the limit Q is a well conditioned matrix, the instabilities which might arise in the iteration (4.1) are essentially those of the map $S \to (\sqrt{S^2})^{-1} S$. Because of $\|S\| \leq b_n^2 R^2 \leq 500^2 R^2$, the main problem will be the analysis of the effects of a perturbation $S \to S'$ (preconditioning) by which we achieve that the eigenvalues of S' are larger in absolute value than $\lambda_0(n)$ an appropriately chosen positive function of n. This however implies that here, i.e. for (4.1), the accuracy ε_0 needed for the computations depend more heavily on n, see (4.27).

Lemma 7. *For the iteration* (4.1) *the estimation*

$$(4.2)\quad |\lambda_j(Q_k)-\operatorname{sign}\lambda_j(S)|\leq\left(\max\left\{\left|\frac{1-\lambda_0(S)}{1+\lambda_0(S)}\right|,\frac{\|S\|-1}{\|S\|+1}\right\}\right)^{2^k}(1+\|S\|)$$

holds for all $k>0$, $j=1,\ldots,n$, *where* $\lambda_0(S)\neq 0$, *denotes the minimum of the absolute values of the eigenvalues of* S.

Proof. Using the spectral decomposition of S one can reduce the study of the convergence of the sequence (4.1) to the one-dimensional case. It is easy to check, that

$$\frac{Q_{k+1}+1}{Q_{k+1}-1}=\left(\frac{Q_k+1}{Q_k-1}\right)^2=\left(\frac{Q_0+1}{Q_0-1}\right)^{2^{k+1}}\quad(\text{if } Q_{k+1}\neq 1).$$

In both cases $Q\leq\lambda_0$, resp. $Q_0>\lambda_0$ one obtains monotone and quadratic convergence, which is the faster the more Q_0 is away from the "singular points" $Q_0=0$, $Q_0=\infty$. From $\|S\|=\max\{|\lambda_j(S)\|, j=1,\ldots,n\}$, (4.2) follows since $\|S\|\leq\max\{\|A\|, R^2\}$, and $\|A\|\leq R^2b_n^2$, for all possible (i. e. occuring) values of A and S, we could replace $\|S\|$ by $R^2b_n^2$ in (4.2). In what follows we shall concentrate on the dependence of (the right hand side of) the estimation (4.2) on $\lambda_0(S)$. Since $b_n\leq 500$ can be ascertained – a condition which is of course independent on the number of iterations (4.2), $k=r$, necessary to yield the accuracies for $|\lambda_j(Q_k)-\operatorname{sign}\lambda_j(S)|$ required below (see (4.26)) – we shall always assume that r is so large that the value of

$$(1+R^2b_n^2)\left(\frac{R^2b_n^2-1}{R^2b_n^2+1}\right)^{2^r}\leq\left(\frac{R^2500^2-1}{R^2500^2+1}\right)^{2^r}(1+R^2500)$$

is smaller than these accuracies. The point is that the speed of convergence in (4.2) – for the large eigenvalues $\lambda_j(S)$ – will be independent of n. Our aim is to show that r can be chosen to be not greater than the value of the maximum bracket in (3.11).

Thus the only problem is to guarantee for $S=A-R^2I$ not to have zero (small) eigenvalues. Therefore prior to the application of the iteration (4.1) we have to "precondition" $S=A-R^2I$. For this we consider the matrices

$$(4.3)\qquad S_j=A-R^2I+\frac{jR^2}{n^2}I,\quad j=0,1,\ldots,n.$$

For at least one value of j we shall have

$$(4.4)\qquad \lambda_0(S_j)\geq\frac{R^2}{2n^2}.$$

In order to find such a value j constructively we note that (4.4) is equivalent to the inequality

$$(4.5)\qquad \|S_j^{-1}\|\leq\frac{2n^2}{R^2}.$$

Instead of computing $\|T_j\|$ for $T_j = S_j^{-1}, j = 0, 1, \ldots, n$ we shall compute the Frobenius norms $\|T_j\|_F, j = 0, \ldots, n$.

It is well-known that for $U = U^* \in \mathbf{R}^{n\times n}$

$$\|U\| \leq \|U\|_F \leq \sqrt{n}\, \|U\|. \tag{4.6}$$

Therefore from (4.4) follows the existence of a value j such that

$$\|S_j^{-1}\|_F \leq \frac{2n^2\sqrt{n}}{R^2},$$

and if we have found such a value j, then

$$\lambda_0(S_j) \geq \frac{R^2}{2n^2\sqrt{n}}. \tag{4.7}$$

Thus with $Q_0 = S_j$ we shall have after r iterations (4.1) (assuming that $\left|1 - \frac{R}{R^2 + 2n^2\sqrt{n}}\right| \geq \left|\frac{R^2 500^2 - 1}{R^2 500^2 + 1}\right|$ holds)

$$|\lambda_m(Q_r) - \operatorname{sign} \lambda_m(S_j)| \leq \left|1 - \frac{2R^2}{R^2 + 2n^2\sqrt{n}}\right|^{2^r} (1 + \|S_j\|) = \delta_{n,j}^r. \tag{4.8}$$

According to (3.4) let us define, see (4.3), (4.7),

$$P_2' = P_2^r = \frac{1}{2}(I + Q_r(S_j)),\ P_1' = I - P_2',\ S_j = S_j(A),\ A = A_k \tag{4.9}$$

$$B'(A) = AP_1' + R^2 P_2'.$$

By the choice (4.3) the matrices A and S_j (thus Q_r) have the same eigenvectors, therefore (see (3.4))

$$\left(1 - \frac{1}{n}\right)(AP_1(S_j) + R^2 P_2(S_j)) \leq B(A) \leq \left(1 - \frac{1}{n}\right)^{-1}(AP_1(S_j) + R^2 P_2(S_j)). \tag{4.10}$$

In order to prove an estimation

$$v_{n,j}^r B'(A) \leq AP_1(S_j) + R^2 P_2(S_j) \leq w_{n,j}^r B'(A) \tag{4.11}$$

we start from (4.8) and set, see (4.7),

$$R_j^2 := \left(1 - \frac{j}{n^2}\right) R^2, \tag{4.12}$$

$$w_{n,j}^r = 1 + \delta_{n,j}^r,\ v_{n,j}^r = (1 + \delta_{n,j}^r)^{-1}. \tag{4.13}$$

Now we shall choose $r = r(n, A_k, j)$ to be so large that

$$\delta_{n,j}^r \leq \frac{1}{n}. \tag{4.14}$$

Then (4.10) and (4.11) imply that

$$B'(A)\left(1+\frac{2}{n-1}\right)^{-1} \subseteq B(A) \subseteq B'(A)\left(1+\frac{2}{n-1}\right). \tag{4.15}$$

According to (3.6), (3.7) we define (see (3.2) and (4.9))

$$x_{k+1} := P_1^r(S_j)x_k,\ S_j = S_j(A_k). \tag{4.16}$$

Before defining A_{k+1} let us prove that – see (3.6) –

$$x_{k+1} - x_{k+1}^* \in 4B^{1/2}(A_k)G_1 \tag{4.17}$$

can be attained by a proper choice of r if $n \geq 6$. We shall use the coordinates x^i, $i = 1, \ldots, n$ of the vector $x = x_k$ with respect to the system of orthonormal eigenvectors of A_k, the origin being – as always – the centre of G_R. From the existence of a vector x^* in G_R such that $x + \sqrt{A_k}\,y = x^*$, for some y, $\|y\| \leq 1$ follows that $\|x\| \leq (b_n + 1)R$, see (3.1), and

$$|x^i| \leq \frac{3}{2}R \text{ if } \sqrt{\lambda_i(A_k)} \leq \frac{R}{2},$$

$$|x^i| \leq \frac{5}{2}R \text{ if } \left|\sqrt{\lambda_i(A_k)} - R\right| \leq \frac{R}{2}.$$

Now we observe that

$$P_1(S_0)x - P_1^r(S_j)x = (P_1(S_0) - P_1(S_j))x + (P_1(S_j) - P_1^r(S_j))x.$$

Here the first term of the right hand side belongs to

$$\frac{5}{2}\left(1-\frac{1}{n}\right)^{-1} B^{1/2}(A_k)G_1 \subseteq 3B^{1/2}(A_k)G_1, \text{ if } n \geq 6. \tag{4.18}$$

Decomposing the set of coordinate indices $[1, \ldots, n]$ into three parts as indicated above corresponding to $\lambda_i^{1/2} \leq \frac{R}{2}$, $\frac{R}{2} \leq \lambda_i^{1/2} \leq \frac{3}{2}R$ and $\lambda_i^{1/2} \geq \frac{3}{2}R$, (the main point is that we need a relatively large accuracy for the subspaces corresponding to the small eigenvalues of A_k) we see that the second term can be included into

$$\max\left\{\frac{3}{2}\,\frac{R\varrho_{n,j}^r}{\sqrt{\lambda_1(A_k)}},\ 5\delta_{n,j}^r,\ (1+b_n)\varrho_{n,j}^r\right\} B^{1/2}(A_k)G_1 \subseteq B^{1/2}(A_k)G_1, \tag{4.19}$$

$$\varrho_{n,j}^r := \max\left\{\left|\frac{1-500^2R^2}{1+500^2R^2}\right|^{2^r},\ \left|\frac{{}^3/_4R^2-1}{{}^3/_4R^2+1}\right|^{2^r}\right\}(1+\|S_j\|)(n \geq 4),$$

if we choose r to be so large that the maximum of the three numbers in (4.19) is not greater than 1. Then (4.18) and (4.19) imply (4.17).

According to (3.6), (3.7), (4.15) and (4.17) we define

$$A_{k+1} := \left(1+\frac{2}{n-1}\right)34B'(A_k), \tag{4.20}$$

(note that $34 = 4^2+18$), then (3.2) will be satisfied.

By adopting the same stopping rule (2.10) and definition of ε_1 in (2.22), and (2.33), we can now completely specify the (stabilized version of the ellipsoid) algorithm, by setting

$$L_{s+1} = L_s + [(\gamma^{-1}(n^2+n)\,3{,}105]+1, \quad L_0 = 0. \tag{4.21}$$

From (4.15) and (4.20) we obtain that

$$A_{k+1} \leq \left(1+\frac{2}{n-1}\right)^2 34\,B(A_k). \tag{4.22}$$

This implies – by $\operatorname{vol} B^{1/2}(A_k)G_1 \leq \operatorname{vol} E_k$ – that

$$\frac{\operatorname{vol} E_{k+1}}{\operatorname{vol} E_k} \leq \left(\left(1+\frac{2}{n-1}\right)^2 34\right)^{n/2} \leq e^{2{,}105n}, \text{ for } n \geq 6.$$

From (2.9) and (4.21) we obtain (3.8). Finally the validity of (3.10) follows from (4.22) and (2.34) by

$$\left(\frac{n^2}{n^2-2}\right)^{[\gamma^{-1}(n^2+n)\,3{,}105]+1}\left(1-\frac{2}{n-1}\right)^2 34 < 500^2 \quad (n \geq 6) \tag{4.23}$$

for $\mu = 1{,}1$, $\gamma = 1/2$.

Notice that by choosing $s(n) = qn^{-3}$, $q \to 0$, we can achieve that $\mu \to 1$, $\gamma = \frac{1}{2}$, then the limiting value of b_n in (4.23) for $n \to \infty$ is less than 130. If we choose $s(n) - qn^{-2}$ for q small enough, see (2.36)–(2.37), then for a suitable pair γ, μ we can get $b_n = b_n(\gamma, \mu)$ near to 50.

In order to prove the validity of the statement expressed in (3.11) note first that r is to be chosen to satisfy the conditions (4.14) and (4.19). According to (2.22) we set

$$\varepsilon_1 = \varepsilon^4(LR)^{-2}\,502^{-4}. \tag{4.24}$$

Then (4.14) implies (when $n \geq 6$) together with the stopping rule (2.10), (2.20) that (4.19) is satisfied if

$$\frac{3}{2}\,\frac{RL}{\varepsilon}\,\varrho^r_{n,j}\,502 \leq 1, \tag{4.25}$$

(notice that $\lambda_1(A_k) \geq \frac{R}{L}\sqrt{\varepsilon_1}$ can be assumed by (2.20) and $\varrho^r_{n,j}\,501 \leq 1$ will be a consequence of (4.25) because $RL \geq \varepsilon$ can be assumed by the meaning of the minimization problem, see (2.4)).

Now we shall choose r – at each step $s = 0,1, \ldots$ – to be the smallses natural number for which the inequalities (4.14) and (4.25) are satisfied (at consequences of (4.8) and the definition of $\varrho^r_{n,j}$ given after (4.19)). We have thus the inequalities

$$\frac{3}{2}\,502\,\max\left\{\left|\frac{1-500^2R^2}{1+500^2R^2}\right|^{2^r}, \left|\frac{\frac{3}{4}R^2-1}{\frac{3}{4}R^2+1}\right|^{2^r}\right\}\leq$$

$$\leq 2e^{-\frac{\gamma N}{(n+1)n}}(1+500^2R^2)^{-1}. \tag{4.26}$$

$$\max\left\{\left|1-\frac{2R^2}{R^2+2n^2\sqrt{n}}\right|^{2^r}\left|\frac{R^2500^2-1}{R^2500^2+1}\right|^{2^r}\right\}(1+R^2500^2)\leq\min\left(\frac{1}{n},\frac{1}{501}\right).$$

From these it is easy to obtain the estimation (3.11) by noting that in each of the steps (4.1) one has to perform $O(n^3)$ arithmetical operations, and during N function evaluations there are no more than $N(3(2n^2+n))^{-1}$ stabilization steps. Note that the procedure (4.3)–(4.7) may require $(n+1)\times$ $\times\left(\frac{n^3}{3}+n^2\right)$ operations at each such step.

Finally we note that if $R>1$, then

$$\log\left|\log\left|\frac{2n^2\sqrt{n}-R^2}{2n^2\sqrt{n}+R^2}\right|\right|^{-1}\leq c_8\log n+c_7\log R.$$

Thus we have proved that the proposed stabilization method retains all the positive features of the original method.

In order to analyse approximately the stability problem for the computation of $(P_2-P_1)(A_k-R^2I)$ we recall the formula (where $S = A_k-R_j^2I$) $(P_2(S)-P_1(S))=(\sqrt{S^2})^{-1}=S$. We expect that the behaviour of the iteration (4.1) with respect to errors in Q_0 and rounding errors is not essentially worse than that of this formula. Assume that the values of S_{ij}, $i,j = 1, \ldots, n$ are stored within accuracy ε_0, thus the elements of S^2 may contain errors of magnitude $\|S\|\varepsilon_0 n^{1/2}$, note that $\|S\|\leq b_n^2R^2$. Using the formula (2.29) and the well-known perturbation bound for the inverses ($A' = \sqrt{S'^2}$, $A = \sqrt{S^2}$)

$$\|A'^{-1}-A^{-1}\|\leq\frac{\|A'^{-1}\|^2\|A'-A\|}{1-\|A^{-1}\|\,\|A'-A\|},\quad\text{if}\quad\|A^{-1}\|\,\|A'-A\|<1,$$

we obtain in a similar way as in the proof of Theorem 2 that if

$$\varepsilon_0\leq\frac{1}{4}R^2b_n^{-4}n^{-10}, \tag{4.27}$$

the value of $P_1(S)-P_2(S)$ can be computed within accuracy $n^{-3/2}\,k_4$ (in the spectral norm) for a universal constant k_4. In view of (4.14) this will suffice for large enough values of n.

If $R\geq 1$, then the small eigenvalues of A_k are well separated from the small eigenvalues of $S_j(A_k)$, therefore no serious problem arises (note that $\|A_k\|\leq b_n^2R^2$), the accuracy (2.23) will be sufficient (by a scaling we can always assume that $R>1$).

5. Acceleration of the ellipsoid method

First we present an algorithm for the one-dimensional problem. This algorithm for computing f^* has (linear) convergence rate not greater than $9^{-1/3}$, it is thus faster than the ellipsoid (bisection) algorithm whose rate of convergence is 2^{-1} even if the accelerated versions (2.10–2.13) or (2.14) are used. Here (for the one-dimensional case) we shall not need the existence of a finite Lipschitz constant for f over $[-R, R] = [a, b]$. Note that the method of "golden sections" is not a concurrent (i. e. better) one if we assume that the values of $g(x)$ can be obtained with little additional cost (in addition to those of f). In fact we expect that the method proposed below – when properly generalized – yields a faster convergence than the method of golden sections even if only function values can be measured (but without errors: in fact, one should not be surprised to note that an accelerated method is less stable with respect to errors in the measurements; for related questions, esp. the stability of the method of golden sections, see [1]). It should be noted that the loss of stability accompanying the acceleration proposed below (as well as the one propozed in (2.12)–(2.14) for $D_k\neq 0$) is dangerous only at the "end" of the algorithm, i. e. in the "beginning" when we are far from the minimum (g_k and $\lambda_1(A_k)$ are not too small) these accelerations can be used freely.

Based on observations made in connection with this method we present a method to accelerate the convergence of the ellipsoid method for arbitrary values of the dimension n. In the new method we consider – at each step k – not only the (approximate) set of localization for the possible minimum-points but also (the approximation of) the „minimum point" x_k^* corresponding to that step.

The latter corresponds to that convex function f_k which provides – simultaneously for all x – the minimum of the possible values of $f(x)$ under the information gathered up to the step k: $f_k(x_k^*) = f_k^*$ (in order to make x_k^* unique, we define it to be that element of $X^*(f_k)$ which is closest to the centre of E_k) and

$$f_k(x) := \inf\{h(x)\,|\,h(x_j) = c_j,\ \operatorname{grad} h(x_j)\ni g_j,\ j = 1, \ldots, k,\ h \text{ convex}\} \tag{5.1}$$

where

$$c_j = f(x_j),\ g_j\in\operatorname{grad} f(x_j),\ j = 1, \ldots, k, \tag{5.2}$$

thus f_k depends on $c_1, \ldots, c_k, g_1, \ldots, g_k$.

In fact – in order to define x_{k+1} – we shall not have to compute the functions f_k, but only an approximation of their minimumpoints x_k^* (in addition to the approximate set of localization for the possible minimumpoints constructed as E_k above).

First we note that in the one-dimensional case it will be enough to remember, i. e. update – at each step k – the values of

$$(f(x_{k,1}), g(x_{k,1}), f(x_{k,2}), g(x_{k,2})), \text{ which fix a "standard" set } K(f, k),$$

$$f(x_{k,1}) := \min_{1 \le j \le k} f(x_j), \quad f(x_{k,2}) := \min \{f(x_j) \mid j \le k, \tag{5.3}$$

$$\operatorname{sign} g(x_j) = -\operatorname{sign} g(x_{k,1})\}.$$

Then the interval $(x_{k,1}, x_{k,2})$ will be free of points x_j, $j \le k$ and

$$\max_{r=1,2} \{f(x_{k,r}) + \langle g(x_{k,r}), x - x_{k,r}\rangle\} = f_k(x), \; x \in [x_{k,1}, x_{k,2}]. \tag{5.4}$$

Let now x_k', be that (uniquely defined) point in $[x_{k,1}, x_{k,2}]$ which satisfies

$$f_k(x_k') = f(x_{k,1}), \; x_k' \neq x_{k,1}. \tag{5.5}$$

Here we suppose that for $k = 2$ (the starting situation) $\operatorname{sign} g(x_1) = -\operatorname{sign} g(x_2) \neq 0$, where one of $g(x_i)$, $i = 1, 2$, may be infinity (thus we do not suppose the existence of a finite Lipschitz constant over $[x_1, x_2]$). We associate to these data, i.e. to $K(f, k)$, the triangle $T(f, k)$ formed by the points $(x_{k,1}, f(x_{k,1}))$, $(x_k', f(x_{k,1}))$ and (x_k^*, f_k^*). We define the one-dimensional algorithm by

$$x_{k+1} := \frac{1}{3}(x_k' + x_{k,1} + x_k^*), \quad k = 2, 3, \ldots . \tag{5.6}$$

In order to obtain an error estimation for this algorithm we introduce a "functional" φ defined over the "standard" (recurrent) situations, $K = K(f, k)$; therefore we shall write

$$\varphi(K(f, k)) = \varphi(f, k), \; x_{k+1} = x(K(f, k)).$$

Thus $K(f, 2)$ corresponds to the initial information: $f(x)$ and $g(x)$ are computed at a and b, providing a set of localization in $C(a, b)$).

$$\varphi(f, k) := (f(x_{k,1}) - f_k^*)\left(\frac{f(x_{k,2}) - f_k^*}{f(x_{k,1}) - f_k^*}\right)^{1/3}.$$

Theorem. *For the algorithm defined by* (5.6) *one has*

$$|f^* - f(x_{k,1})| \le 9^{-\frac{k-2}{3}} \varphi(f, 2), \; k = 2, 3, \ldots .$$

The proof of this theorem is reduced – by induction with respect to k – to the following Lemma.

Lemma 8. *The value of the functional φ is diminished — at each step k, i. e. for arbitrary values of $c_{k+1} = f(x_{k+1})$ and $g_{k+1} \in f'(x_{k+1})$ — at least by the factor $9^{-1/3}$.*

The proof of this lemma is elementary, for brevity we present here only the main observation: the functional φ is "invariant" under the "affine" transformations (scalings)

$$\varphi(f^{**}, k) = d\varphi(f, k), \text{ if } f^{**} = df - e,\ d, e \in R^1,\ d > 0.$$

$$\varphi(\tilde{f}, k) = \varphi(f, k), \text{ if } \tilde{f}(x) = f(\alpha(x - x_k^*) + x_k^*).$$

Moreover one can show that is is enough to prove Lemma 8 for the cases where $f(x_{k,1}) = 0, f(x_{k,2}) = 1$ and $f'(x_{k,2}) = -\infty$, (or $f'(x_{k,1}) = -\infty$, in fact, even the latter case can be easily reduced to the first one). This is so because among the "standard" sets $K = K(k)$, with fixed values of $\varphi(K)$ and $|x_k' - x_{k,1}|$ (we can set $x_{k,1} = 0$, $x_{k,2} = 1$) the largest possible value for $\varphi(K(k+1))$ will be realized (for some, worse values of $f(x_{k+1}) = c_{k+1}^* = c(K)$ and $g(x_{k+1}) = g_{k+1}^* = g(K)$) in the case when f' is infinite at $x_{k,2}$ (or $x_{k,1}$).

This can be proved using the following observations. The affine transformation of R^2 which keeps the lines $f = \text{constant}$ fixed and transforms the triangle T corresponding to K into a triangle T^* (having one vertical side and corresponding to a standard set K^*) takes the centre (of gravity) of T, (x_{k+1}, y_{k+1}) into the centre of T^*. Now one can consider the four cases — corresponding to the alternatives (where $x_{k+1} = x(K)$)

$$f(x_{k+1}) \lessgtr y_{k+1},\ g(x_{k+1}) \lessgtr 0,$$

separately, and show that in all cases of possible outcomes of $f(x_{k+1})$, $g(x_{k+1})$ (which define the new value $\varphi(K(k+1))$ there exist an outcome $(f^{**}(x_{k+1}'), g^{**}(x_{k+1}'))$ (not necessarily the one corresponding by the above affinity) such that for the new sets of localizations $K(k+1)$, $K^{**}(k+1)$

$$\varphi(K^{**}(k+1) \geq \varphi(K(k+1)).$$

In the worst cases specified above, Lemma 8 is obtained by straightforward computation (the minimal value of a cubic polynomial should be estimated only); the constant $9^{1/3}$ is not the optimal one.

Notice that in order to construct x_{k+1} we have to know only the value of x_k^* and the (exact) interval of localization $(x_{k,1}, x_k')$. Based on this observation we propose now the following (heuristic) algorithm for the multidimensional case, its implementation is not essentially more complicated than that of (2.11)–(2.13).

The standard situation which appears and is updated at each step k will be that in addition to the ellipsoid $E_k^1 = E(x_k^1, A_k^1)$ — providing a set of localization for $X^*(f) \cap G_R$ — we know also an other ellipsoid $E_k^2 = E(x_k^2, A_k^2)$ which contains all the points z in G_R, for which $f(z) \leq d_k$, where $d_k < f(x_{k,1})$ is chosen by the algorithm. Here $x_{k,1}$ is defined as in (5.3).

Further it is supposed that we know an other number $d_k^* < d_k$ which provides a lower bound for f^*. We define

$$x_{k+1} = \alpha x_k^1 + \beta x_k^2, \quad \frac{\|x_k^1 - x_{k+1}\|}{\|x_k^1 - x_k^2\|} = \frac{f(x_{k,1}) - d_k^*}{(2+n)(f(x_{k,1}) - d_k)}, \tag{5.7}$$

this determines α, β uniquely. In the one-dimensional case we can set $d_k^* = f_k^*$ and work with exact sets of localizations E_k^1, E_k^2. These uniquely determine the point x_k^* so that (5.7) is equivalent to (5.6) when the following updating of E_k^1, E_k^2, d_k, d_k^* is used. (Let us note that the choice (5.7), specially the appearance of $(n+2)$ in the denominator is indicated from the study of the problem of providing lower bounds of complexity for our algorithmic (minimization) problem, see our remark after (2.23).

The value of d_k^* can be updated by

$$d_{k+1}^* = \max\big(d_k^*, f(x_{k+1}) + \\ + \langle g(x_{k+1}), x_k^1 - x_{k+1}\rangle - \langle A_k^i g(x_{k+1}), g(x_{k+1})\rangle^{1/2}\big), \tag{5.8}$$

where $i = 1$, if $f(x_{k+1}) > d_k$ and $i = 2$ otherwise. The meaning of (5.8) is explained by the fact that the minimum of the linear function $f(x_{k+1}) + \langle g(x_{k+1}), x - x_{k+1}\rangle$ over an ellipsoid $E(x_k, A_k)$ is equal to

$$f(x_{k+1}) + \langle g(x_{k+1}), x_k - x_{k+1}\rangle - \langle A_k g(x_{k+1}), g(x_{k+1})\rangle^{1/2}.$$

We shall set

$$d_{k+1} = d_k \text{ if } d_k \in [f(x_{k+1,1}) - \delta(f(x_{k+1,1}) - \\ - d_{k+1}^*), d_{k+1}^* + \gamma(f(x_{k+1,1}) - d_{k+1}^*)] \tag{5.9}$$

and set

$$d_{k+1} = \frac{1}{2}(f(x_{k+1,1}) + d_{k+1}^*) \quad \text{otherwise},$$

where $0 < \delta, \gamma < 1/2$ are parameters which may depend on n.

For the updating of E_k^1 and E_k^2 we shall use Lemma 1, i. e. similar formulas as given in (2.11)–(2.13).

For the definition of E_{k+1}^1 we shall use – besides the values of $f(x_{k+1})$ and $g(x_{k+1})$ – E_k^1, if $f(x_{k+1}) > d_k$, and E_k^2, if $f(x_{k+1}) \leq d_k$, doing this according to the above mentioned formulas. For the definition of E_{k+1}^2 we shall use – besides the values of $f(x_{k+1})$ and $g(x_{k+1})$ – E_k^2, if $d_{k+1} = d_k$ and E_{k+1} otherwise. Note that, when we compute the update of E_k^2, i. e. E_{k+1}^2, it may happen that the prescription given just above leads to an empty set, i. e. the value corresponding to the right hand side of (2.11) becomes greater than 1. In such a case we redefine d_{k+1}^* and d_{k+1} according to (with the usual misuse of notation)

$$d_{k+1}^* := d_{k+1}, \; d_{k+1} := \frac{1}{2}(d_{k+1}^* + f(x_{k+1,1})),$$

as many times as necessary.

Finally let us note that by keeping in memory – at each – step $k-r$ pairs of values from $f(x_j)$, $g(x_j)$, $j = 0, 1, \ldots, k$, we can use them for updating E_k^i, $i = 1, 2$ according to (2.14). This seems to be especially relevant (e. g. for the updating of E_k^2 in such cases when $d_{k+1} \neq d_k$) with the following specification. One sets $r = n+1$, and uses an "exchange algorithm" to update the index sets I_k of $(n+1)$ elements by maintaining the following condition

$$0 \in \text{convex hull } \{g(x_{k_j}),\ k_j \in I_k,\ j = 1, \ldots, n+2\} \tag{5.10}$$

so that (in case of several possibilities) $\max \{k-k_j \mid j = 1, \ldots, n+2\}$ be minimal. □

After the submission of this paper significant advances have been made in linear (convex) programming by the use of interior point methods. Motivated by N. Karmarkar's projective method, the present author proposed a new method for linear (convex) programming based on the notion of an "analytic centre" for polyhedrons and exploiting "analycity" (of the constraints) by the use of rational (multipoint, Pade) extrapolation, with Newton corrections in following the curve $x(\lambda)$ of the "analytic centres" of the polyhedrons $P(\lambda) = \{x \mid \langle c, x\rangle \leqq \lambda,\ Ax-b\}$, $A \in \mathbf{R}^{mxn}$, see [14].

These interior point ("analytic") methods seem to be superior (in the "worst case" sense) to the ellipsoid method *only if* the ratio m/n is not too large, (in the simplest implementation of these methods it is important that all constraints are "simultanously regarded" when choosing the next step, while in ellipsoid methods only one constraint is used at each step).

Acknowledgements. The author thanks Professor J. Stoer for constructive criticism of an earlier version of this paper.

REFERENCES

[1] *Fiala T.* and *Sonnevend Gy.:* An algorithm for the computation of the minimal value of a convex function f over $[0,1]^p$ within accuracy ε, when the values of f are known within accuracy ε_0. *Optimization* **17** (3) (1986), 367 – 377.

[2] *Gershovitch V. I.* and *Shor N. Z.:* The ellipsoid method, its generalizations and applications. *Kibernetika* **18** (5) (1982), 61 – 69 (in Russian).

[3] *Goffin J. L.:* Convergence rates of the ellipsoid method on general convex functions. *Math. of Operations Research* **8** (1) (1983), 135 – 150.

[4] *Khacian L. G.:* Polynomial algorithm for linear programming. *Soviet. Math. Dokl.* **20** (1979), 191 – 194.

[5] *Psenitsnii B. N.:* The Method of Linearization. Nauka, Moscow, 1983 (in Russian).

[6] *Shrader R.:* Ellipsoid methods. In: Modern Applied Mathematics – Optimization and Operations Research (Ed. by B. Korte). North-Holland, Amsterdam, 1982, 266 – 311.

[7] *Stoer J.* and *Bulirsch R.:* Introduction to Numerical Analysis, Springer Verlag, New York, 1980.

[8] *Sonnevend Gy.:* On the optimization of algorithms for function minimization. *USSR. Comp. Math. and Math. Phys.* **17** (3) (1977), 591 – 609.

[9] *Valeev K, G.* and *Finin G. S.:* On dividing the spectrum of a matrix. *Ukr. Dokl. Akad. Nauk.* A8 (1981), 7 – 10 (in Russian).

[10] *Walukiewicz S.:* Ellipsoidal algorithm for linear programming. Res. Report R – 80 – 11, Linköping University, Linköping, 1980.

[11] *Yudin D. B.* and *Nemirowskii A. S.:* Informational Complexity and Effectivity of Optimization Methods. Nauka, Moscow, 1979 (in Russian).
[12] *Sonnevend Gy.:* A modified ellipsoid method with superlinear convergence (resp. finite termination) for C^3 smooth, strongly convex (resp. piecewise linear functions). To appear in *Lect. Notes in Economics and Math. Systems*, Vol. **255** (Ed. by. D. Pallaschke and V. Demyanov).
[13] *Gergó L.:* Numerical experiments with a stabilized ellipsoid method for the minimization of convex, nonsmooth functions. In: First Conf. of Program Designers (Ed. by A. Iványi). Eötvös University, Budapest, 1985, 131 – 136.
[14] *Sonnevend Gy.:* An analytic centre for polyhedrons and new classes of global algorithms for linear (smooth, convex) programming. In: Lect. Notes in Control and Inf. Sciences, Vol. **84** (Ed. by A. Prekopa, J. Szelezsán and B. Straziczky). Springer Verlag, Berlin, 1986, 866 – 876.