The Simplex Method Is Not Always Well Behaved

Włodzimierz Ogryczak
Department of Mathematics, Informatics and Mechanics
Warsaw University
Warsaw, Poland

Submitted by Gene H. Golub

ABSTRACT

This paper deals with the rounding-error analysis of the simplex method for solving linear-programming problems. We prove that in general any simplex-type algorithm is not well behaved, which means that the computed solution cannot be considered as an exact solution to a slightly perturbed problem. We also point out that simplex algorithms with well-behaved updating techniques (such as the Bartels-Golub algorithm) are numerically stable whenever proper tolerances are introduced into the optimality criteria. This means that the error in the computed solution is of a similar order to the sensitivity of the optimal solution to slight data perturbations.

1. INTRODUCTION

This paper deals with the rounding-error analysis of the simplex method for solution of linear-programming (LP) problems. We consider computations performed in floating-point (fl) arithmetic (see [20]). This arithmetic is characterized by the relative computer precision ε. In the case of binary fl arithmetic with t-digit mantissa, ε is equal to 2^{-t}.

We summarize the results of this paper. The concepts of numerical stability and good behavior in linear programming are made precise in Section 2. In Section 3 we give a necessary condition for good behavior of linear-programming algorithms.

Section 4 contains the principal result of the paper. In this section we show that each simplex-type algorithm is not well behaved on a sufficiently

655 Avenue of the Americas, New York, NY 10010 0024-3795/88/$3.50

general class of LP problems (i.e., whenever degeneracy can occur). This result is rather unexpected since the Bartels-Golub simplex algorithm [2] and some others [5–7, 9] preserve good behavior of basic solutions throughout all iterations. We show, however, that this property is not sufficient for good behavior of the whole simplex algorithm.

In Section 5 we state some sufficient conditions for numerical stability of simplex-type algorithms. We show also that the Bartels-Golub simplex algorithm is numerically stable provided that some reasonable tolerances are used in fl implementation of the algorithm and cycling does not occur.

2. PRELIMINARIES

In this section we define what we mean by *numerical stability* and *good behavior* of an algorithm for solving LP problems. We deal with LP problems in the standard form

$$\min\{\mathbf{c}^T\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geqslant \mathbf{0}\}, \tag{2.1}$$

where

$\mathbf{A}$ is an $m \times n$ matrix,
$\mathbf{b}$ is an m-dimensional vector,
$\mathbf{c}$ is an n-dimensional vector,
$\mathbf{x}$ is an n-dimensional vector of variables.

We shall denote the feasible set of the LP problem by Q and the optimal set by S, i.e.,

$$Q = \{\mathbf{x} \in R^n : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geqslant \mathbf{0}\},$$

$$S = \{\mathbf{x} \in Q : \mathbf{c}^T\mathbf{x} \leqslant \mathbf{c}^T\mathbf{z} \text{ for each } \mathbf{z} \in Q\}.$$

We consider only stable LP problems which are solvable and remain solvable for small but otherwise arbitrary perturbations in the data $\mathbf{A}$, $\mathbf{b}$, $\mathbf{c}$. Stability of an LP problem is equivalent to the so-called regularity conditions (see [19]) imposed on the constraints of the problem (2.1) and its dual

$$\max\{\mathbf{b}^T\mathbf{y} : \mathbf{A}^T\mathbf{y} \leqslant \mathbf{c}\} \tag{2.2}$$

where $\mathbf{y}$ is an m-dimensional vector of (dual) variables. Namely, the problem

(2.1) is stable if and only if the following conditions are satisfied:

(1) the matrix $\mathbf{A}$ has full row rank;

(2) there exists a positive feasible vector $\mathbf{x}^0$, i.e., $\mathbf{A}\mathbf{x}^0 = \mathbf{b}$, $\mathbf{x}^0 > \mathbf{0}$;

(3) there exists a vector $\mathbf{y}^0$ that satisfies strongly all the dual constraints, i.e., $\mathbf{A}^{\mathrm{T}}\mathbf{y}^0 < \mathbf{c}$.

We shall denote the class of all the regular problems (2.1) by $\mathbb{D}_0$.

Throughout this paper $\|\cdot\|$ denotes the spectral norm and $\mathrm{dist}(x, U)$ denotes the distance between the vector x and set U, i.e.,

$$\mathrm{dist}(x, U) = \inf\{\|\mathbf{x} - \mathbf{u}\| : \mathbf{u} \in U\}.$$

Further, for any number k let

$$(k)_+ = \begin{cases} k & \text{if} \quad k \geqslant 0, \\ 0 & \text{otherwise.} \end{cases}$$

The operator $(\cdot)_+$ applied to a vector will be understood to be applied componentwise.

Given regularity of the problem, we can consider feasible and optimal sets to slightly perturbed LP problems. We introduce the slightly perturbed feasible set $Q^\varepsilon = Q^\varepsilon(k_1, k_2)$ and the slightly perturbed optimal set $S^\varepsilon = S^\varepsilon(k_1, k_2, k_3)$ defined as follows:

$$Q^\varepsilon(k_1, k_2) = \{\mathbf{x} \in R^n : \mathbf{x} \geqslant \mathbf{0} \text{ and } (\mathbf{A} + \delta\mathbf{A})\mathbf{x} = \mathbf{b} + \delta\mathbf{b} \text{ for some data perturbations } \delta\mathbf{A} \text{ and } \delta\mathbf{b} \text{ such that } \|\delta\mathbf{A}\| \leqslant k_1\varepsilon\|\mathbf{A}\| \text{ and } \|\delta\mathbf{b}\| \leqslant k_2\varepsilon\|\mathbf{b}\|\},$$

$$S^\varepsilon(k_1, k_2, k_3) = \{x \in Q^\varepsilon(k_1, k_2) : \text{there exists a perturbation } \delta\mathbf{c} \text{ such that } \|\delta\mathbf{c}\| \leqslant k_3\varepsilon\|\mathbf{c}\| \text{ and } (\mathbf{c} + \delta\mathbf{c})^{\mathrm{T}}\mathbf{x} \leqslant (\mathbf{c} + \delta\mathbf{c})^{\mathrm{T}}\mathbf{z} \text{ for each } \mathbf{z} \in Q^\varepsilon(k_1, k_2)\},$$

where k_1, k_2, k_3 are some arbitrary numbers. Note that $S^\varepsilon \subset Q^\varepsilon$ and both sets are nonempty for each sufficiently small ε.

Let F be an algorithm that gives for any data complex $\mathbf{d} = (\mathbf{A}, \mathbf{b}, \mathbf{c}) \in \mathbb{D}_0$, in a finite number of elementary operations, an optimal solution to the problem (2.1). Let $\mathbf{x}^\varepsilon$ denote the vector (solution) generated by the algorithm F with all computations performed in fl arithmetic (with the relative precision ε). Apart from the trivial case when $\mathbf{b} = \mathbf{0}$ and therefore $S = Q = \{\mathbf{0}\}$, the feasible set Q does not include the zero vector. Therefore, we shall assume that $\mathbf{x}^\varepsilon \neq \mathbf{0}$. Good behavior and numerical stability of the algorithm F are defined as follows.

DEFINITION 2.1. An algorithm F is called *well behaved* (or equivalently F has *good behavior*) on a class $\mathbb{D} \subset \mathbb{D}_0$ if there exist constants $k_i = k_i(\mathbb{D})$ $(i = 0, 1, 2, 3)$ such that for each $\mathbf{d} \in \mathbb{D}$ and for each sufficiently small ε the computed solution x^{ε} satisfies

$$\operatorname{dist}\left(\mathbf{x}^{\varepsilon}, S^{\varepsilon}(k_1, k_2, k_3)\right) \leqslant k_0 \varepsilon \|\mathbf{x}^{\varepsilon}\|.$$

DEFINITION 2.2. An algorithm F is called *numerically stable* on a class $\mathbb{D} \subset \mathbb{D}_0$ if there exist constants $k_i = k_i(\mathbb{D})$ $(i = 1,2,3)$ such that for each $\mathbf{d} \in \mathbb{D}$ and for each sufficiently small ε the computed solution $\mathbf{x}^{\varepsilon}$ satisfies

$$\operatorname{dist}(\mathbf{x}^{\varepsilon}, S) \leqslant k_4\left[\varepsilon\|\mathbf{x}^{\varepsilon}\| + \sup\left\{\operatorname{dist}(\mathbf{x}, S) : \mathbf{x} \in S^{\varepsilon}(k_1, k_2, k_3)\right\}\right].$$

In other words, a well-behaved algorithm generates a slightly perturbed solution to a slightly perturbed data complex, whereas a numerically stable algorithm generates a solution with the same error bound as a well-behaved algorithm (see [13]). It is easy to verify that good behavior implies numerical stability but not, in general, vice versa.

Recall now some elementary facts connected with numerical solution of linear systems. An algorithm for solving a linear system $\mathbf{Bx} = \mathbf{b}$ with $m \times m$ nonsingular matrix $\mathbf{B}$ is well behaved if it gives a computed solution $\mathbf{x}^{\varepsilon}$ satisfying

$$(\mathbf{B} + \delta\mathbf{B})\mathbf{x}^{\varepsilon} = \mathbf{b} + \delta\mathbf{b}$$

with $\|\delta\mathbf{B}\|$ of order $\varepsilon\|\mathbf{B}\|$ and $\|\delta\mathbf{b}\|$ of order $\varepsilon\|\mathbf{b}\|$. The vector $\mathbf{x}^{\varepsilon}$ generated by a well-behaved algorithm approximates the exact solution $\mathbf{B}^{-1}\mathbf{b}$ with relative error of order $\varepsilon\|\mathbf{B}\|\,\|\mathbf{B}^{-1}\|\,\|\mathbf{x}^{\varepsilon}\|$, where $\|\mathbf{B}\|\,\|\mathbf{B}^{-1}\| = \varkappa$ is the condition number of the matrix $\mathbf{B}$. This property is usually treated as a definition of numerical stability.

3. NECESSARY CONDITION FOR GOOD BEHAVIOR

In this section we give a necessary condition for good behavior of an algorithm for solving LP problems. This condition has crucial meaning for the proof of our principal result in the next section.

According to Definition 2.1 an algorithm F is well behaved if $\operatorname{dist}(\mathbf{x}^{\varepsilon}, S^{\varepsilon})$ is of order $\varepsilon\|\mathbf{x}^{\varepsilon}\|$. The set S^{ε} is, however, a subset of Q^{ε}, and therefore the inequality

$$\operatorname{dist}(\mathbf{x}^\varepsilon, S^\varepsilon) \geqslant \operatorname{dist}(\mathbf{x}^\varepsilon, Q^\varepsilon)$$

holds. Thus we can state the following remark.

REMARK 3.1. If an algorithm F is well behaved on a class $\mathbb{D} \subset \mathbb{D}_0$, then there exist constants $k_i = k_i(\mathbb{D})$ $(i = 0, 1, 2)$ such that for each $\mathbf{d} \in \mathbb{D}$ and for each sufficiently small ε the computed solution $\mathbf{x}^\varepsilon$ satisfies

$$\operatorname{dist}(\mathbf{x}^\varepsilon, Q^\varepsilon(k_1, k_2)) \leqslant k_0\|\mathbf{x}^\varepsilon\|.$$

In other words, a well-behaved algorithm for solving LP problems is also well behaved with respect to solving the simpler problems: find a feasible vector $\mathbf{x}$. For this reason, we analyze in detail good behavior with respect to solving linear systems

$$\mathbf{Ax} = \mathbf{b}, \tag{3.1}$$

$$\mathbf{x} \geqslant \mathbf{0}, \tag{3.2}$$

as a necessary condition for good behavior with respect to solving the whole LP problem (2.1).

It is known (see e.g. [21]) that an algorithm for solving a linear system $\mathbf{Bx} = \mathbf{b}$ with a square matrix $\mathbf{B}$ is well behaved if and only if

$$\|\mathbf{b} - \mathbf{Bx}^\varepsilon\| \text{ is of order } \varepsilon(\|\mathbf{B}\|\,\|\mathbf{x}^\varepsilon\| + \|\mathbf{b}\|).$$

We extend this residual condition to the system (3.1)–(3.2). In this system there are two separate residual vectors: the vector $\mathbf{b} - \mathbf{Ax}$ connected with the equality $\mathbf{Ax} = \mathbf{b}$, and the vector $(-\mathbf{x})_+$ connected with the inequality $\mathbf{x} \geqslant \mathbf{0}$. Lemma 3.2 utilizes both these vectors to state the residual criterion of good behavior with respect to solving linear system (3.1)–(3.2).

LEMMA 3.2. *For each algorithm F the following statements are equivalent*:

(1) *there exist constants* $k_i = k_i(\mathbb{D})$, $(i = 0, 1, 2)$ *such that for each* $\mathbf{d} \in \mathbb{D}$ *and for each sufficiently small* ε *the computed solution* $\mathbf{x}^\varepsilon$ *satisfies*

$$\operatorname{dist}(\mathbf{x}^\varepsilon, Q^\varepsilon(k_1, k_2)) \leqslant k_0\varepsilon\|\mathbf{x}^\varepsilon\|;$$

(2) *there exist constants* $k_i = k_i(\mathbb{D})$ $(i = 5,6)$ *such that for each* $\mathbf{d} \in \mathbb{D}$ *and for each sufficiently small* ε *the computed solution* $\mathbf{x}^\varepsilon$ *satisfies*

$$\|\mathbf{b} - \mathbf{A}\mathbf{x}^\varepsilon\| \leqslant k_5(\|\mathbf{A}\|\,\|\mathbf{x}^\varepsilon\| + \|\mathbf{b}\|) \tag{3.3}$$

and

$$\|(-\mathbf{x}^\varepsilon)_+\| \leqslant k_6\varepsilon\|\mathbf{x}^\varepsilon\|. \tag{3.4}$$

Proof. Assume that statement (1) is valid. It means that there exist constants k_0, k_1, k_2 such that for each $\mathbf{d} \in \mathbb{D}$ and for sufficiently small ε there exist perturbations $\delta\mathbf{A}, \delta\mathbf{b}, \delta\mathbf{x}$ such that

$$\|\delta\mathbf{x}\| \leqslant k_0\varepsilon\|\mathbf{x}^\varepsilon\|, \tag{3.5}$$

$$\|\delta\mathbf{A}\| \leqslant k_1\varepsilon\|\mathbf{A}\|, \tag{3.6}$$

$$\|\delta\mathbf{b}\| \leqslant k_2\varepsilon\|\mathbf{b}\|, \tag{3.7}$$

$$(\mathbf{A} + \delta\mathbf{A})(\mathbf{x}^\varepsilon - \delta\mathbf{x}) = \mathbf{b} + \delta\mathbf{b}, \tag{3.8}$$

$$\mathbf{x}^\varepsilon - \delta\mathbf{x} \geqslant \mathbf{0}. \tag{3.9}$$

From these relations we get

$$\begin{aligned}\|\mathbf{b} - \mathbf{A}\mathbf{x}^\varepsilon\| &= \|\delta\mathbf{b} - \delta\mathbf{A}\mathbf{x}^\varepsilon + \mathbf{A}\,\delta\mathbf{x} + \delta\mathbf{A}\,\delta\mathbf{x}\| \\ &\leqslant k_2\varepsilon\|\mathbf{b}\| + (k_0 + k_1 + \varepsilon k_0 k_1)\varepsilon\|\mathbf{A}\|\,\|\mathbf{x}^\varepsilon\|\end{aligned}$$

and

$$\|(-\mathbf{x}^\varepsilon)_+\| \leqslant \|\delta\mathbf{x}\| \leqslant k_0\varepsilon\|\mathbf{x}^\varepsilon\|.$$

Thus statement (2) is valid.

Assume now that the statement (2) is valid. We shall define perturbations $\delta\mathbf{A}$, $\delta\mathbf{b}$, and $\delta\mathbf{c}$ satisfying the conditions (3.5)–(3.9). At first let $\delta\mathbf{x} = -(-\mathbf{x}^\varepsilon)_+$. Due to (3.4), the inequality (3.5) holds with the constants $k_0 = k_6$. Further, due to (3.3) we can define the perturbation $\delta\mathbf{b}$ such that

$$\|\delta\mathbf{b}\| \leqslant k_5\varepsilon\|\mathbf{b}\|$$

and

$$\|\mathbf{b}+\delta\mathbf{b}-\mathbf{A}\mathbf{x}^{\varepsilon}\| \leqslant k_5\varepsilon\|\mathbf{A}\|\,\|\mathbf{x}^{\varepsilon}\|.$$

So the inequality (3.7) holds with the constant $k_2 = k_5$. Furthermore, we define the perturbation $\delta\mathbf{A}$ as follows:

$$\delta\mathbf{A} = \frac{1}{(x^{\varepsilon})_+^{\mathrm{T}}(x^{\varepsilon})_+}\left[\mathbf{b}+\delta\mathbf{b}-\mathbf{A}(\mathbf{x}^{\varepsilon})_+\right](\mathbf{x}^{\varepsilon})_+^{\mathrm{T}}.$$

One can easily verify that the conditions (3.8) and (3.9) are satisfied for such perturbations. Moreover

$$\begin{aligned}\|\delta\mathbf{A}\| &\leqslant \frac{\|(\mathbf{x}^{\varepsilon}\|)_+\|}{(\mathbf{x}^{\varepsilon})_+^{\mathrm{T}}(\mathbf{x}^{\varepsilon})_+}\|\mathbf{b}+\delta\mathbf{b}-\mathbf{A}(\mathbf{x}^{\varepsilon})_+\|\\
&\leqslant \frac{1}{\|(\mathbf{x}^{\varepsilon})_+\|}\left(k_4\varepsilon\|\mathbf{A}\|\,\|\mathbf{x}^{\varepsilon}\|+\|\mathbf{A}\|\,\|(-\mathbf{x}^{\varepsilon})_+\|\right)\\
&\leqslant \frac{k_4+k_5}{1-\varepsilon k_5}\varepsilon\|\mathbf{A}\|.\end{aligned}$$

So, for sufficiently small ε the inequality (3.6) holds, and finally, statement (1) is valid.

The proof is completed.

COROLLARY 3.3. *An algorithm for solving the linear systems* (3.1)–(3.2) *is well behaved on a class* $\mathbb{D} \subset \mathbb{D}_0$ *if and only if there exist constants* $k_i = k_i(\mathbb{D})$ $(i = 5,6)$ *such that for each* $\mathbf{d} \in \mathbb{D}$ *and for each sufficiently small* ε *the computed solution* $\mathbf{x}^{\varepsilon}$ *satisfies the inequalities* (3.3) *and* (3.4).

COROLLARY 3.4. *If an algorithm F is well behaved on a class* $\mathbb{D} \subset \mathbb{D}_0$, *then there exist constants* $k_i = k_i(\mathbb{D})$ $(i = 5,6)$ *such that for each* $\mathbf{d} \in \mathbb{D}$ *and for each sufficiently small* ε *the computed solution* $\mathbf{x}^{\varepsilon}$ *satisfies the inequalities* (3.3) *and* (3.4).

In the next section we shall use the necessary condition stated by Corollary 3.4 in order to show that any simplex-type algorithm is not well behaved on a sufficiently general class of LP problems.

4. PRINCIPAL RESULT

The essence of the simplex method is that only basic solutions are considered and the best basic solution is chosen as optimal. In this section we show that each method that generates some basic solution as an optimal solution is not well behaved on any sufficiently general class of LP problems.

Recall that a basic solution to the system (3.1) is defined as follows. Let $\mathbf{B}$ be a basis of the matrix $\mathbf{A}$, i.e., an $m \times m$ nonsingular matrix consisting of some columns of the matrix $\mathbf{A}$. The nonbasic part of $\mathbf{A}$ we shall denote by $\mathbf{N}$. The basic solution generated by the basis $\mathbf{B}$ is defined by the linear system

$$\mathbf{B}\mathbf{x}_{\mathbf{B}} = \mathbf{b} \quad \text{and} \quad \mathbf{x}_{\mathbf{N}} = \mathbf{0} \tag{4.1}$$

where $\mathbf{x}_{\mathbf{B}}$ and $\mathbf{x}_{\mathbf{N}}$ denote the basic and nonbasic parts of vector $\mathbf{x}$, respectively. In other words, nonbasic coefficients of the basic solution are directly defined as equal to zero, while basic coefficients are defined as a solution of the basic linear system $\mathbf{B}\mathbf{x}_{\mathbf{B}} = \mathbf{b}$. Any algorithm for solving the LP problem (2.1) or the linear system (3.1)–(3.2) that generates a solution according to this scheme will be called a simplex-type algorithm.

The basic solution is feasible if $\mathbf{x}_{\mathbf{B}} = \mathbf{B}^{-1}\mathbf{b} \geqslant \mathbf{0}$. Roundoff errors can cause the computed vector $\mathbf{x}_{\mathbf{B}}$ to violate this inequality even if the exactly calculated vector would be feasible. For this reason, small negative coefficients of the computed vector $\mathbf{x}_{\mathbf{B}}$ are sometimes set equal to zero. Such an algorithm we shall also regard as a simplex-type algorithm.

Definition 4.1. An algorithm for solving the LP problem (2.1) or the linear system (3.1)–(3.2) is said to be *simplex-type* if it generates a solution $\mathbf{x}$ in such a way that for some basis $\mathbf{B}$:

(1) $\mathbf{x}_{\mathbf{N}} = \mathbf{0}$,

(2) $\mathbf{x}_{\mathbf{B}} = \mathbf{w}$ or $\mathbf{x}_{\mathbf{B}} = (\mathbf{w})_{+}$, where $\mathbf{w}$ is a solution of the basic system $\mathbf{B}\mathbf{w} = \mathbf{b}$.

In theoretical considerations one frequently makes an assumption that all the feasible basic solutions are nondegenerate, i.e., the vectors $\mathbf{x}_{\mathbf{B}} = \mathbf{B}^{-1}\mathbf{b}$ are strictly positive. This assumption significantly simplifies the simplex method but it is not necessary for convergence of the method. Furthermore, this assumption stands in contradiction with linear-programming practice, since in practical LP models degeneracy usually occurs. So it is necessary to allow degeneracy of the problem in analyzing computational properties of the simplex method. For this reason, we do not regard as sufficiently general any class of nondegenerate LP problems.

Further, we do not treat as sufficiently general any class of LP problems with a special structure of the matrix, in the following sense: we assume that in a sufficiently general class of problems a quantity of order $\varepsilon\|\mathbf{B}\|\,\|\mathbf{B}^{-1}\|\,\|\mathbf{x}^{\varepsilon}\|$ cannot be considered as a quantity of order $\varepsilon\|\mathbf{x}^{\varepsilon}\|$.

Having defined what we mean by a simplex-type algorithm and a sufficiently general class of LP problems, we can state the following principal result.

THEOREM 4.2. *If a class* $\mathbb{D}$ *of regular* LP *problems is sufficiently general, then simplex-type algorithms are not well behaved on the class* $\mathbb{D}$.

Proof. According to Definition 4.1 a simplex-type algorithm implemented in fl arithmetic generates a solution $\mathbf{x}^{\varepsilon}$ such that

(1) $\mathbf{x}_{\mathbf{N}}^{\varepsilon} = \mathbf{0}$,

(2) $\mathbf{x}_{\mathbf{B}}^{\varepsilon} = \mathbf{w}^{\varepsilon}$ or $\mathbf{x}_{\mathbf{B}}^{\varepsilon} = (\mathbf{w}^{\varepsilon})_{+}$, where $\mathbf{w}^{\varepsilon}$ is a solution of the system $\mathbf{Bw} = \mathbf{b}$ (computed in fl).

Suppose that for each $\mathbf{d} \in \mathbb{D}$ and for each sufficiently small ε the proper basis $\mathbf{B}$ is identified as optimal. If that is not true, then the distance $\operatorname{dist}(\mathbf{x}^{\varepsilon}, S)$ does not tend to zero as ε tends to zero, and so the algorithm is obviously not stable and not well behaved.

Even if the linear system $\mathbf{Bw} = \mathbf{b}$ is solved by a well-behaved technique, we get a solution $\mathbf{w}^{\varepsilon}$ with error $\|\mathbf{w}^{\varepsilon} - \mathbf{B}^{-1}\mathbf{b}\|$ of order $\varepsilon\|\mathbf{B}\|\,\|\mathbf{B}^{-1}\|\,\|\mathbf{w}^{\varepsilon}\|$. So if degeneracy of the basis occurs, then $(-\mathbf{w}^{\varepsilon})_{+}$ is of order $\varepsilon\|\mathbf{B}\|\,\|\mathbf{B}^{-1}\|\,\|\mathbf{w}^{\varepsilon}\|$. Thus the simplex-type algorithm with directly defined basic solution (i.e., $\mathbf{x}_{\mathbf{B}}^{\varepsilon} = \mathbf{w}^{\varepsilon}$) is not well behaved (by Corollary 3.4), since the inequality (3.4) is not valid.

If the basic solution is defined with truncation [i.e., $\mathbf{x}_{\mathbf{B}}^{\varepsilon} = (\mathbf{w}^{\varepsilon})_{+}$], then the inequality (3.4) holds but the inequality (3.3) is not valid. So in this case too the algorithm is not well behaved.

Thus the proof is completed. ∎

Note that the proof of Theorem 4.2 is based on Lemma 3.2. So our result is also valid for simplex-type algorithms considered with respect to solving linear systems (3.1)–(3.2).

In the 1970s there were published a few papers (see [2,3,5–7]) which presented some well-behaved forms of the simplex method. The most famous and widely used is the Bartels-Golub algorithm [2] with LU basis decomposition. However, in these papers good behavior of simplex algorithms was considered only with respect to updating of the basis factorization. It was clearly shown for the Bartels-Golub algorithm (see [1]). In other words, these algorithms guarantee only that throughout all simplex steps the basic system $\mathbf{Bw} = \mathbf{b}$ and its dual analogue are solved by a well-behaved technique.

Thus our analysis does not contradict these results. Rather, it can be considered as an extension, since we analyze whether good behavior in solving basic systems is sufficient for good behavior of the whole simplex algorithm or not. The results of our analysis allow us to conclude that if a sufficiently general class of LP problems is considered, then none of the known simplex algorithms (including the Bartels-Golub algorithm) are well behaved and there is no possibility of constructing a simplex-type algorithm that shall be well behaved on this class.

5. NUMERICAL STABILITY OF THE SIMPLEX METHOD

In the previous section we have shown there does not exist any simplex-type algorithm that is well behaved on a sufficiently general class of LP problems. Nevertheless, a simplex-type algorithm can be numerically stable with respect to Definition 2.2. In this section we give some sufficient conditions for numerical stability of simplex-type algorithms. As the background of our analysis we use the following theorem.

THEOREM 5.1. *Let $\mathbb{D} \subset \mathbb{D}_0$ be an arbitrary class of regular LP problems. If for each $\mathbf{d} \in \mathbb{D}$ and for each sufficiently small ε a simplex-type algorithm F produces a solution $\mathbf{x}^\varepsilon$ generated by the true optimal basis $\mathbf{B}$, and if the basic system $\mathbf{Bw} = \mathbf{b}$ is solved by a well-behaved technique, then the algorithm F is numerically stable on the class $\mathbb{D}$.*

Proof. Since the optimal basis $\mathbf{B}$ is properly identified and the basic system $\mathbf{Bw} = \mathbf{b}$ is solved by a well-behaved technique, we state that

$$\operatorname{dist}(\mathbf{x}^\varepsilon, S) \text{ is of order } \varepsilon\|\mathbf{B}\|\,\|\mathbf{B}^{-1}\|\,\|\mathbf{x}^\varepsilon\|.$$

More precisely, there exists a constant $k_7 = k_7(\mathbb{D})$ such that for each $\mathbf{d} \in \mathbb{D}$ and for each sufficiently small ε the computed solution $\mathbf{x}^\varepsilon$ satisfies

$$\operatorname{dist}(\mathbf{x}^\varepsilon, S) \leqslant k_7\varepsilon\|\mathbf{B}\|\,\|\mathbf{B}^{-1}\|\,\|\mathbf{x}^\varepsilon\|.$$

On the other hand, for each optimal basis there exist perturbations $\delta\mathbf{A}$, $\delta\mathbf{b}$, and $\delta\mathbf{c}$ of order $\varepsilon\|\mathbf{A}\|$, $\varepsilon\|\mathbf{b}\|$, $\varepsilon\|\mathbf{c}\|$, respectively, such that this basis remains optimal for the perturbed problem. So we can state that

$$\sup\{\operatorname{dist}(\mathbf{x}, S) : \mathbf{x} \in S^{\varepsilon}(k_1, k_2, k_3)\} \geqslant k_8 \varepsilon \|\mathbf{B}\| \, \|\mathbf{B}^{-1}\| \, \|\mathbf{B}^{-1}\mathbf{b}\|$$

for some $k_8 = k_8(k_1, k_2, k_3)$.

Finally, we state that there exist constants $k_i = k_i(\mathbb{D})$ $(i = 1,2,3,4)$ such that for each $\mathbf{d} \in \mathbb{D}$ and for each sufficiently small ε the computed solution $\mathbf{x}^{\varepsilon}$ satisfies

$$\operatorname{dist}(\mathbf{x}^{\varepsilon}, S) \leqslant k_4 \left[\varepsilon \|\mathbf{x}^{\varepsilon}\| + \sup\{\operatorname{dist}(\mathbf{x}, S) : \mathbf{x} \in S^{\varepsilon}(k_1, k_2, k_3)\}\right].$$

Thus the proof is completed. ∎

By Theorem 5.1 we conclude that the Bartels-Golub algorithm and the other simplex algorithms with well-behaved techniques for updating the basis factorization are numerically stable provided that they properly identify an optimal basis. However, in papers dealing with numerical analysis of the simplex method the problem of optimal basis identification was usually not of interest. We now concentrate on this problem.

Any simplex method (primal, dual, self-dual, etc.) has the same general scheme. According to some rule (specific to the method) a sequence of bases is generated. For each basis $\mathbf{B}$ the basic solution $\mathbf{x}$ and the dual solution $\mathbf{y}$ (the so-called vector of simplex multipliers) are calculated, i.e., the linear systems

$$\mathbf{B}\mathbf{x}_{\mathbf{B}} = \mathbf{b} \tag{5.1}$$

and

$$\mathbf{B}^{\mathrm{T}}\mathbf{y} = \mathbf{c}_{\mathbf{B}} \tag{5.2}$$

are solved. The basis $\mathbf{B}$ is identified as optimal if the solutions of linear systems (5.1) and (5.2) satisfy inequalities

$$\mathbf{x}_{\mathbf{B}} \geqslant \mathbf{0} \tag{5.3}$$

and

$$\mathbf{c} - \mathbf{A}^{\mathrm{T}}\mathbf{y} \geqslant \mathbf{0}, \tag{5.4}$$

i.e., if both primal and dual basic solutions are feasible. The quantities $d_j = c_j - \mathbf{y}^{\mathrm{T}}\mathbf{A}_j$, where $\mathbf{A}_j$ denotes column j of the matrix $\mathbf{A}$, are referred to as *reduced costs.*

Certainly, if the linear systems (5.1) and (5.2) are solved in fl arithmetic, then roundoff errors may cause the inequalities (5.3) and (5.4) not to be valid for any computed basic solution. In other words, it is possible that no basic solution will be found to be optimal. Therefore, in practice it is necessary to introduce some tolerances into the optimality criteria (5.3) and (5.4). In effect, we get inequalities

$$\mathrm{fl}\big((\mathbf{B}^{-1}\mathbf{b})_i\big) \geqslant -t_i \qquad \text{for} \quad i=1,2,\ldots,m, \tag{5.5}$$

$$\mathrm{fl}\big(c_j - \mathbf{c}_{\mathbf{B}}^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{A}_j\big) \geqslant -t_j^* \qquad \text{for} \quad j=1,2,\ldots,n, \tag{5.6}$$

where t_i and t_j^* are tolerances for feasibility and optimality (dual feasibility), respectively. The tolerances, obviously, take some positive values. They may be defined once for the whole algorithm or redefined at each simplex step. All linear-programming codes use such tolerances (see [14, 17]). Thus the problem of optimal basis identification in fl arithmetic can be considered as a problem of tolerance definition. It turns out that a natural approach to tolerance definition leads to numerical stability of the simplex method. Definition 5.2 formalizes a property of the tolerances sufficient for numerical stability.

Definition 5.2. The tolerances t_i and t_j^* are called *error-estimating* if they are of similar order to the bounds on computational errors of the corresponding quantities, i.e., if

(i) one has

$$t_i \geqslant \big|\mathrm{fl}\big((\mathbf{B}^{-1}\mathbf{b})_i\big) - (\mathbf{B}^{-1}\mathbf{b})_i\big| \qquad \text{for} \quad i=1,2,\ldots,m; \tag{5.7}$$

(ii) one has

$$t_j^* \geqslant \big|\mathrm{fl}\big(c_j - \mathbf{c}_{\mathbf{B}}^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{A}_j\big) - \big(c_j - \mathbf{c}_{\mathbf{B}}^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{A}_j\big)\big| \qquad \text{for} \quad j=1,2,\ldots,n; \tag{5.8}$$

(iii) t_i and t_j^* tend to zero as ε tends to zero.

Theorem 5.3. *Let $\mathbb{D} \subset \mathbb{D}_0$ be an arbitrary class of regular* LP *problems. If for each $\mathbf{d} \in \mathbb{D}$ a simplex algorithm F produces a solution $\mathbf{x}^\varepsilon$ generated by a basis $\mathbf{B}$ for which the inequalities* (5.5) *and* (5.6) *are satisfied with some error-estimating tolerances, and if the basic systems $\mathbf{B}\mathbf{w} = \mathbf{b}$ and $\mathbf{B}^{\mathrm{T}}\mathbf{y} = \mathbf{c}_{\mathbf{B}}$*

are solved by well-behaved techniques, then the algorithm F is numerically stable on the class $\mathbb{D}$.

Proof. Since the tolerances t_i and t_j^* are on the level of the corresponding error bounds, each optimal basis satisfies the inequalities (5.5) and (5.6).

Consider now the computed vectors $(\mathbf{x}^\varepsilon)_+$ and $\mathbf{y}^\varepsilon$ generated by some basis $\mathbf{B}$ (perhaps nonoptimal for the original problem) that satisfies the inequalities (5.5) and (5.6). They can be treated as exact solutions of linear systems

$$\mathbf{A}\mathbf{x} = \mathbf{b} + \mathbf{r} \quad \text{and} \quad \mathbf{B}^{\mathrm{T}}\mathbf{y} = \mathbf{c}_{\mathbf{B}} + \mathbf{r}_{\mathbf{B}}^*,$$

where $\mathbf{r} = \mathbf{A}(\mathbf{x}^\varepsilon)_+ - \mathbf{b}$ and $\mathbf{r}_{\mathbf{B}}^* = \mathbf{B}^{\mathrm{T}}\mathbf{y}^\varepsilon - \mathbf{c}_{\mathbf{B}}$. Further, the vectors $(\mathbf{x}^\varepsilon)_+$ and $\mathbf{y}^\varepsilon$ can be also treated as optimal solutions (primal and dual, respectively) to the perturbed LP problem

$$P_\varepsilon : \min\left\{ (\mathbf{c} + \mathbf{r}^*)^{\mathrm{T}}\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b} + \mathbf{r},\ \mathbf{x} \geqslant \mathbf{0} \right\},$$

where $\mathbf{r}_{\mathbf{N}}^* = (\mathbf{N}^{\mathrm{T}}\mathbf{y}^\varepsilon - \mathbf{c}_{\mathbf{N}})_+$.

The norms $\|\mathbf{r}\|$ and $\|\mathbf{r}^*\|$ are obviously proportional to ε, since the tolerances t_i and t_j^* are error-estimating. So, by the theory of LP stability (see [19]) there exists a constant k_9 such that for each sufficiently small ε

$$\operatorname{dist}\big(((\mathbf{x}^\varepsilon)_+, \mathbf{y}^\varepsilon), S \times S^*\big) \leqslant k_9 \varepsilon,$$

where S^* denotes the optimal set to the dual problem (2.2). On the other hand, there exists a constant k_{10} such that for sufficiently small ε

$$\|(\mathbf{x}^\varepsilon)_+ - \mathbf{B}^{-1}\mathbf{b}\| \leqslant k_{10}\varepsilon$$

and

$$\|\mathbf{y}^\varepsilon - \mathbf{B}^{-\mathrm{T}}\mathbf{c}_{\mathbf{B}}\| \leqslant k_{10}\varepsilon.$$

Taking into account all the above inequalities, we conclude that for each $\mathbf{d} \in \mathbb{D}$ and for each sufficiently small ε only optimal bases satisfy the inequalities (5.5) and (5.6).

Thus by Theorem 5.1 the proof is completed. ∎

Note that any simplex algorithm with a well-behaved technique for basis-factorization updating satisfies the assumptions of Theorem 5.3, provided that the inequalities (5.5) and (5.6) with error-estimating tolerances are used as optimality criteria and cycling does not occur. The latter is necessary in order to guarantee that after a finite number of steps the algorithm will find a basis that satisfies the optimality criteria, i.e., the algorithm generates some solution. We formalize this result for the Bartels-Golub algorithm.

COROLLARY 5.4. *Let* $\mathbb{D} \subset \mathbb{D}_0$ *be an arbitrary class of regular* LP *problems. If cycling does not occur and if the inequalities* (5.5) *and* (5.6) *with error-estimating tolerances are used as optimality criteria, then the Bartels-Golub algorithm is numerically stable on the class* $\mathbb{D}$.

Analyzing carefully the inequalities (5.7) and (5.8), one can easily find (see [15] or [16] for details) that the simplest formulae for error-estimating tolerances take the following form:

$$t_i \approx \|(\mathbf{B}^{-1})_i\| \, \|\mathbf{B}\mathbf{x}_{\mathrm{T}}^{\varepsilon} - \mathbf{b}\| \qquad \text{for} \quad i = 1,2,\dots,m, \tag{5.9}$$

$$t_j^* \approx \left\| \begin{pmatrix} \mathbf{B}^{-1}\mathbf{A}_j \\ -\mathbf{e}_j \end{pmatrix} \right\| \|\mathbf{B}^{\mathrm{T}}\mathbf{y}^{\varepsilon} - \mathbf{c}_{\mathbf{B}}\| \qquad \text{for nonbasic } j, \tag{5.10}$$

where $(\mathbf{B}^{-1})_i$ denotes the ith row of $\mathbf{B}^{-1}$, and $\mathbf{e}_j$ is the jth unit vector in the $(n-m)$-dimensional space of nonbasic variables. The basic residual vectors $\mathbf{B}\mathbf{x}_{\mathbf{B}}^{\varepsilon} - \mathbf{b}$ and $\mathbf{B}^{\mathrm{T}}\mathbf{y}^{\varepsilon} - \mathbf{c}_{\mathbf{B}}$ are usually computed at each simplex step for control of the so-called reinvert mechanism. Large values of their norms mean the loss of accuracy of basic solutions and indicate the need for refactorization of the current basis. The factors

$$\|(\mathbf{B}^{-1})_i\| \quad \text{and} \quad \left\| \begin{pmatrix} \mathbf{B}^{-1}\mathbf{A}_j \\ -\mathbf{e}_j \end{pmatrix} \right\|$$

are available in advanced simplex algorithms, since they are used in the pivot selection mechanism. The vectors $\begin{pmatrix} \mathbf{B}^{-1}\mathbf{A}_j \\ -\mathbf{e}_j \end{pmatrix}$ (for nonbasic indices j) point down the $n-m$ edges of the feasible set that emanate from the current vertex $\mathbf{x}$. Similarly, the vectors $(\mathbf{B}^{-1})_i$ (for $= 1,2,\dots,m$) point down the m edges of the dual feasible region that emanate from the current vertex $\mathbf{y}$. Their spectral norms are used as normalizing scales in the so-called steepest-edge strategy, which is known to be very effective in reducing the number of simplex steps. The direct computation of all the normalizing scales at each simplex step is too expensive. However, they can be cheaply computed by some special updating formulae, especially when triangular basis factorization is used (see [8]). Most linear-programming codes compute the normalizing scales only approximately, using the so-called DEVEX technique (see [10]). Thus all the quantities used in the formulae (5.9)–(5.10) can be estimated in advanced simplex codes.

Unfortunately, it turns out that commercial LP codes do not always allow the user to define appropriate error-estimating tolerances. For instance, our experience with solving hard (ill-conditioned) LP problems with the MPSX/370 package [11] shows that its tolerances are independent of the global error bounds. The package is equipped with relative tolerances, but they work in a specific manner. Namely, the relative tolerances define the value t_i or t_j^* as a product of a given parameter by the largest absolute value of elements computed in solving the linear system (5.1) or (5.2), respectively. In such a definition of the tolerances the condition number of the corresponding linear system is ignored. In other words, the tolerances are too closely related to error bounds for single operations instead of being related to global error bounds. As a result, there is a clear failure of the simplex procedure on ill-conditioned LP problems (see [15]).

Note that Theorem 5.3 does not require good behavior of basis factorization updating. Only basic solutions generated by bases that satisfy the optimality criteria must be calculated with high accuracy (i.e., by a well-behaved technique). So numerical stability of the whole simplex algorithm can also be achieved for an unstable updating technique, provided that the tolerances are properly defined and the accuracy of solutions that satisfy the optimality criteria is improved by using a direct well-behaved technique or the iterative refinement process (see [12]). This unexpected conclusion suggests that a proper definition of the tolerances seems to have greater importance for numerical stability of the simplex algorithm than a well-behaved technique for basis factorization updating. This can be regarded as some explanation for why practitioners have preferred to sacrifice good behavior of updating for apparent advantages in storage and computational effort (see [4]). Of course, an implementation of the simplex method may never reach a basis satisfying the optimality criteria if an unstable updating procedure is used.

6. CONCLUSION

The importance of numerical stability in methods used for the solution of LP problems was appreciated in the 1970s. There were proposed a few well-behaved techniques for updating the basis factorization in the simplex method. The best known was the Bartels-Golub algorithm, which was also successfully adapted to handling sparsity of the matrix (see e.g. [18]). Stability analysis for these algorithms was, however, limited to several simplex steps. In other words, the Bartels-Golub algorithm, and others like it, generate by well-behaved techniques all the quantities used at each simplex step.

In our analysis we have concentrated on stability of the whole simplex algorithm. We have shown that simplex-type algorithms cannot be well behaved on a sufficiently general class of LP problems. On the other hand, we have also shown that simplex algorithms with well-behaved updating techniques (such as the Bartels-Golub algorithm) are numerically stable provided that some reasonable tolerances are introduced into the optimality criteria.

I am indebted to G. H. Golub, A. Kiełbasiński, H. Woźniakowski, and the referee for their comments and help during the preparation of this paper.

REFERENCES

1 R. H. Bartels, A stabilization of the simplex method, *Numer. Math.* 16:414–434 (1971).
2 R. H. Bartels and G. H. Golub, The simplex method of linear programming using LU decomposition, *Comm. ACM* 12:266–268 (1969).
3 R. H. Bartels, J. Stoer, and Ch. Zenger, A realization of the simplex method based on triangular decompositions, in *Handbook for Automatic Computation: Linear Algebra* (J. H. Wilkinson and C. Reinsch Eds.), Springer, Berlin, 1971, pp. 152–190.
4 J. J. H. Forrest and J. A. Tomlin, Updating triangular factors of the basis to maintain sparsity in the product form simplex method, *Math. Programming* 2:263–278 (1972).
5 P. E. Gill, G. H. Golub, W. Murray, and M. A. Saunders, Methods for modifying matrix factorizations, *Math. Comp.* 28:505–535 (1974).
6 P. E. Gill and W. Murray, A numerically stable form of the simplex algorithm, *Linear Algebra Appl.* 7:99–138 (1973).
7 P. E. Gill, W. Murray, and M. A. Saunders, Methods for computing and modifying the LDV factors of a matrix, *Math. Comp.* 29:1051–1077 (1975).
8 D. Goldfarb and J. K. Reid, A practicable steepest-edge simplex algorithm, *Math. Programming* 12:361–371 (1977).
9 R. J. Hanson and J. A. Wisniewski, A mathematical programming updating method using modified Givens transformations and applied to LP problems, *Comm. ACM* 22(4):245–251 (1979).
10 P. M. J. Harris, Pivot selection methods of the DEVEX LP code, *Math. Programming* 5:1–28 (1973).

11 IBM Mathematical Programming System Extended—MPSX/370: Program Reference Manual, SH19-1095, IBM, 1979.
12 M. Jankowski and H. Woźniakowski, Iterative refinement implies numerical stability, *BIT* 17:303–311 (1977).
13 A. Kiełbasiński, Basic concepts in numerical error analysis (suggested by linear algebra problems), in *Mathematical Models and Numerical Methods* (A. N. Tikhonov Ed.), PWN, Warsaw, 1978, pp. 223–232.
14 B. A. Murtagh, *Advanced Linear Programming: Computation and Practice*, McGraw-Hill, New York, 1981.
15 W. Ogryczak, On Feasibility Tolerances in the Simplex Method, Inf Report 153, Warsaw Univ., 1986.
16 W. Ogryczak, On practical stopping rules for the simplex method, *Math. Programming Stud.* 31:167–174 (1987).
17 W. Orchard-Hays, *Advanced Linear-Programming Computing Techniques*, McGraw-Hill, New York, 1968.
18 J. K. Reid, A sparsity-exploiting variant of the Bartels-Golub decomposition for linear programming bases, *Math. Programming* 24:55–69 (1982).
19 S. M. Robinson, A characterization of stability in linear programming, *Oper. Res.* 25:435–447 (1977).
20 J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs, N.J., 1963.
21 H. Woźniakowski, Numerical stability of the Chebyshev method for the solution of large linear systems, *Numer. Math.* 28:191–209 (1971).

Received October 1985; final manuscript accepted 2 November 1987