

The use of Grossone in Mathematical Programming and Operations Research

Sonia De Cosmis^a, Renato De Leone^{a,*}

^a*School of Science and Technology, Università di Camerino, via Madonna delle Carceri 9, Camerino (MC) 62032 Italy*

Abstract

The concepts of infinity and infinitesimal in mathematics date back to ancient Greek and have always attracted great attention. Very recently, a new methodology has been proposed by Sergeyev [10] for performing calculations with infinite and infinitesimal quantities, by introducing an infinite unit of measure expressed by the numeral $\textcircled{1}$ (grossone). An important characteristic of this novel approach is its attention to numerical aspects. In this paper we will present some possible applications and use of $\textcircled{1}$ in Operations Research and Mathematical Programming. In particular, we will show how the use of $\textcircled{1}$ can be beneficial in anti-cycling procedure for the well-known simplex method for solving Linear Programming Problems and in defining exact differentiable Penalty Functions in Nonlinear Programming.

Keywords: Linear Programming, Simplex Method, Nonlinear Programming, Penalty Methods

2008 MSC: 90C05,

2008 MSC: 90C30

1. Introduction

A novel approach to infinite and infinitesimal numbers has been recently proposed by Sergeyev in a book and in a series of papers [10, 11, 12, 13]. By introducing a new infinite unit of measure (the numeral *grossone*, indicated

*Corresponding author

Email addresses: `sonia.decosmis@unicam.it` (Sonia De Cosmis),
`renato.deleone@unicam.it` (Renato De Leone)

by $\textcircled{1}$) as the number of elements of the set of the natural numbers, he shows that it is possible to effectively work with infinite and infinitesimal quantities and to solve many problems connected to them in the field of applied and theoretical mathematics. In this new system, there is the opportunity to treat infinite and infinitesimal numbers as particular cases of a single structure, offering a new view and alternative approaches to important aspects of mathematics such as sums of series (in particular, divergent series), limits, derivatives, etc.

The new numeral *grossone* can be introduced by describing its properties (in a similar way as done in the past with the introduction of 0 to switch from natural to integer numbers). The Infinity Unit Axiom postulate (IUA) [11, 10] is composed of three parts: Infinity, Identity, and Divisibility:

- *Infinity.* Any finite natural number n is less than grossone, i.e., $n < \textcircled{1}$.
- *Identity.* The following relationships link $\textcircled{1}$ to the identity elements 0 and 1

$$0 \cdot \textcircled{1} = \textcircled{1} \cdot 0 = 0, \quad \textcircled{1} - \textcircled{1} = 0, \quad \frac{\textcircled{1}}{\textcircled{1}} = 1, \quad \textcircled{1}^0 = 1, \quad 1^{\textcircled{1}} = 1, \quad 0^{\textcircled{1}} = 0 \quad (1)$$

- *Divisibility.* For any finite natural number n , the sets $\mathbb{N}_{k,n}, 1 \leq k \leq n$,

$$\mathbb{N}_{k,n} = k, k + n, k + 2n, k + 3n, \dots, \quad 1 \leq k \leq n, \quad \bigcup_{k=1}^n \mathbb{N}_{k,n} = \mathbb{N} \quad (2)$$

have the same number of elements indicated by $\frac{\textcircled{1}}{n}$.

The axiom above states that the infinite number $\textcircled{1}$, greater than any finite number, behaves as any natural number with the elements 0 and 1. Moreover, the quantities $\frac{\textcircled{1}}{n}$ are integers for any natural n . This axiom is added to the standard axioms of real numbers and, therefore, all standard properties (commutative, associative, existence of inverse, etc.) also apply to $\textcircled{1}$.

Sergeyev [12, 13] also defines a new way to express the infinite and infinitesimal numbers using a register similar to traditional positional number system, but with base number $\textcircled{1}$. A number \mathbf{C} in this new system can be constructed by subdividing it into groups corresponding to powers of $\textcircled{1}$ and has the following representation:

$$\mathbf{C} = c_{p_m} \textcircled{1}^{p_m} + \dots + c_{p_1} \textcircled{1}^{p_1} + c_{p_0} \textcircled{1}^{p_0} + c_{p_{-1}} \textcircled{1}^{p_{-1}} + \dots + c_{p_{-k}} \textcircled{1}^{p_{-k}}. \quad (3)$$

where the quantities c_i (the *grossdigits*) and p_i (the *grosspowers*) are expressed by the traditional numerical system for representing finite numbers (for example, floating point numbers). The grosspowers are sorted in descending order:

$$p_m > p_{m-1} > \dots > p_1 > p_0 > p_{-1} > \dots p_{-(k-1)} > p_{-k}$$

with $p_0 = 0$.

In this new numeral system, finite numbers are represented by numerals with only one grosspower $p_0 = 0$. Infinitesimal numbers are represented by numeral \mathbf{C} having only negative finite or infinite grosspowers. The simplest infinitesimal number is $\mathbf{1}^{-1}$ for which

$$\mathbf{1}^{-1}\mathbf{1} = \mathbf{1} \mathbf{1}^{-1} = 1. \tag{4}$$

We note that infinitesimal numbers are not equal to zero. In particular, $\frac{1}{\mathbf{1}} > 0$. Infinite numbers are expressed by numerals having at least one finite or infinite grosspower greater than zero.

A peculiar characteristic of the newly proposed numeral system is its attention to its numerical aspects and to applications. The Infinity Computer proposed by Sergeyev is able to execute computations with infinite, finite, and infinitesimal numbers numerically (not symbolically) in a novel framework.

In this paper we will present two possible uses of this numeral system in Mathematical Programming and Operations Research. In particular, in Section 2 we will show a simple way to implement anti-cycling strategies in the simplex method for solving Linear Programming problems. Various anti-cycling procedures have been proposed and implemented in state-of-the-art softwares. The lexicographic strategies has received particular attention since it allows, in contrast to Bland's rule, complete freedom in choosing, at each iteration, the entering variable. In Section 3 we revert our attention to Nonlinear Programming problems and, in particular, to differentiable penalty functions. In the new numeral system it is possible to define an exact, differentiable penalty function and we will show that stationary points of this penalty function are KKT points for the original Nonlinear Programming problem. Two simple examples are also provided showing the effectiveness of this approach. Conclusions and indications for further applications of $\mathbf{1}$ in Mathematical Programming are reported in Section 4.

We briefly describe our notation now. All vectors are column vectors and will be indicated with lower case Latin letter (x, z, \dots). Subscripts indicate

components of a vector, while superscripts are used to identify different vectors. Matrices will be indicated with upper case roman letter (A, B, \dots). If $A \in \mathbb{R}^{m \times n}$, $A_{.j}$ is the j -th column of A ; if $B \subseteq \{1, \dots, n\}$, $A_{.B}$ is the submatrix of A composed by all columns $A_{.j}$ such that $j \in B$. The set of real numbers and the set of nonnegative real numbers will be denoted by \mathbb{R} and \mathbb{R}_+ respectively. The rank of a matrix A will be indicated by $\text{rank } A$. The space of the n -dimensional vectors with real components will be indicated by \mathbb{R}^n and \mathbb{R}_+^n is an abbreviation for the nonnegative orthant in \mathbb{R}^n . The symbol $\|x\|$ indicates the Euclidean norm of a vector x . Superscript T indicates transpose. The scalar product of two vectors x and y in \mathbb{R}^n will be denoted by $x^T y$. Here and throughout the symbols $:=$ and $=:$ denote definition of the term on the left and the right sides of each symbol, respectively. The gradient $\nabla f(x)$ of a continuously differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ at a point $x \in \mathbb{R}^n$ is assumed to be a column vector. If $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a continuously differentiable vector-valued function, then $\nabla F(x)$ denotes the Jacobian matrix of F at $x \in \mathbb{R}^n$.

2. Lexicographic rule and grossone

The simplex method, originally proposed by G.B. Dantzig [4] more than half a century ago, is still today one of the most used algorithms for solving Linear Programming problems. Finite termination of the method can only be guaranteed if special techniques are employed to eliminate cycling. In this section we will show how in the new numeral system it is very simple to implement such anti-cycling rules.

Given a matrix $A \in \mathbb{R}^{m \times n}$ with $\text{rank } A = m$ and vectors $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$, the Linear Programming problem in standard form can be stated as follows

$$\begin{aligned} \min_x \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & x \geq 0. \end{aligned} \tag{5}$$

The simplex algorithm moves from a Basic Feasible Solution (BFS) to an adjacent Basic Feasible Solution until an optimal solution is reached or unboundness of the problem is detected.

More precisely, a submatrix $A_B \in \mathbb{R}^{m \times m}$ is a basis matrix if it is non-singular; a point $\bar{x} \in \mathbb{R}^n$ is a BFS if it is feasible and the columns of A corresponding to positive components of \bar{x} are linearly independent. Basic

Feasible Solutions correspond to vertices of the feasible region. A vertex \bar{x} is non-degenerate if exactly m components of \bar{x} are positive: in this case there is a single Basis matrix associated to the point. A vertex \bar{x} is degenerate if fewer than m components are strictly positive: in this case more than one basis matrix can be associated to the point.

Finding an initial Basic Feasible Solution, if it exists, requires to solve (always using the Simplex Method) an auxiliary problem, whose initial solution is trivially obtained adding artificial variables.

A single iteration of the (primal) simplex method requires the following steps:

Step 0 Let $B \subseteq \{1, \dots, n\}$ be the *current* base and let $x \in X$ the *current* BFS

$$x_B = A_B^{-1}b \geq 0, \quad x_N = 0, \quad |B| = m.$$

Assume

$$B = \{j_1, j_2, \dots, j_m\}$$

and

$$N = \{1, \dots, n\} \setminus B = \{j_{m+1}, \dots, j_n\}.$$

Step 1 Compute

$$\pi = A_B^{-T} c_B$$

and the reduced cost vector

$$\bar{c}_{j_k} = c_{j_k} - A_{.j_k}^T \pi, \quad k = m + 1, \dots, n.$$

Step 2 If

$$\bar{c}_{j_k} \geq 0, \quad \forall k = m + 1, \dots, n$$

the current point is an optimal BFS and the algorithm stops.

Instead, if $\bar{c}_N \not\geq 0$, choose j_r with $r \in \{m + 1, \dots, n\}$ such that $\bar{c}_{j_r} < 0$. This is the variable candidate to enter the base.

Step 3 Compute

$$\bar{A}_{.j_r} = A_B^{-1} A_{.j_r}$$

Step 4 If

$$\bar{A}_{.j_r} \leq 0$$

the problem is unbounded below and the algorithm stops.
Otherwise, compute

$$\bar{\rho} = \min_{i: \bar{A}_{ij_r} > 0} \left\{ \frac{(A_{.B}^{-1}b)_i}{\bar{A}_{ij_r}} \right\}$$

and let $s \in \{1, \dots, m\}$ such that $\left\{ \frac{(A_{.B}^{-1}b)_s}{\bar{A}_{sj_r}} \right\} = \bar{\rho}$; j_s is the leaving variable.

Step 5 Define

$$\begin{aligned} \bar{x}_{j_k} &= 0, & k &= m+1, \dots, n, k \neq r \\ \bar{x}_{j_r} &= \bar{\rho} \\ \bar{x}_B(\rho) &= A_{.B}^{-1}b - \bar{\rho}\bar{A}_{.j_r}. \end{aligned}$$

and

$$\bar{B} = B \setminus \{j_s\} \cup \{j_r\} = \{j_1, j_2, \dots, j_{s-1}, j_r, j_{s+1}, \dots, j_m\}$$

Note that, when a non-degenerate step is performed, the objective function value strictly decreases. Therefore, the Simplex Method will terminate after a finite number of steps if all the BFS are non-degenerate. In case of degenerate BFS, the objective function value may remain the same for a number of steps and the algorithm will cycle. Therefore, specific anti-cycling rules must be implemented to avoid this negative feature.

Among the various anti-cycling criteria, the lexicographic pivoting rule [5, 15] has received special attention since, in contrast to other rules such as Bland's rule, there is a complete freedom in choosing the entering variable.

The lexicographic simplex method requires, at each iteration, to choose the leaving variable using a specific procedure (the lexicographic rule) [15].

Let B_0 be the initial base and $N_0 = \{1, \dots, n\} \setminus B_0$. We can always assume, after columns reordering, that A has the form

$$A = \begin{bmatrix} A_{.B_0} & \vdots & A_{.N_0} \end{bmatrix}$$

Let

$$\bar{\rho} = \min_{i: \bar{A}_{ij_r} > 0} \frac{(A_{\cdot B}^{-1}b)_i}{\bar{A}_{ij_r}}$$

if such minimum value is reached in only one index, this is the leaving variable. Otherwise, let

$$\mathcal{I}_1 := \left\{ i \in \{1, \dots, m\} : \bar{A}_{ij_r} > 0 \text{ and } \frac{(A_{\cdot B}^{-1}b)_i}{\bar{A}_{ij_r}} = \bar{\rho} \right\}$$

and

$$\bar{\rho}_1 := \min_{i \in \mathcal{I}_1} \frac{(A_{\cdot B}^{-1}A_{\cdot B_0})_{i1}}{\bar{A}_{ij_r}}.$$

and choose the index $i_1 \in \mathcal{I}_1$ getting the minimum, that is the index i_1 such that

$$\frac{(A_{\cdot B}^{-1}A_{\cdot B_0})_{i_1 1}}{\bar{A}_{i_1 j_r}} = \bar{\rho}_1.$$

If the minimum is reached by only one index i_1 , then this is the leaving variable. Otherwise, let

$$\mathcal{I}_2 := \left\{ i \in \mathcal{I}_1 : \frac{(A_{\cdot B}^{-1}A_{\cdot B_0})_{i1}}{\bar{A}_{ij_r}} = \bar{\rho}_1 \right\} \text{ and } \bar{\rho}_2 := \min_{i \in \mathcal{I}_2} \frac{(A_{\cdot B}^{-1}A_{\cdot B_0})_{i2}}{\bar{A}_{ij_r}}$$

and choose the index $i_2 \in \mathcal{I}_2$ getting the minimum, that is the index i_2 such that

$$\frac{(A_{\cdot B}^{-1}A_{\cdot B_0})_{i_2 2}}{\bar{A}_{i_2 j_r}} = \bar{\rho}_2.$$

This procedure will terminate providing a single index since the rows of the matrix $(A_{\cdot B}^{-1}A_{\cdot B_0})$ are linearly independent. The finiteness of the lexicographic simplex method follows from the simple observation that the vector whose first element is the current objective function value and the other components are the reduced costs, strictly lexicographically decreases at each iteration.

The procedure outlined above is equivalent to perturb each component of the RHS vector b by a very small quantity [2].

If this perturbation is small enough, the new Linear Programming problem is nondegenerate and the simplex method produces exactly the same pivot sequence as the lexicographic pivot rule.

However, is very difficult to determine how small this perturbation must be. More often a symbolic perturbation is used (with higher computational costs).

In the new numeral system obtained by the introduction of $\mathbb{1}$ we propose to replace b_i by

$$\tilde{b}_i = b_i + \sum_{j \in B_0} A_{ij} \mathbb{1}^{-j}. \quad (6)$$

More specifically, let

$$e = \begin{bmatrix} \mathbb{1}^{-1} \\ \mathbb{1}^{-2} \\ \vdots \\ \mathbb{1}^{-m} \end{bmatrix}$$

and define

$$\tilde{b} = A_{\cdot B}^{-1}(b + A_{\cdot B_0} e) = A_{\cdot B}^{-1} b + A_{\cdot B}^{-1} A_{\cdot B_0} e. \quad (7)$$

Therefore, $\tilde{b}_i = (A_{\cdot B}^{-1} b)_i + \sum_{k=1}^m (A_{\cdot B}^{-1} A_{\cdot B_0})_{ik} \mathbb{1}^{-k}$ and

$$\begin{aligned} & \min_{i: \bar{A}_{ij_r} > 0} \frac{(A_{\cdot B}^{-1} b)_i + \sum_{k=1}^m (A_{\cdot B}^{-1} A_{\cdot B_0})_{ik} \mathbb{1}^{-k}}{\bar{A}_{ij_r}} \\ &= \min_{i: \bar{A}_{ij_r} > 0} \frac{(A_{\cdot B}^{-1} b)_i}{\bar{A}_{ij_r}} + \frac{(A_{\cdot B}^{-1} A_{\cdot B_0})_{i1}}{\bar{A}_{ij_r}} \mathbb{1}^{-1} + \dots + \frac{(A_{\cdot B}^{-1} A_{\cdot B_0})_{im}}{\bar{A}_{ij_r}} \mathbb{1}^{-m} \end{aligned} \quad (8)$$

Taking into account the properties of the power of $\mathbb{1}$, the index i that will be chosen by the formula (8) will be exactly the same obtained by the lexicographic pivoting rule outlined before.

3. Nonlinear programming

Nonlinear constrained optimization problems are an important class of problems with a broad range of engineering, scientific, and operational applications. The problem can be stated as follows:

$$\begin{aligned} & \min_x && f(x) \\ & \text{subject to} && g(x) \leq 0 \\ & && h(x) = 0 \end{aligned} \quad (9)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$, and $h : \mathbb{R}^n \rightarrow \mathbb{R}^p$. For simplicity we will assume that the functions f , g , and h are twice continuously differentiable. The feasible set will be indicated by X . In this section we will show how, using the new numeral system, it is possible to introduce exact differentiable penalty functions and we will discuss the relationship between stationary point of the penalty function and KKT points of the original constrained problem. After introducing some basic concepts and definitions, we consider in Subsection 3.1 the equality constrained case and later, in Subsection 3.2, the most general case of equality and inequality constraints.

Definition 1. Given a point $x^0 \in X$, the set of active constraints at x^0 is

$$I(x^0) = \{1, \dots, p\} \cup \{i \in \{1, \dots, m\} : g_i(x^0) = 0\}.$$

Definition 2. The linear independence constraint qualification (LICQ) condition is said to hold true at $x^0 \in X$ if the set of gradients of the active constraints at x^0 is linearly independent.

The associated Lagrangian function $L(x, \mu, \pi)$ is :

$$L(x, \pi, \mu) := f(x) + \mu^T g(x) + \pi^T h(x) \quad (10)$$

where $\pi \in \mathbb{R}^p$ and $\mu \in \mathbb{R}^m$ are the multipliers associated to the equality and inequality constraints respectively.

Definition 3. A triplet (x^*, π^*, μ^*) is a Karush-Kuhn-Tucker (KKT) point if

$$\begin{aligned} \nabla_x L(x^*, \mu^*, \pi^*) &= \nabla f(x^*) + \sum_{i=1}^m \nabla g_i(x^*) \mu_i^* + \sum_{j=1}^p \nabla h_j(x^*) \pi_j^* = 0 \\ \nabla_\mu L(x^*, \mu^*, \pi^*) &= g(x^*) \leq 0 \\ \nabla_\pi L(x^*, \mu^*, \pi^*) &= h(x^*) = 0 \\ \mu^* &\geq 0 \\ \mu^{*T} \nabla_\mu L(x^*, \mu^*, \pi^*) &= 0 \end{aligned}$$

The following theorem [9] states the first order necessary optimality conditions for the nonlinear optimization problems.

Theorem 3.1. *Consider the nonlinear optimization problem (9). Let $x^* \in X$ be a local minimum of Problem (9), and assume that at x^* the LICQ¹ condition holds true. Then, there exist vectors μ^* and π^* such that the triplet (x^*, π^*, μ^*) is a Karush-Kuhn-Tucker point.*

Different algorithms have been proposed and studied for solving the general nonlinear optimization problems. We refer the interested reader to the classical book of Fletcher [7] and to the more recent book of Boyd and Vandenberghe [1] for a general discussion of Nonlinear Programming algorithms. Here we will concentrate our attention on exact penalty functions and we will show how the use of $\textcircled{1}$ can be beneficial in defining exact differentiable penalty functions.

Let $\phi : x \in \mathbb{R}^n \rightarrow \mathbb{R}_+$ be a continuously differentiable function such that

$$\phi(x) \begin{cases} = 0 & \text{if } x \in X \\ > 0 & \text{otherwise} \end{cases}$$

A possible choice for the function $\phi(x)$ is

$$\phi(x) = \sum_{i=1}^m \max\{g_i(x), 0\}^2 + \sum_{j=1}^k h_j(x)^2$$

Note that this function is continuously differentiable but not twice differentiable.

From the function $\phi(x)$, the exterior penalty function

$$P(x, \epsilon) = f(x) + \frac{1}{2\epsilon}\phi(x)$$

can be constructed and the following unconstrained optimization problem can be defined:

$$\min_x P(x, \epsilon). \tag{11}$$

It can be shown that there is no finite value of the penalty parameter ϵ for which, by solving Problem (11), a solution of the original Problem (9) is obtained. Sequential penalty methods require to solve a sequence of minimization problems (similar to Problem (11)) for decreasing values of

¹Weaker Constraint Qualification conditions can be imposed. See [14] for a review of different Constraints Qualification conditions and the relationship among them.

the parameter ϵ . It is possible to construct exact non-differentiable penalty functions [7] as well as differentiable exact penalty function but, in this case, it is necessary to introduce terms related to first order optimality conditions [6], thus making the penalty function much more complicate.

We propose here to substitute the term $1/\epsilon$ with $\mathbf{1}$. In the next subsection we will present convergence results for the simpler case of equality constraints, while Subsection 3.2 will present results and a simple example of exact penalty function for the more general case.

3.1. The equality constraint case.

Consider the optimization problem with equality constraints:

$$\begin{aligned} \min_x \quad & f(x) \\ \text{subject to} \quad & h(x) = 0 \end{aligned} \tag{12}$$

where f and h are defined as before.

The following theorem [7, Theorem 12.1.1] states the convergence for the sequential penalty method.

Theorem 3.2. *Let $f(x)$ be bounded below in the (nonempty) feasible region, and let $\{\epsilon_k\}$ be a monotonic non-increasing sequence such that $\{\epsilon_k\} \downarrow 0$, and assume that for each k there exists a global minimum $x(\epsilon_k)$ of $P(x; \epsilon_k)$. Then*

- (i) $\{P(x(\epsilon_k); \epsilon_k)\}$ is monotonically non-decreasing
- (ii) $\{\phi(x(\epsilon_k))\}$ is monotonically non-increasing
- (iii) $f(x(\epsilon_k))$ is monotonically non-decreasing

Moreover, $\lim_k h(x(\epsilon_k)) = 0$ and each limit point x^* of the sequence $\{x(\epsilon_k)\}_k$ solves Problem (12).

In order to derive first order optimality condition, we will make the following assumptions on the functions $f(x)$ and $h(x)$, on the gradient $\nabla f(x)$ and on the Jacobian $\nabla h(x)$:

Assumption 1. *If*

$$x = x^0 + \mathbf{1}^{-1}x^1 + \mathbf{1}^{-2}x^2 + \dots$$

with $x^i \in \mathbb{R}^n$, then

$$\begin{aligned} f(x) &= f(x^0) + \mathbb{1}^{-1}f^{(1)}(x) + \mathbb{1}^{-2}f^{(2)}(x) + \dots \\ h(x) &= h(x^0) + \mathbb{1}^{-1}h^{(1)}(x) + \mathbb{1}^{-2}h^{(2)}(x) + \dots \\ \nabla f(x) &= \nabla f(x^0) + \mathbb{1}^{-1}F^{(1)}(x) + \mathbb{1}^{-2}F^{(2)}(x) + \dots \\ \nabla h(x) &= \nabla h(x^0) + \mathbb{1}^{-1}H^{(1)}(x) + \mathbb{1}^{-2}H^{(2)}(x) + \dots \end{aligned}$$

where $f^{(i)} : \mathbb{R}^n \rightarrow \mathbb{R}$, $h^{(i)} : \mathbb{R}^n \rightarrow \mathbb{R}^p$, $F^{(i)} : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $H^{(i)} : \mathbb{R}^n \rightarrow \mathbb{R}^{p \times n}$ are all finite-value functions.

In the sequel, we will always suppose that the assumption above holds. Note that Assumption 1 is satisfied (for example) by functions that are product of polynomial functions in a single variable, i.e.,

$$p(x) = p_1(x_1)p_2(x_2) \cdots p_n(x_n)$$

where $p_i(x_i)$ is a polynomial function.

We are now ready to state the convergence theorem for the exact penalty function using $\mathbb{1}$.

Theorem 3.3. *Consider Problem (12) and define the following unconstrained problem*

$$\min_x f(x) + \frac{\mathbb{1}}{2} \|h(x)\|^2. \quad (13)$$

Let

$$x^* = x^{*0} + \mathbb{1}^{-1}x^{*1} + \mathbb{1}^{-2}x^{*2} + \dots$$

be a stationary point for (13) and assume that the LICQ condition holds true at x^{*0} . Then, the pair $(x^{*0}, \pi^* = h^{(1)}(x^*))$ is a KKT point of (12).

Proof. Since $x^* = x^{*0} + \mathbb{1}^{-1}x^{*1} + \mathbb{1}^{-2}x^{*2} + \dots$ is a stationary point we have that

$$\nabla f(x^*) + \mathbb{1} \sum_{j=1}^p \nabla h_j(x^*) h_j(x^*) = 0$$

Therefore, from Assumption 1

$$\begin{aligned} & \nabla f(x^{*0}) + \mathbb{1}^{-1}F^{(1)}(x^*) + \mathbb{1}^{-2}F^{(2)}(x^*) + \dots + \\ & + \mathbb{1} \sum_{j=1}^p \left[\left(\nabla h_j(x^{*0}) + \mathbb{1}^{-1}H_j^{(1)}(x^*) + \mathbb{1}^{-2}H_j^{(2)}(x^*) + \dots \right) \right. \\ & \quad \left. \left(h_j(x^{*0}) + \mathbb{1}^{-1}h_j^{(1)}(x^*) + \mathbb{1}^{-2}h_j^{(2)}(x^*) + \dots \right) \right] = 0 \end{aligned}$$

and rearranging the terms we obtain:

$$\begin{aligned} & \mathbb{1} \left(\sum_{j=1}^p \nabla h_j(x^{*0}) h_j(x^{*0}) \right) + \\ & + \left(\nabla f(x^{*0}) + \sum_{j=1}^p \nabla h_j(x^{*0}) h_j^{(1)}(x^*) + \sum_{j=1}^p H_j^{(1)}(x^*) h_j(x^{*0}) \right) + \\ & \quad + \mathbb{1}^{-1} \left(\dots \right) + \mathbb{1}^{-2} \left(\dots \right) + \dots = 0 \end{aligned}$$

Hence,

$$\sum_{j=1}^p \nabla h_j(x^{*0}) h_j(x^{*0}) = 0$$

and, from the LICQ condition, it follows that

$$h(x^{*0}) = 0. \tag{14}$$

Therefore, the point x^{*0} is feasible for (12). Moreover, from

$$\nabla f(x^{*0}) + \sum_{j=1}^p \nabla h_j(x^{*0}) h_j^{(1)}(x^*) + \sum_{j=1}^p H_j^{(1)}(x^*) h_j(x^{*0}) = 0$$

and (14) above, it follows that

$$\nabla f(x^{*0}) + \sum_{j=1}^k \nabla h_j(x^{*0}) h_j^{(1)}(x^*) = 0 \tag{15}$$

that shows that $(x^{*0}, \pi^* = h^{(1)}(x^*))$ is a KKT point of (12). \square

3.1.1. A simple example

Consider the following simple 2-dimensional optimization problem with a single linear constraint²:

$$\begin{aligned} \min_x \quad & \frac{1}{2}x_1^2 + \frac{1}{6}x_2^2 \\ \text{subject to} \quad & x_1 + x_2 = 1 \end{aligned} \quad (16)$$

The pair $(\bar{x}, \bar{\pi})$ with $\bar{x} = \begin{bmatrix} 1/4 \\ 3/4 \end{bmatrix}$, $\bar{\pi} = -\frac{1}{4}$ is a KKT point. The corresponding unconstrained optimization problem is

$$\min_x \frac{1}{2}x_1^2 + \frac{1}{6}x_2^2 - \frac{1}{2}\mathbb{1}(1 - x_1 - x_2)^2 \quad (17)$$

and the first Order Optimality Conditions are

$$\begin{cases} x_1 - \mathbb{1}(1 - x_1 - x_2) = 0 \\ \frac{1}{3}x_2 - \mathbb{1}(1 - x_1 - x_2) = 0 \end{cases}$$

Therefore, the point

$$x_1^* = \frac{1\mathbb{1}}{1 + 4\mathbb{1}}, \quad x_2^* = \frac{3\mathbb{1}}{1 + 4\mathbb{1}}$$

is a stationary point of Problem (17). Note that

$$x_1^* = \frac{1}{4} - \mathbb{1}^{-1} \left(\frac{1}{16} - \frac{1}{64}\mathbb{1}^{-1} \dots \right)$$

$$x_2^* = \frac{3}{4} - \mathbb{1}^{-1} \left(\frac{3}{16} - \frac{3}{64}\mathbb{1}^{-1} \dots \right)$$

and $x_1^{*0} = \frac{1}{4} = \bar{x}_1$ and $x_2^{*0} = \frac{3}{4} = \bar{x}_2$. Moreover,

$$\begin{aligned} -\mathbb{1}(1 - x_1^* - x_2^*) &= -\mathbb{1} \left(1 - \frac{1}{4} + \mathbb{1}^{-1} \frac{1}{16} - \frac{1}{64}\mathbb{1}^{-2} \dots \right. \\ &\quad \left. - \frac{3}{4} + \mathbb{1}^{-1} \frac{3}{16} - \frac{3}{64}\mathbb{1}^{-2} \dots \right) \\ &= -\mathbb{1} \left(\mathbb{1}^{-1} \frac{1}{16} + \mathbb{1}^{-1} \frac{3}{16} - \mathbb{1}^{-2} \frac{4}{64} \dots \right) \\ &= -\frac{1}{4} + \frac{4}{64}\mathbb{1}^{-1} \dots \end{aligned}$$

and $h^{(1)}(x^*) = -\frac{1}{4} = \bar{\pi}$

²The example is taken from [8]

3.2. The general constrained case.

In this subsection we return back to the most general nonlinear optimization problem (9) with equality and inequality constraints and we introduce the following corresponding unconstrained optimization problem:

$$\min_x f(x) + \frac{1}{2} \mathbb{1} \|\max\{0, g_i(x)\}\|^2 + \frac{1}{2} \mathbb{1} \|h(x)\|^2 \quad (18)$$

In addition to the conditions stated in Assumption 1 we similarly require that, under the same hypothesis on x , the following conditions hold true:

$$\begin{aligned} g(x) &= g(x^0) + \mathbb{1}^{-1} g^{(1)}(x) + \mathbb{1}^{-2} g^{(2)}(x) + \dots \\ \nabla g(x) &= \nabla g(x^0) + \mathbb{1}^{-1} G^{(1)}(x) + \mathbb{1}^{-2} G^{(2)}(x) + \dots \end{aligned}$$

In order to derive optimality conditions we need a modified version of the LICQ condition.

Definition 4. Let $x^0 \in \mathbb{R}^n$. The Modified LICQ (MLICQ) condition is said to hold true at x^0 if the vectors $\left\{ \nabla g_i(x^0)_{i:g_i(x^0) \geq 0}, \nabla h_j(x^0)_{j=1, \dots, k} \right\}$ are linearly independent.

The following theorem shows the relationship between stationary points of Problem (18) and KKT points of the general optimization Problem (9).

Theorem 3.4. Consider the Problem (9) and the corresponding unconstrained Problem (18). Let

$$x^* = x^{*0} + \mathbb{1}^{-1} x^{*1} + \mathbb{1}^{-2} x^{*2} + \dots$$

be a stationary point for (18) and assume that MLICQ condition holds true at x^{*0} . Then, the triplet $\left(x^{*0}, \mu^*, \pi^* = h^{(1)}(x^*) \right)$ is a KKT point of (9) where

$$\mu_i^* = \begin{cases} 0 & \text{if } g_i(x^{*0}) < 0 \\ \max\{0, g_i^{(1)}(x^*)\} & \text{if } g_i(x^{*0}) = 0 \end{cases}$$

Proof. Since $x^* = x^{*0} + \mathbb{1}^{-1} x^{*1} + \mathbb{1}^{-2} x^{*2} + \dots$ is a stationary point we have

$$\nabla f(x^*) + \mathbb{1} \sum_{i=1}^m \nabla g_i(x^*) \max\{0, g_i(x^*)\} + \mathbb{1} \sum_{j=1}^p \nabla h_j(x^*) h_j(x^*) = 0.$$

From Assumption 1 we obtain

$$\begin{aligned}
& \nabla f(x^{*0}) + \mathbb{1}^{-1}F^{(1)}(x^*) + \mathbb{1}^{-2}F^{(2)}(x^*) + \dots + \\
& + \mathbb{1} \sum_{i=1}^m \left[\left(\nabla g_i(x^{*0}) + \mathbb{1}^{-1}G_i^{(1)}(x^*) + \mathbb{1}^{-2}G_i^{(2)}(x^*) + \dots \right) \right. \\
& \quad \left. \left(\max \left\{ 0, g_i(x^{*0}) + \mathbb{1}^{-1}g_i^{(1)}(x^*) + \mathbb{1}^{-2}g_i^{(2)}(x^*) + \dots \right\} \right) \right] + \\
& + \mathbb{1} \sum_{j=1}^p \left[\left(\nabla h_j(x^{*0}) + \mathbb{1}^{-1}H_j^{(1)}(x^*) + \mathbb{1}^{-2}H_j^{(2)}(x^*) + \dots \right) \right. \\
& \quad \left. \left(h_j(x^{*0}) + \mathbb{1}^{-1}h_j^{(1)}(x^*) + \mathbb{1}^{-2}h_j^{(2)}(x^*) + \dots \right) \right] = 0
\end{aligned}$$

Note that the following implications hold:

$$\begin{aligned}
g_i(x^{*0}) > 0 & \Rightarrow \max \{0, g_i(x^*)\} = g_i(x^{*0}) + \mathbb{1}^{-1}g_i^{(1)}(x^*) + \mathbb{1}^{-2}g_i^{(2)}(x^*) \dots \\
g_i(x^{*0}) < 0 & \Rightarrow \max \{0, g_i(x^*)\} = 0 \\
g_i(x^{*0}) = 0 & \Rightarrow \max \{0, g_i(x^*)\} = \mathbb{1}^{-1} \max \left\{ 0, g_i^{(1)}(x^*) + \mathbb{1}^{-1}g_i^{(2)}(x^*) + \dots \right\}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
& \nabla f(x^{*0}) + \mathbb{1}^{-1}F^{(1)}(x^*) + \mathbb{1}^{-2}F^{(2)}(x^*) + \dots + \\
& + \mathbb{1} \sum_{\substack{i=1 \\ g_i(x^{*0}) > 0}}^m \left[\left(\nabla g_i(x^{*0}) + \mathbb{1}^{-1}G_i^{(1)}(x^*) + \mathbb{1}^{-2}G_i^{(2)}(x^*) + \dots \right) \right. \\
& \quad \left. \left(g_i(x^{*0}) + \mathbb{1}^{-1}g_i^{(1)}(x^*) + \mathbb{1}^{-2}g_i^{(2)}(x^*) + \dots \right) \right] + \\
& + \mathbb{1} \sum_{\substack{i=1 \\ g_i(x^{*0}) = 0}}^m \left[\left(\nabla g_i(x^{*0}) + \mathbb{1}^{-1}G_i^{(1)}(x^*) + \mathbb{1}^{-2}G_i^{(2)}(x^*) + \dots \right) \right. \\
& \quad \left. \left(\mathbb{1}^{-1} \max \left\{ 0, g_i^{(1)}(x^*) + \mathbb{1}^{-1}g_i^{(2)}(x^*) + \dots \right\} \right) \right] + \\
& + \mathbb{1} \sum_{j=1}^p \left[\left(\nabla h_j(x^{*0}) + \mathbb{1}^{-1}H_j^{(1)}(x^*) + \mathbb{1}^{-2}H_j^{(2)}(x^*) + \dots \right) \right. \\
& \quad \left. \left(h_j(x^{*0}) + \mathbb{1}^{-1}h_j^{(1)}(x^*) + \mathbb{1}^{-2}h_j^{(2)}(x^*) + \dots \right) \right] = 0
\end{aligned}$$

Similarly to the equality constraint case, after rearranging terms we obtain:

$$\sum_{\substack{i=1 \\ g_i(x^{*0}) \geq 0}}^m \nabla g_i(x^{*0})g_i(x^{*0}) + \sum_{j=1}^p \nabla h_j(x^{*0})h_j(x^{*0}) = 0$$

and, since MLICQ holds at x^{*0} , $g(x^{*0}) \leq 0$ and $h(x^{*0}) = 0$. Moreover, taking into account that $g(x^{*0}) \leq 0$, we obtain that

$$\nabla f(x^{*0}) + \sum_{\substack{i=1 \\ g_i(x^{*0})=0}}^m \nabla g_i(x^{*0})\max\{0, g_i^{(1)}(x^*)\} + \sum_{j=1}^p \nabla h_j(x^{*0})h_j^{(1)}(x^*) = 0$$

that completes the proof. \square

It is worth to note that, when $g_i(x^{*0}) = 0$, it is still possible that $g_i^{(1)}(x^*) + \textcircled{1}^{-1}g_i^{(2)}(x^*) + \dots > 0$ and, therefore, the quantity $\sum_{\substack{i=1 \\ g_i(x^{*0})=0}}^m \nabla g_i(x^{*0})g_i^{(1)}(x^*)$ is

present in the last equation of the proof.

3.2.1. A second example

Also in this case we present a very simple example to clarify the previous result. Consider the minimization problem

$$\begin{aligned} \min_{x \in \mathbb{R}} \quad & x \\ \text{subject to} \quad & x \geq 1 \end{aligned} \tag{19}$$

(i.e., $g(x) = 1 - x$) for which the optimal solution is $\bar{x} = 1$ with associated multiplier $\bar{\mu} = 1$. The corresponding unconstrained optimization problem is

$$\min_{x \in \mathbb{R}} x + \frac{\textcircled{1}}{2} \max\{0, 1 - x\}^2. \tag{20}$$

The first order optimality condition is

$$1 - \textcircled{1} \max\{0, 1 - x\} = 0.$$

The above equation has no solution when $1 - x \leq 0$ and for $x < 1$ the only solution is

$$x^* = \frac{\textcircled{1} - 1}{\textcircled{1}} = 1 - \textcircled{1}^{-1}$$

Therefore, as expected, $x^{*0} = 1$. Moreover, $g(x^*) = 1 - (1 - \textcircled{1}^{-1}) = \textcircled{1}^{-1}$ and $\mu^* = 1$.

4. Conclusions

In this paper we have presented possible uses of $\textcircled{1}$ in Linear and Nonlinear Programming. In the new numerical system, making full use of $\textcircled{1}$, it is possible to implement, in a very simple way, anti-cycling strategies for the simplex method. Moreover, we have shown that exact, differentiable penalty methods can be constructed for general nonlinear programming. These are not the only possible applications of $\textcircled{1}$ in Operations Research and Mathematical Programming. Another interesting application is in Data Envelopment Analysis (DEA) methodology. In the basic version proposed by Charnes, Cooper and Rhodes [3] the use of a infinitesimal non-archimedean quantity ϵ is proposed. Negative power of $\textcircled{1}$ will allow to achieve the same theoretical results and, thus, the efficiency of a single Decision Making Unit (DMU) can be easily obtained by solving a single Linear Programming problems using the new arithmetic based on $\textcircled{1}$.

Acknowledges

The authors would like to thank Prof. Gaetano Zanghirati and Giada Galeotti for their useful suggestions on the definition of the multipliers for the general nonlinear case.

References

- [1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [2] A. Charnes. Optimality and degeneracy in linear programming. *Econometrica*, 20(2):160–170, 1952.
- [3] A. Charnes, W.W. Cooper, and E. Rhodes. Measuring the efficiency of decision-making units. *European Journal of Operational Research*, 2:429–444, 1978.
- [4] G.B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.
- [5] George B. Dantzig, Alex Orden, and Philip Wolfe. The generalized simplex method for minimizing a linear form under linear inequality restraints. *Pacific Journal of Mathematics*, 5(2):183–195, 1955.

- [6] G. Di Pillo and L. Grippo. An exact penalty method with global convergence properties for nonlinear programming problems. *Mathematical Programming*, 36:1–18, 1986.
- [7] R. Fletcher. *Practical Methods of Optimization, Second Edition*. John Wiley & Sons, 1990.
- [8] Giampaolo Liuzzi. Metodi di ottimizzazione vincolata. Lecture Notes, <http://www.iasi.cnr.it/liuzzi/corsodottorato2004/Vincolato.pdf>, 2007.
- [9] O.L. Mangasarian. *Nonlinear Programming*. McGraw–Hill, New York, 1969.
- [10] Yaroslav D. Sergeyev. *Arithmetic of Infinity*. Edizioni Orizzonti Meridionali, CS, 2003.
- [11] Yaroslav D. Sergeyev. A new applied approach for executing computations with infinite and infinitesimal quantities. *Informatica*, 19(4):567–596, 2008.
- [12] Yaroslav D. Sergeyev. Numerical computations and mathematical modelling with infinite and infinitesimal numbers. *Journal of Applied Mathematics and Computing*, 29:177195, 2009.
- [13] Yaroslav D. Sergeyev. Numerical point of view on calculus for functions assuming finite, infinite, and infinitesimal values over finite, infinite, and infinitesimal domains. *Nonlinear Analysis Series A: Theory, Methods & Applications*, 71(12):e1688e1707, 2009.
- [14] M.V. Solodov. Constraint qualifications. In J. Cochran, J, editor, *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Inc, 2010. to appear.
- [15] Tamas Terlaky. Lexicographic pivoting rules. In C.A. Floudas and P.M. Pardalos, editors, *Encyclopedia of Optimization*, volume 6, pages 173–177. Kluwer Academic Press, 2009.