P. Armand · D. Orban

The Squared Slacks Transformation in Nonlinear Programming

August 29, 2007

Abstract. We recall the use of squared slacks used to transform inequality constraints into equalities and several reasons why their introduction may be harmful in many algorithmic frameworks routinely used in nonlinear programming. Numerical examples performed with the sequential quadratic programming method illustrate those reasons.

1. Introduction

Consider the nonlinear inequality constrained optimization problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ f(x) \quad \text{subject to}\ c(x) \le 0, \tag{1}$$

where f and c are $\mathcal{C}^2$ functions from $\mathbb{R}^n$ into $\mathbb{R}$. To simplify we assume that there is a single inequality constraint, but our arguments still hold in the general case. A simple technique to convert the inequality constraint into an equality constraint is to add an artificial variable $y \in \mathbb{R}$ that appears *squared* in the reformulation

$$\underset{x\in\mathbb{R}^n,\ y\in\mathbb{R}}{\text{minimize}}\ f(x) \quad \text{subject to}\ c(x) + y^2 = 0, \tag{2}$$

hence the name, *squared slack*. The two problems are equivalent in the sense that x^* solves (1) if and only if $(x^*, \pm\sqrt{-c(x^*)})$ solves (2). This transformation can be used to derive optimality conditions of an inequality constrained problem, see for example [Ber99, §3.3.2] and an interesting discusion on this subject in [Nas98]. The use of squared slacks has also been advocated for algorithmic purposes, see for example [Tap80]. In [GMW81, §7.4], the authors give theoretical reasons why converting (1) into (2) may be a bad idea, but the literature does not appear to state the numerical drawbacks of the approach clearly.

The purpose of this note is to demonstrate why many algorithmic frameworks of nonlinear programming will fail on problems of the form (2), arising from a squared slack transformation or in their own right. The frameworks that are concerned are the sequential quadratic programming approach (SQP), the augmented Lagrangian approach, conjugate-gradient-based approaches, among

Paul Armand: XLIM UMR 6172, Département de Mathématiques et Informatique, CNRS et Université de Limoges, Limoges, France. E-mail: paul.armand@unilim.fr

Dominique Orban: GERAD and Département de Mathématiques et de Génie Industriel, École Polytechnique de Montréal, CP 6079, Succ. Centre-Ville, Montréal (Québec) H3C 3A7, Canada. E-mail: dominique.orban@gerad.ca (corresponding author)

others. It is important to realize that (2) satisfies the same regularity conditions as (1) and does not defeat the convergence theory of those methods, but illustrates cases where things can, and do, go wrong. We show that the main difficulty comes from the linearization of the first order conditions and a bad choice of the starting point. We believe that the examples given in this note can serve as pedagogical tools to expose the somewhat surprising behavior of some minimization methods for the solution of nonlinear problems. Hopefully, they will also be useful to design better methods that do not suffer the same shortcomings.

2. Shortcomings of some Current Methods

In this section, we present a few example problems on which several of the most widely used nonlinear optimization algorithms are bound to fail. The reason for this failure is that the problems, without being complex or artificial, cause a certain behaviour of the algorithm which corresponds precisely to a case that is usually "ruled out" of the theory by means of assumptions. By way of example, we cite [BT95], where the authors mention various types of solution that an SQP algorithm can produce and that do not correspond to a local constrained minimizer of (1). Those cases are when the iterates

1. are unbounded (violation of Assumption C1 in [BT95]),
2. have a limit point which is a critical point of (1) but not a local minimizer,
3. have a limit point which is a critical point of the measure of infeasibility $(x, y) \mapsto |c(x) + y^2|$.

We will use the following simple parametrized example to expose those problematic cases. It is based on the addition of squared slacks to convert an inequality constraint into an equality.

Example 1. Consider the problem

$$\underset{x,y \in \mathbb{R}}{\text{minimize}} \ \tfrac{1}{2}x^2 \quad \text{subject to } ax - e^x + y^2 = 0, \tag{3}$$

where $a \in \mathbb{R}$ is a parameter. Various values of a will illustrate each of the three shortcomings listed above.

Figures 1, 2 and 3 show the various situations graphically. Note that for all $a \in \mathbb{R}$, Example 1 satisfies the linear independance constraint qualification at all feasible points. One of its features is that the inequality constraint is not active at the minimum of each inequality constrained problem.

The main reason for the failure on Example 1 is the following. Assume that Problem (1) is regular in the sense that $\nabla c(x^*) \neq 0$ at any critical point x^*. Problem (2) is then regular as well and the KKT conditions are necessary for first-order optimality. The Lagrangian for problem (2) is

$$L(x, y, \lambda) = f(x) + \lambda(c(x) + y^2), \tag{4}$$

where λ is the Lagrange multiplier associated to the equality constraint, and at a first-order critical point, we must have

$$\begin{bmatrix} \nabla f(x) + \lambda \nabla c(x) \\ \lambda y \\ c(x) + y^2 \end{bmatrix} = 0. \tag{5}$$
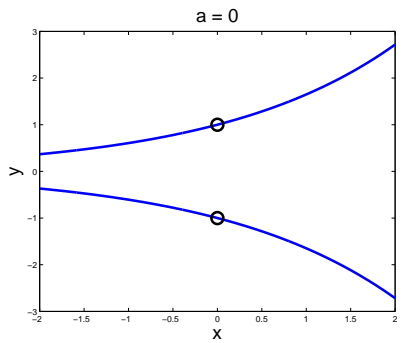
Fig. 1. Illustration of Problem (3) with $a = 0$. The two solutions, $(0, \pm 1)$ are indicated by the black circles and correspond to global minimizers. Starting from $(x_0, 0)$, we observe $x_k \to -\infty$ and $y_k = 0$ for all k. A typical SQP method will stop and claim to have found an optimal solution, with the help of a (very) large multiplier.
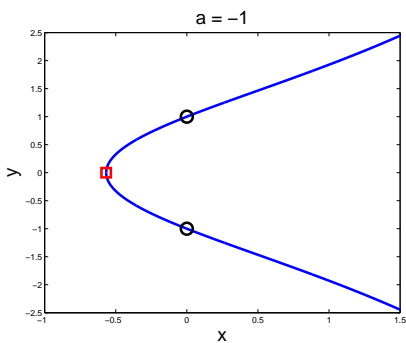

Fig. 2. Illustration of Problem (3) with $a < 0$. The (global) minimizers are $(0, \pm 1)$, indicated by black circles. The point indicated by a red square is a local maximizer. Starting from $(x_0, 0)$, an SQP method will converge to the local maximizer.
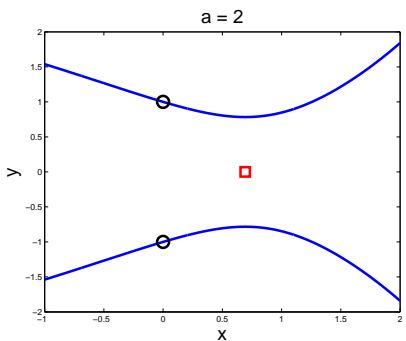

Fig. 3. Illustration of Problem (3) with $a > 0$. The (global) minimizers are $(0, \pm 1)$, indicated by black circles. The point indicated by a red square is a critical point of the infeasibility measure. Using a starting point of the form $(x_0, 0)$, an SQP method will converge to the red square.

Any method that linearizes these conditions at a point (x, y, λ) will compute a step $d = (d_x, d_y, d_\lambda)$ satisfying

$$\lambda d_y + y d_\lambda + \lambda y = 0. \tag{6}$$

Assume at some stage in the process we have $y = 0$ and $\lambda \neq 0$. The linearization (6) then necessarily implies that $d_y = 0$. Thus, if ever $y = 0$ it will always remain equal to zero at subsequent iterations. Of course, if (1) is such that the constraint is inactive at x^*, any method based on a linearization of (5) is bound to fail. Such is the case with the examples of this section. A typical class of methods computing the step according to (6) is the class of SQP methods.

3. Numerical Experiments

In order to eliminate side effects sometimes found in sophisticated implementations, we implemented a basic SQP method for equality-constrained problems, as given in, e.g., [NW99, Algorithm 18.3]. The method is globalized with a backtracking linesearch on the ℓ_1-penalty merit function

$$(x, y) \mapsto f(x) + \nu \|c(x) + y^2\|_1,$$

where $\nu > 0$ is a penalty parameter, and the Lagrange multipliers are updated using the least-squares estimates. However, the results given in this section should be reproducible with state-of-the-art implementations. In all cases, the starting point is chosen as $(0, 0)$ with the initial multiplier set to $\frac{1}{2}$. In the following tables, k is the iteration number, (x_k, y_k) is the current iterate, λ_k is the current Lagrange multiplier estimate, ∇L_k is the gradient of the Lagrangian evaluated at (x_k, y_k, λ_k) and α_k is the steplength computed from the backtracking linesearch.

Table 1 gives the detail of the iterations in the case where $a = 0$. As expected, we see that $y_k = 0$ for all k. Because the least-squares estimates happen to yield the exact multipliers in the present case, the gradient of the Lagrangian is always equal to 0.0. In order to satisfy the first-order optimality conditions, there thus only remains to attain feasibility, which is achieved by having x_k converge to $-\infty$. Note also that $|\lambda_k|$ converges to $+\infty$. This behaviour is that of the first shortcoming of §2. Some backtracking linesearch iterations were only necessary at the first iteration, the unit step was always accepted at the other iterations. Note that the fact that x_k "escapes" to $-\infty$ can be observed in practice by tightening the stopping tolerance.

Table 2 gives the detail of the iterations in the case where $a = -1$. The results are representative of any value $a < 0$. Here, x_k converges to a value which is a local maximizer. This illustrates the second shortcoming of §2. Again, no backtracking was necessary on this problem, except at the first iteration.

Finally, Table 3 gives the detail of the iterations in the case where $a = 2$, but again, the results are representative of any value $a \in (0, e)$. This situation is that given in the third shortcoming of §2. Backtracking was used in this case and the algorithm stopped claiming that the steplength was too small.

As a side note, we remark that when $a = e$, there is a unique feasible point for (2) that has $y = 0$. The SQP algorithm converges to that point, which is in fact a saddle point. For $a > e$, the feasible set is made of two disconnected curves. Each one intersects the axis $y = 0$. One of those intersection points is a local maximizer while the other one is a local minimizer, but neither of them has $x = 0$. Depending on the value of the starting point, the SQP algorithm converges to one or the other. We decided to not report those results here since they do not add new elements to the present analysis.

Finally, the above numerical results hold not only for an initial $y_0 = 0$ but of course, also for y_0 sufficiently close to 0.

4. Other Algorithmic Frameworks

As we showed in (6), any traditional SQP-type method will necessarily generate iterates of the form $(x_k, 0)$ when started from $(x_0, 0)$ with a nonzero Lagrange

k	$\|\nabla L_k\|$	$|c(x_k, y_k)|$	α_{k-1}	x_k	y_k	λ_k
0	5.00e−01	1.00e+00		0.00e+00	0.00e+00	5.00e−01
1	0.00e+00	6.92e−01	3.68e−01	−3.68e−01	0.00e+00	−5.32e−01
2	0.00e+00	2.55e−01	1.00e+00	−1.37e+00	0.00e+00	−5.37e+00
3	0.00e+00	9.36e−02	1.00e+00	−2.37e+00	0.00e+00	−2.53e+01
4	0.00e+00	3.45e−02	1.00e+00	−3.37e+00	0.00e+00	−9.78e+01
5	0.00e+00	1.27e−02	1.00e+00	−4.37e+00	0.00e+00	−3.45e+02
6	0.00e+00	4.66e−03	1.00e+00	−5.37e+00	0.00e+00	−1.15e+03
7	0.00e+00	1.72e−03	1.00e+00	−6.37e+00	0.00e+00	−3.71e+03
8	8.88e−16	6.31e−04	1.00e+00	−7.37e+00	0.00e+00	−1.17e+04
9	0.00e+00	2.32e−04	1.00e+00	−8.37e+00	0.00e+00	−3.60e+04
10	0.00e+00	8.54e−05	1.00e+00	−9.37e+00	0.00e+00	−1.10e+05
11	0.00e+00	3.14e−05	1.00e+00	−1.04e+01	0.00e+00	−3.30e+05
12	0.00e+00	1.16e−05	1.00e+00	−1.14e+01	0.00e+00	−9.84e+05
13	1.78e−15	4.25e−06	1.00e+00	−1.24e+01	0.00e+00	−2.91e+06
14	1.78e−15	1.56e−06	1.00e+00	−1.34e+01	0.00e+00	−8.55e+06
15	1.78e−15	5.75e−07	1.00e+00	−1.44e+01	0.00e+00	−2.50e+07
16	0.00e+00	2.12e−07	1.00e+00	−1.54e+01	0.00e+00	−7.26e+07
17	0.00e+00	7.79e−08	1.00e+00	−1.64e+01	0.00e+00	−2.10e+08
18	0.00e+00	2.86e−08	1.00e+00	−1.74e+01	0.00e+00	−6.06e+08
19	0.00e+00	1.05e−08	1.00e+00	−1.84e+01	0.00e+00	−1.74e+09
20	0.00e+00	3.88e−09	1.00e+00	−1.94e+01	0.00e+00	−5.00e+09

Table 1. Iterations of the SQP method on problem (2) with $a = 0$.

k	$\|\nabla L_k\|$	$|c(x_k, y_k)|$	α_{k-1}	x_k	y_k	λ_k
0	1.00e+00	1.00e+00		0.00e+00	0.00e+00	5.00e−01
1	2.78e−17	5.68e−01	4.56e−01	−2.28e−01	0.00e+00	−1.27e−01
2	0.00e+00	3.59e−02	1.00e+00	−5.44e−01	0.00e+00	−3.44e−01
3	0.00e+00	1.49e−04	1.00e+00	−5.67e−01	0.00e+00	−3.62e−01
4	0.00e+00	2.56e−09	1.00e+00	−5.67e−01	0.00e+00	−3.62e−01

Table 2. Iterations of the SQP method on problem (2) with $a = -1$.

k	$\|\nabla L_k\|$	$|c(x_k, y_k)|$	α_{k-1}	x_k	y_k	λ_k
0	5.00e−01	1.00e+00		0.00e+00	0.00e+00	5.00e−01
1	0.00e+00	7.55e−01	2.93e−01	2.93e−01	0.00e+00	−4.43e−01
2	0.00e+00	6.39e−01	2.06e−01	5.29e−01	0.00e+00	−1.74e+00
3	0.00e+00	6.16e−01	1.00e−01	7.39e−01	0.00e+00	7.82e+00
4	0.00e+00	6.15e−01	1.26e−02	6.57e−01	0.00e+00	−9.23e+00
5	1.11e−16	6.14e−01	4.12e−03	6.92e−01	0.00e+00	−5.18e+02
6	0.00e+00	6.14e−01	1.46e−06	6.93e−01	0.00e+00	−4.33e+05
7	0.00e+00	6.14e−01	2.09e−12	6.93e−01	0.00e+00	7.55e+10
8	0.00e+00	6.14e−01	1.00e−19	6.93e−01	0.00e+00	−5.19e+07

Table 3. Iterations of the SQP method on problem (2) with $a = 2$.

multiplier. In this section, we show that similar conclusions hold for a variety of families of algorithms for nonlinear programming.

Augmented-Lagrangian-based methods fail for a reason similar to that given in §2. Consider the augmented Lagrangian for (2)

$$\mathcal{L}(x, y, \lambda; \rho) = f(x) + \lambda \left(c(x) + y^2\right) + \tfrac{1}{2}\rho \left(c(x) + y^2\right)^2, \tag{7}$$

where $\rho > 0$ is a penalty parameter and where λ is the current estimate of the Lagrange multiplier. The step is computed from the Newton equations for (7), where

$$\nabla \mathcal{L}(x, y, \lambda; \rho) = \begin{bmatrix} \nabla f(x) + \sigma(x, y)\nabla c(x) \\ 2y\sigma(x, y) \end{bmatrix} \tag{8}$$

and

$$\nabla^2 \mathcal{L}(x, y, \lambda; \rho) = \begin{bmatrix} \nabla^2 f(x) + \sigma(x, y)\nabla^2 c(x) + \rho\nabla c(x)\nabla c(x)^\top & 2\rho y \nabla c(x) \\ \\ 2\rho y \nabla c(x)^\top & 2\sigma(x, y) + 4\rho y^2 \end{bmatrix}$$

where $\sigma(x, y) = \lambda + \rho(c(x) + y^2)$. In particular, the Newton equations yield

$$\rho y \nabla c(x)^\top d_x + (\sigma(x, y) + 2\rho y^2) d_y = -y\sigma(x, y)$$

where d_x and d_y are the search directions in x and y respectively. Whenever $y = 0$ the latter equation becomes

$$(\lambda + \rho c(x))\, d_y = 0,$$

so that $d_y = 0$ provided that $\lambda + \rho c(x) \neq 0$, or equivalently, provided that $\sigma(x, 0) \neq 0$. Note that in augmented-Lagrangian methods, the next Lagrange multiplier estimate would be chosen as $\lambda^+ = \lambda + \rho c(x + d_x)$. Note also in the second component of (8) a "complementarity" expression similar to that in the second component of (5).

The iterative minimization of (7) using the truncated conjugate gradient also preserves $y = 0$. Indeed, when started from $(x_0, 0)$, the first search direction p_0 is the steepest descent direction and has the form $(\delta_0, 0)$ for some $\delta_0 \in \mathbb{R}$. The initial residual $r_0 = -p_0$ thus has the same form and so will the next iterate $(x_1, y_1) = (x_0 + \alpha_0 \delta_0, 0)$ and the residual r_1. A property of the conjugate gradient method is that at the k-th iteration, the search direction $p_k \in \operatorname{span}\{p_{k-1}, r_k\}$ [GL96, Corollary 10.2.4]. Therefore p_1 necessarily shares the same form $(\delta_1, 0)$. A recursion argument thus shows that the k-th conjugate-gradient iterate has the form $(\xi_k, 0)$. This method will therefore also be unable to depart from the hyperplane $y = 0$.

5. Discussion

When the constraint of (1) is inactive, the optimal multiplier is $\lambda = 0$. However, in (2) we will frequently observe $|\lambda| \to +\infty$ when started with $y_0 = 0$ to reach dual feasibility. Looking for instance at the results of Table 1, the final iterate $(-19.4, 0)$ is feasible to within the prescribed tolerance. To compensate in dual feasibility, we need to have a large multiplier. In this sense, the addition of squared slacks has created a critical point at infinity.

Unfortunately, the problem is difficult to avoid. Of course, it seems that we should not add squared slacks to convert inequality constraints into equality constraints. However problems having the form (2) where the variable y does not appear in the objective may arise in their own right.

It is worth noting that any problem of the form

$$\begin{aligned} \underset{x,y}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c(x) + g(y) = 0, \end{aligned}$$

where $g(0) = 0$, $g'(0) = 0$ and $g''(0) \neq 0$ will exhibit a similar behavior. For instance, the functions $g(y) = \cosh(y)$, and $g(y) = \int \arctan(y) \mathrm{d}y$ have the desired properties. The objective may also have the form $f(x,y)$ and the same conclusions hold if $\partial f(x,0)/\partial y = 0$.

References

Ber99.   D. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont MA, USA, 2nd edition, 1999.

BT95.   P. T. Boggs and J. W. Tolle. Sequential quadratic programming. *Acta Numerica*, 4:1–51, 1995.

GL96.   G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, USA, 3rd edition, 1996.

GMW81. Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization*. Academic Press, Inc., London, 1981.

GOT03.   N. I. M. Gould, D. Orban, and Ph. L. Toint. An Interior-Point ℓ_1-Penalty Method for Nonlinear Optimization. Technical Report RAL-TR-2003-022, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 2003.

Nas98.   S. G. Nash. SUMT (Revisited). *Operations Research*, 46:763–775, 1998.

NW99.   J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer series in Operations Research. Springer-Verlag New York, Inc., 175 Fifth Avenue, New York, NY 10010, USA, 1999.

Tap80.   R. A. Tapia. On the role of slack variables in quasi-Newton methods for constrained optimization. In L. C. W. Dixon and G. P. Szegö, editors, *Numerical Optimization of Dynamic Systems*, pages 235–246. North Holland Publishing Company, 1980.