ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

# Optimality Measures for Performance Profiles

**Elizabeth D. Dolan, Jorge J. Moré and Todd S. Munson**

Mathematics and Computer Science Division

Preprint ANL/MCS-P1155-0504

May 2004

*Work supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract W-31-109-Eng-38, and by the National Science Foundation grant DMI-0322580.

## Contents


**Abstract** **1**

**1 Introduction** **1**

**2 Approximate Active Sets and Multipliers** **3**

**3 Optimality Measures and Convergence Tests** **5**

**4 Scale Invariance** **10**

**5 Benchmarking with COPS** **12**

**6 Computational Experiments** **13**

**7 Concluding Remarks** **17**

**References** **17**

# Optimality Measures for Performance Profiles*

**Elizabeth D. Dolan,† Jorge J. Moré,§ and Todd S. Munson§**

**Abstract**

We examine the importance of optimality measures when benchmarking a set of solvers, and show that scaling requirements lead to a convergence test for nonlinearly constrained optimization solvers that uses a mixture of absolute and relative error measures. We demonstrate that this convergence test is well behaved at any point where the constraints satisfy the Mangasarian-Fromovitz constraint qualification and also avoids the explicit use of a complementarity measure. Computational experiments explore the impact of this convergence test on the benchmarking process with performance profiles.

## 1 Introduction

Benchmarking is essential when developing numerical software because this process reveals the strengths and weaknesses of the software. To obtain useful information from the benchmark, attention must be given to the convergence tests and tolerances used by competing solvers. In particular, a comparison between a solver that computes a highly accurate solution and another that computes an inaccurate solution can be highly biased.

We propose a convergence test for benchmarking nonlinearly constrained optimization solvers. The use of a specific convergence test means that all solvers are treated equally and that we can guarantee that the approximate solutions returned have the same level of accuracy. This removes a major bias in the benchmarking process. While we are primarily interested in convergence tests that can be used by solvers for the constrained optimization problem

$$\min \{ f(x) : l \leq c(x) \leq u \} , \tag{1.1}$$

our remarks apply to the benchmarking of general iterative solvers.

We could introduce a specific convergence test in the benchmarking process by modifying all the solvers in the benchmark. This approach is not feasible, however, unless we have access to the source code for all of the solvers. Even with this access, adding a new convergence test requires detailed knowledge of the solver. In general, only the developers can reliably modify their code. Furthermore, the cost of applying a convergence test may be nonnegligible if some of the required information is not readily available and must be computed, resulting in a noticeable increase in the total time taken to solve the benchmark problem.

*Work supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract W-31-109-Eng-38, and by the National Science Foundation grant DMI-0322580.

†Department of Computer Science, The University of North Carolina at Chapel Hill (dolan@cs.unc.edu).

§Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois 60439 ({more,tmunson}@mcs.anl.gov).

An alternative approach is to compute and check a specific convergence test for all the solvers a posteriori. In this approach each solver is first asked to solve the problem by enforcing its native convergence test with default tolerances. If the approximate solution returned does not satisfy the a posteriori convergence test, then the native solver tolerances are reduced and the problem is re-solved. This process is repeated until the a posteriori convergence test is satisfied or a time limit is exceeded. This approach guarantees that the solution reported is at least as accurate as all the other solutions. Moreover, this approach can be readily implemented because access to the source code for each solver is not required.

We impose two major requirements on the convergence test. The first requirement is that the accuracy be based only on the approximate solution returned by the solver; all other quantities needed to assess the level of accuracy must be computed independently. The other requirement is scale invariance of the convergence test when either the function and constraints are scaled or when the variables are scaled.

Our convergence test for constrained optimization problems is based on the first-order optimality conditions but uses a mixture of absolute and relative error measures. Section 2 defines these measures and uses them to define $\tau$-active constraints and the associated approximate multipliers. We have used the nonstandard term $\tau$-active constraint because the term $\epsilon$-active constraint is invariably related to absolute error measures.

We define a convergence test in Section 3 in terms of measures for feasibility and stationarity that take into account the relative size of the constraints. We show that this convergence test is well behaved at any point where the constraints satisfy the Mangasarian-Fromovitz constraint qualification, but may fail if this constraint qualification does not hold. We also examine the relationship of our convergence test to other tests commonly used in optimization software and show that our convergence test avoids the explicit use of a complementarity measure.

Section 4 examines the scaling properties of the convergence tests introduced in Section 3. We show that these convergence tests are scale invariant under reasonable conditions. We also examine the scale invariance properties of other convergence tests.

Section 5 describes the benchmarking process based on the convergence test introduced in Section 3. We use performance profiles [3] and Version 3.0 of COPS [4], the Constrained Optimization Problem Set, to evaluate the effect of the convergence tests. Performance profiles have been used in a wide variety of benchmarking studies (for example, [1, 2, 8, 9]), but in almost all cases the convergence criteria have not been specified in detail. This is certainly the case for benchmarking studies in the AMPL or GAMS modeling environment because in these cases there is no easy access to the source of the solvers. Our computational results in Section 6 show the importance of using a specific convergence test for all the solvers in the benchmark because some of the solvers have widely different notions of optimality with their default tolerances. Our results show that the trends in the performance profiles remain the same, but that important differences arise with a consistent convergence test.

## 2 Approximate Active Sets and Multipliers

Given $\tau$ in $(0,1)$ and an approximate solution $x$ to the general optimization problem (1.1), we define measures of optimality in terms of the set of $\tau$-active constraints and associated approximate multipliers.

The set of $\tau$-active constraints at $x$ consists of all constraints that are near the bounds $l$ and $u$ as measured by $\tau$. Since we want our measure to be almost scale independent, the measure of nearness involves a mixture of the absolute and relative errors. Given real numbers $\xi_1$ and $\xi_2$, we define

$$\delta[\xi_1,\xi_2] = \min\left\{|\xi_1-\xi_2|, \frac{|\xi_1-\xi_2|}{|\xi_1|+|\xi_2|}\right\},$$

with $\delta[0,0]=0$. We then have $\delta[\xi_1,\xi_2]\in[0,1]$. Moreover,

$$\delta[\xi_1,\xi_2]\le\tau \quad \text{if and only if} \quad |\xi_1-\xi_2|\le\tau\max\{1,|\xi_1|+|\xi_2|\}.$$

In particular, $\delta[\xi_1,\xi_2]$ is the relative error whenever $|\xi_1|+|\xi_2|\ge 1$.

Given $x,y$ in $\mathbb{R}^n$, we extend this definition of error to vectors by defining a vector-valued error measure $d:\mathbb{R}^n\times\mathbb{R}^n\mapsto\mathbb{R}^n$ by

$$d[x,y] = (d_k[x,y]) = (\delta[x_k,y_k])$$

so that $d[x,y]$ is a vector where $d_k[x,y]=\delta[x_k,y_k]$ for each component $k$. This definition implies that $d[x,y]=|x-y|$ for vectors of modest size, that is, $\|x\|_\infty+\|y\|_\infty\le 1$. Moreover, if $\|\cdot\|$ is a monotone norm ($|x|\le|y|$ implies that $\|x\|\le\|y\|$), then

$$\min\left\{\|x-y\|,\ \frac{\|x-y\|}{\|x\|_\infty+\|y\|_\infty}\right\}\le\|d[x,y]\|\le\min\left\{\|x-y\|,\left\|\left(\frac{|x_k-y_k|}{|x_k|+|y_k|}\right)\right\|\right\}.$$

This inequality shows that $\|d[x,y]\|$ is closely related to the absolute and relative error between $x$ and $y$.

This definition of error between vectors can be modified in various ways. In particular, if we wish to introduce different levels of absolute and relative errors by requiring that

$$|\xi_1-\xi_2|\le\tau_a\tau, \quad \text{or} \quad \frac{|\xi_1-\xi_2|}{|\xi_1|+|\xi_2|}\le\tau \tag{2.1}$$

for some $\tau_a\ge 0$, then we can define

$$\delta[\xi_1,\xi_2] = \min\left\{\left(\frac{1}{\tau_a}\right)|\xi_1-\xi_2|, \frac{|\xi_1-\xi_2|}{(|\xi_1|+|\xi_2|)}\right\}. \tag{2.2}$$

The case where $\tau_a=0$ is equivalent to using a purely relative error test and can be obtained as the limit of $\delta[\xi_1,\xi_2]$ as $\tau_a$ converges to zero.

If we use the definition (2.2), then the inequality $\delta[\xi_1,\xi_2]\le\tau$ is equivalent to (2.1). Moreover, we now have

$$\delta[\xi_1,\xi_2]\le\tau \quad \text{if and only if} \quad |\xi_1-\xi_2|\le\tau\max\{\tau_a,|\xi_1|+|\xi_2|\}.$$

This equivalence shows that the test $\delta[\xi_1, \xi_2] \le \tau$ reduces to the relative convergence test in (2.1) unless $|\xi_1| + |\xi_2| \le \tau_a$, and then becomes the absolute convergence test in (2.1). We can emphasize the relative error in (2.2) by choosing $\tau_a \le \tau$. We later show that this influences the scale invariance of convergence tests based on $\delta[\cdot, \cdot]$. We set $\tau_a = 1$ in the remainder of this paper, but all results can be easily modified for any $\tau_a \ge 0$.

Given $\tau$ in $(0, 1)$ and the definition $d : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}^n$ of error between vectors, we define the set of $\tau$-active constraints at $x$ by

$$\mathcal{A}_\tau(x) = \{k : \min\{d_k[c(x), l], d_k[c(x), u]\} \le \tau\}. \tag{2.3}$$

In general, $\tau$ is related to the expected accuracy of the optimization algorithm because the set $\mathcal{A}_\tau(x)$ contains all constraints that are nearly active as measured by $\tau$. Moreover, for $\tau$ sufficiently small, $\mathcal{A}_\tau(x)$ is the set $\mathcal{A}_0(x)$ of active constraints at $x$.

This definition of $\tau$-active constraints for the optimization problem (1.1) reduces to standard notions for optimization problems in generic form. For example, if we consider

$$\min\{f(x) : c(x) \le 0\}, \tag{2.4}$$

then $d_k[c(x), 0] = \min\{|c_k(x)|, 1\}$, and thus $\mathcal{A}_\tau(x) = \{k : |c_k(x)| \le \tau\}$. The more general definition (2.3) is needed for dealing with realistic problems that have not been forced into standard forms.

We measure optimality by computing multipliers explicitly with the requirement that the multipliers lie in the cone $S_\tau(x)$ associated with $\mathcal{A}_\tau(x)$, where

$$S_\tau(x) = \left\{ v : \begin{array}{lll} v_k \text{ free} & \text{if} & d_k[c(x), l] \le \tau,\ d_k[c(x), u] \le \tau \\ v_k \ge 0 & \text{if} & d_k[c(x), l] \le \tau,\ d_k[c(x), u] > \tau \\ v_k \le 0 & \text{if} & d_k[c(x), l] > \tau,\ d_k[c(x), u] \le \tau \\ v_k = 0 & \text{if} & d_k[c(x), l] > \tau,\ d_k[c(x), u] > \tau. \end{array} \right. \tag{2.5}$$

If the $k$th constraint is an equality constraint and is $\tau$-active, then $l_k = u_k$, and thus the $k$th component is free in the cone $S_\tau(x)$. Moreover, the $k$th component is zero if the $k$th constraint is not $\tau$-active.

Recall that a Karush-Kuhn-Tucker (KKT) pair $(x^*, \lambda^*)$ for the optimization problem (1.1) satisfies

$$\nabla f(x^*) = \nabla c(x^*)\lambda^*, \qquad \lambda^* \in S_0(x^*),$$

where $\lambda^*$ are the multipliers. Thus it is natural to determine multipliers via the optimization problem

$$\min\{\|\nabla f(x) - \nabla c(x) v\| : v \in S_\tau(x)\}, \tag{2.6}$$

for some norm $\|\cdot\|$. If $\lambda(x, \tau)$ is a solution of (2.6), then

$$r(x, \tau) = \nabla f(x) - \nabla c(x)\lambda(x, \tau)$$

is the residual in the KKT conditions. In the next section we will show that $\|r(x, \tau)\|$ is an upper bound on our measure of optimality.

If we consider the special case of bound-constrained problems where $c(x) = x$, the residual is given by

$$r(x,\tau) = \begin{cases} 0 & \text{if} \quad d_k[x,l] \le \tau,\ d_k[x,u] \le \tau \\ \max(0, \partial_k f(x)) & \text{if} \quad d_k[x,l] \le \tau,\ d_k[x,u] > \tau \\ \min(0, \partial_k f(x)) & \text{if} \quad d_k[x,l] > \tau,\ d_k[x,u] \le \tau \\ \partial_k f(x) & \text{if} \quad d_k[x,l] > \tau,\ d_k[x,u] > \tau \end{cases}$$

if we use any $l_p$ norm ($p < \infty$) in (2.6). This expression shows that $-r(x,0)$ agrees with the projected gradient as used, for example, by Chin and Moré [7, page 1105]. Thus $-r(x,\tau)$ can be viewed as a generalization of the projected gradient to general nonlinearly constrained problems.

The choice of the $l_2$ norm in the computation of the multipliers in (2.6) leads to a linearly constrained least squares problem. We prefer to use the $l_\infty$ norm and thus define $\lambda(x,\tau)$ as a solution of

$$\min \{\|y\|_\infty : y = \nabla f(x) - \nabla c(x) v,\ v \in S_\tau(x)\}. \tag{2.7}$$

This problem can be formulated as a linear programming problem, and thus $\lambda(x,\tau)$ can be readily computed by several solvers. The solution $\lambda(x,\tau)$ to (2.7) is a set of approximate multipliers for the original optimization problem.

## 3 Optimality Measures and Convergence Tests

We define a convergence test for the optimization problem (1.1) in terms of measures of feasibility, complementarity, and stationarity that takes into account the relative size of the constraints. Given tolerances $\tau_1, \ldots, \tau_p$ and measures of optimality $\nu_i : \mathbb{R}^n \mapsto \mathbb{R}_+$, a convergence test defines a set

$$\mathcal{C}(\tau) = \{x \in \mathbb{R}^n : \nu_i(x) \le \tau_i,\ 1 \le i \le p\}$$

of acceptable points. A minimal requirement on the convergence test is that $\mathcal{C}(0)$ contain only KKT points.

The standard measure of feasibility for constraints of the form $l \le c(x) \le u$ can be written in the form

$$\|\text{mid}\{c(x) - l, 0, c(x) - u\}\|,$$

where $\text{mid}\{\cdot,\cdot,\cdot\}$ is the argument in the middle. We introduce the relative size of the constraints by defining the *feasibility measure*

$$\nu_f(x) = \|v(x)\|,$$

where $\|\cdot\|$ is any norm and

$$v_k(x) = \begin{cases} 0 & \text{if} \quad l_k \le c_k(x) \le u_k \\ \min(d_k[c(x), l], d_k[c(x), u]) & \quad \text{otherwise.} \end{cases}$$

A short computation shows that if $\|\cdot\|$ is a monotone norm, then

$$\nu_f(x) \leq \|\text{mid}\{c(x) - l, 0, c(x) - u\}\|,$$

and thus $\nu_f(x)$ is bounded above by the absolute error in the constraint violation. We define a vector $x \in \mathbb{R}^n$ to be $\tau$-feasible if $\nu_f(x) \leq \tau$.

In most cases we use the $l_\infty$ norm. For this norm a $\tau$-feasible vector is precisely a vector that satisfies

$$l_k \leq c_k(x) \leq u_k \quad \text{or} \quad \min\{d_k[c(x), l], d_k[c(x), u]\} \leq \tau, \qquad 1 \leq k \leq n.$$

Moreover, if we consider the generic optimization problem (2.4) and any $\tau \in (0, 1)$, then $x$ is $\tau$-feasible if and only if $\|c(x)_+\| \leq \tau$. The equivalence of norms in finite dimensions shows that these results hold for any norm provided $\tau$ is sufficiently small. Finally, $0 \leq \nu_f(x) \leq 1$ with the $l_\infty$ norm.

An advantage of computing the multipliers by either (2.6) or (2.7) is that all the multipliers of the $\tau$-active constraints have the proper sign. Moreover, $\lambda_k(x) = 0$ if the $k$th constraint is not $\tau$-active. Hence, complementarity should be defined in terms of the constraint violation for the $\tau$-active constraints. We define

$$\nu_c(x, \tau) = \|w(x, \tau)\|$$

as a *measure of complementarity*, where

$$w_k(x, \tau) = \begin{cases} \min(d_k[c(x), l], d_k[c(x), u]) & k \in \mathcal{A}_\tau(x) \\ 0 & \text{otherwise} \end{cases}$$

The definition of $\mathcal{A}_\tau(x)$ implies that $\nu_c(x, \tau) \leq \tau$ when we use the $l_\infty$ norm, and thus the complementarity measure $\nu_c$ is never large.

An optimization algorithm should deliver approximate solutions that are $\tau$-feasible, that is, $\nu_f(x) \leq \tau$. For $\tau$-feasible vectors in the $l_\infty$ norm we have

$$\nu_f(x) \leq \nu_c(x, \tau) \leq \tau,$$

and thus the complementarity measure dominates for $\tau$-feasible vectors.

The classical method to measure stationarity is with the norm of the residual in the KKT conditions. However, in order to take into account the relative size of $\nabla f(x)$, we use the *stationarity measure*

$$\nu_s(x, \tau) = \|d[\nabla f(x), \nabla c(x)\lambda(x, \tau)]\|.$$

If $\|\cdot\|$ is a monotone norm, then the definition of the function $d : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$ implies that

$$\nu_s(x, \tau) \leq \|\nabla f(x) - \nabla c(x)\lambda(x, \tau)\|$$

so that $\nu_s(x, \tau)$ is bounded above by the norm of the residual in the KKT conditions.

Table 3.1 summarizes all the measures of optimality for an optimization problem. We have defined these measures in terms of an arbitrary norm, but we use the $l_\infty$ norm. We define a convergence test in terms of tolerances $\tau_f$ and $\tau_s$ by computing multipliers via (2.7) with the $\tau$-active set determined by (2.3) with $\tau = \tau_f$, and requiring that

$$\nu_f(x) \le \tau_f, \qquad \nu_s(x, \tau_f) \le \tau_s. \tag{3.1}$$

The definition of $\nu_c$ guarantees that $\nu_c(x, \tau_f) \le \tau_f$ in the $l_\infty$ norm, and thus it is not necessary to require a test on complementarity. This statement seems to be incorrect at first sight but is a consequence of using a set of multipliers that are zero if the constraint is not $\tau$-active but otherwise have the proper sign.

Table 3.1: Measures of optimality

| | |
|---|---|
| Feasibility | $\nu_f(x)$ |
| Complementarity | $\nu_c(x, \tau)$ |
| Stationarity | $\nu_s(x, \tau)$ |

The following result shows that the optimality measures in Table 3.1 behave appropriately if we consider a sequence of tolerances that converge to zero. We assume that we have a sequence that converges to a point $x^*$ that satisfies the Mangasarian-Fromovitz constraint qualification at $x^*$ in the sense that

$$\nabla c(x^*)v = 0, \quad v \in S_0(x^*) \qquad \Longrightarrow \qquad v = 0,$$

where $S_0(x^*)$ is the cone (2.5) at $\tau = 0$. This constraint qualification reduces to the classical Mangasarian-Fromovitz constraint qualification for the generic optimization problem (2.4).

**Theorem 3.1** *Assume that $\tau_a > 0$ in* (2.1)*, that $\{\tau_k\}$ is a sequence of tolerances that converges to zero, and that $\{x_k\}$ is a sequence that converges to $x^*$. If $\{\nu_f(x_k)\}$ converges to zero, then $x^*$ is feasible. Moreover, if $\{\nu_s(x_k, \tau_k)\}$ converges to zero and the constraints satisfy the Mangasarian-Fromovitz constraint qualification at $x^*$, then $x^*$ is a KKT point of the optimization problem* (1.1)*.*

**Proof.** Since $\tau_a > 0$, the definition (2.1) of $\delta[\cdot,\cdot]$ implies that $d[\cdot,\cdot]$ preserves convergent sequences, that is, $\{d[y_k, y^*]\}$ converges to zero if and only if $\{y_k\}$ converges to $y^*$. Hence, the definition of $\nu_f$ shows that $x^*$ is feasible.

We now assume that $\{\nu_s(x_k, \tau_k)\}$ converges to zero and show that $x^*$ is a KKT point of the optimization problem (1.1) if the constraints satisfy the Mangasarian-Fromovitz constraint qualification at $x^*$. Consider the sequence $S_{\tau_k}(x_k)$, and note that for any subsequence $\mathcal{K}$ there is a refinement $\mathcal{K}_0$ such that $S_{\tau_k}(x_k)$ is constant for $k \in \mathcal{K}_0$. Moreover, $S_{\tau_k}(x_k) \subset S_0(x^*)$ for $k \in \mathcal{K}_0$. Thus, if $\{\lambda(x_k, \tau_k)\}$ is bounded, then the definition of

$S_{\tau_k}(x_k)$, shows that any limit point $\lambda^*$ of $\{\lambda(x_k,\tau_k)\}$ is a valid set of multipliers for $x^*$, that is, $\lambda^* \in S_0(x^*)$.

We show that $\{\lambda(x_k,\tau_k)\}$ is bounded by first noting that since $\{\nu_s(x_k,\tau_k)\}$ converges to zero, $\{\nabla c(x_k)\lambda(x_k,\tau_k)\}$ is bounded. Since $\{x_k\}$ converges to $x^*$ and the constraints satisfy the Mangasarian-Fromovitz constraint qualification at $x^*$, the sequence $\{\lambda(x_k,\tau_k)\}$ is bounded.

We have now shown that $\{\lambda(x_k,\tau_k)\}$ is bounded and that any limit point of this sequence is a valid set of multipliers for $x^*$. Since $\{\nu_s(x_k,\tau_k)\}$ converges to zero, $x^*$ is a KKT point of the optimization problem (1.1). ■

We can generalize Theorem 3.1 by noting that the proof of this result shows that $\nu_s$ is lower semicontinuous, that is,

$$\liminf_{(x,\tau)\to(x^*,0)} \nu_s(x,\tau) \geq \nu_s(x^*,0),$$

where we have made explicit the dependence of $\nu_s$ on $\tau$. This is of interest because simple examples show that $\nu_s$ is not continuous in $(x,\tau)$, although $\nu_s(\cdot,\tau)$ is certainly continuous.

The assumption that the constraints satisfy the Mangasarian-Fromovitz constraint qualification at $x^*$ is essential for Theorem 3.1. Consider, for example, the optimization problem in $\mathbb{R}$,

$$\min\left\{\xi : \tfrac{1}{2}\xi^2 \geq 0\right\}.$$

If $\{\xi_k\}$ is any monotone sequence that converges to zero and $\tau_k = \xi_k$, then the multipliers determined by (2.7) are $\lambda(\xi) = 1/\xi$. Thus, in this case, $\{\nu_f(\xi_k,\tau_k)\}$ converges to zero, but $\xi^* = 0$ is not a KKT point. Of course, in this case the Mangasarian-Fromovitz constraint qualification fails and the approximate multipliers are unbounded.

We now relate the optimality conditions in Table 3.1 to convergence tests used in optimization algorithms. Consider, for example, the generic optimization problem

$$\min\{f(x) : c(x) \leq 0\}.$$

Given tolerances $\tau_f$, $\tau_c$ and $\tau_s$, assume that the convergence tests are

$$c_k(x) \leq \tau_f, \qquad \pi_k(x) \leq \tau_c, \qquad |\pi_k(x)| \leq \tau_c \quad \text{if} \quad |c_k(x)| > \tau_f \tag{3.2}$$

on the approximate solution $x$ and multiplier estimates $\pi(x)$, and

$$\|\nabla f(x) - \nabla c(x)\pi(x)\|_\infty \leq \tau_s \tag{3.3}$$

on the residual of the KKT conditions. These are suitable convergence tests in the sense that if all the tolerances are set to zero, then we recover the KKT conditions.

An important difference between these convergence tests and the optimality measures in Table 3.1 is that the multiplier estimates $\pi_k(x)$ are not guaranteed to be nonpositive. However, if we define

$$\lambda_k(x) = \begin{cases} \min(\pi_k(x),0) & \text{if} \quad k \in \mathcal{A}_{\tau_f}(x) \\ 0 & \text{otherwise,} \end{cases}$$

then $\lambda_k(x) \le 0$ are multiplier estimates with

$$|\lambda_k(x) - \pi_k(x)| \le \tau_c. \tag{3.4}$$

This estimate holds if $k \in \mathcal{A}_{\tau_f}(x)$ and $\pi_k(x) \le 0$ because then $\lambda_k(x) = \pi_k(x)$. If $k \notin \mathcal{A}_{\tau_f}(x)$ or $\pi_k(x) > 0$, then $\lambda_k(x) = 0$. Moreover, in either case (3.2) implies that $|\pi_k(x)| \le \tau_c$. Hence, (3.4) also holds in this case.

The estimate (3.4) shows that the residual of the KKT condition is bounded in terms of the tolerances and the problem data. Indeed, a direct consequence of (3.3) and (3.4) is that

$$\|\nabla f(x) - \nabla c(x)\lambda(x)\|_\infty \le \tau_s + \|\nabla c(x)\|_\infty \tau_c.$$

Hence, we have shown that if the convergence tests (3.2) and (3.3) hold, then

$$\nu_f(x) \le \tau_f, \qquad \nu_s(x, \tau_f) \le \tau_s + \|\nabla c(x)\|_\infty \tau_c. \tag{3.5}$$

This is an important observation because (3.2) and (3.3) are closely related to convergence tests used by optimization solvers such as SNOPT and KNITRO. For example, instead of (3.2), SNOPT [6] requires that

$$c_k(x) \le \tau_f, \qquad \pi_k(x) \le \tau_c, \qquad |c_k(x)\pi_k(x)| \le \tau_c. \tag{3.6}$$

This is a stronger convergence test than (3.2) because if (3.6) holds, then

$$|\pi_k(x)| \le \frac{\tau_c}{\tau_f} \qquad \text{if} \quad |c_k(x)| > \tau_f.$$

We also note that (3.6) not only implies (3.2) but also implies a bound on the multipliers of the $\tau$-active constraints. Assume, for example, that the $k$th constraint is $\tau$-active with

$$|c_k(x)| = \sigma\tau_f, \qquad \sigma \in (0, 1).$$

Under this assumption (3.6) implies that

$$|\pi_k(x)| \le \frac{1}{\sigma}\frac{\tau_c}{\tau_f}.$$

Thus, for problems with large multipliers, this bound shows that $\tau_f$ may have to be relatively small in order to satisfy (3.6).

Similar remarks apply to the optimization solver KNITRO [13]. The convergence test in KNITRO replaces (3.2) by

$$c_k(x) \le \tau_f, \qquad \pi_k(x) < 0, \qquad |c_k(x)\pi_k(x)| \le \tau_c. \tag{3.7}$$

Thus, the only difference between the convergence test in SNOPT and KNITRO is that KNITRO guarantees that the multipliers are negative. For the KNITRO convergence test we can show that if

$$\lambda_k(x) = \begin{cases} \pi_k(x) & \text{if} \quad k \in \mathcal{A}_{\tau_f}(x) \\ 0 & \quad\quad \text{otherwise,} \end{cases}$$

then (3.4) holds. Hence, (3.5) also holds.

The convergence tests in SNOPT and KNITRO require the user to choose tolerances $\tau_P$ and $\tau_D$. SNOPT sets

$$\tau_f = \tau_P(1 + \|x\|), \qquad \tau_c = \tau_s = \tau_D(1 + \|\pi(x)\|),$$

while KNITRO sets

$$\tau_f = \max\left\{\tau_P \max(1, \|c(x_0)_+\|_\infty), \tau_0\right\}, \qquad \tau_c = \tau_s = \max\left\{\tau_D \max(1, \|\nabla f(x)\|_\infty), \tau_0\right\}$$

for some absolute tolerance $\tau_0 \geq 0$. An important difference between these tests and (3.1) is that there is no explicit test of the complementarity error in (3.1). Another difference is that with (3.1) it is readily apparent when the solution is not sufficiently accurate because in these cases one of the measures in Table 3.1 is above the required tolerance (but less than one). On the other hand, with unscaled tests, large or small values for an optimality measure may just be a reflection of the scale of the problem.

## 4 Scale Invariance

We now examine the invariance properties of convergence tests when the general optimization problem (1.1) is transformed into an equivalent optimization problem

$$\min\left\{\hat{f}(x) : \hat{l} \leq \hat{c}(x) \leq \hat{u}\right\}. \tag{4.1}$$

Scale invariance is a desirable attribute of a convergence test because then the choice of tolerances can be made on the basis of the desired accuracy, without interference from the scale of the problem.

We first consider the change of scale $\hat{f} = \alpha f$, $\hat{c} = \beta c$, where $f$ is scaled by $\alpha > 0$, and the constraints $c$ are scaled by $\beta > 0$. With this change of scale we must also scale the bounds $l$ and $u$ in the optimization problem (1.1) by $\beta$ so that $\hat{l} = \beta l$ and $\hat{u} = \beta u$. The optimization problems (1.1) and (4.1) are equivalent under this change of scale in the sense that they have the same solution. Moreover, most optimization algorithms generate the same set of iterates when the optimization problem is scaled in this manner.

We also consider the change of scale $x \mapsto Sx$ in (1.1), where $S$ is a nonsingular diagonal matrix. In this case we have $\hat{f}(x) = f(Sx)$ and $\hat{c}(x) = c(Sx)$ in (4.1). With this change of scale any minimizer $x^*$ of the optimization problem (1.1) generates a minimizer $\hat{x}^*$ of (4.1) via $x^* = S\hat{x}^*$. The converse of this statement also holds. Thus, both optimization problems, (1.1) and (4.1), have the same solution sets.

We explore the scale invariance of convergence tests under the assumption that all absolute tolerances are set to zero. For the optimality measures in Table 3.1, this means that $\tau_a = 0$ in the definition (2.2) of $\delta[\cdot,\cdot]$. Under this assumption

$$d[Sx, Sy] = d[x, y], \tag{4.2}$$

for all nonsingular diagonal matrices $S$ and vectors $x$ and $y$. If $\tau_a > 0$, then (4.2) holds if

$$|s_k| \left( |x_k| + |y_k| \right) \geq \tau_a, \qquad 1 \leq k \leq n.$$

Hence, scale invariance of $d[\cdot,\cdot]$ holds if the scaled variables for at least one of the vectors is above the absolute tolerance level, that is, $|s_k||x_k| \geq \tau_a$ or $|s_k||y_k| \geq \tau_a$.

The change of scale $\hat{f} = \alpha f$, $\hat{c} = \beta c$, in $f$ and $c$ implies that $\hat{\lambda}(x) = (\alpha/\beta)\lambda(x)$ for the multiplier defined by (2.7), and hence

$$\nabla \hat{f}(x) = \alpha \nabla f(x), \qquad \nabla \hat{c}(x)\hat{\lambda}(x) = \alpha \nabla c(x)\lambda(x).$$

Thus, (4.2) shows that both $\nu_f(x)$ and $\nu_s(x)$ are invariant under this change of scale.

The scaling behavior (4.2) of $d[\cdot,\cdot]$ also shows that the set of $\tau$-active constraints is invariant under the change of scale $x \mapsto Sx$, where $S$ is a nonsingular diagonal matrix. On the other hand, the multipliers $\lambda(x)$ defined by (2.7) are not invariant under this change of scale because

$$\nabla \hat{f}(x) = S\nabla f(x), \qquad \nabla \hat{c}(x) = S\nabla c(x).$$

However, if we modify the norm in (2.7) and consider

$$\min \left\{ \|D^{-1}y\|_\infty : y = \nabla f(x) - C(x)v, \ v \in S_\tau \right\},$$

where $D$ is determined from the problem data, then we can have scale invariance. For example, if

$$d_k = \max \left\{ |\partial_k f(x_0)|, \|\partial_k c(x_0)\| \right\},$$

then $\lambda(x)$ is invariant under this change of scale. If we take into account (4.2), then we have shown that with this modification all the optimality measures in Table 3.1 are invariant.

An analysis of the scaling behavior of the convergence tests (3.2) and (3.3) requires that we specify how the multiplier estimate $\pi(x)$ depends on the change of scale. If we consider the change of scale where $\hat{f} = \alpha f$ and $\hat{c} = \beta c$, and assume that $\hat{\pi}(x) = (\alpha/\beta)\pi(x)$ under this change of scale, then (3.3) shows that this convergence test is scale invariant if the tolerances are scaled by the appropriate problem data. On the other hand, if we consider the change of scale $x \mapsto Sx$, where $S$ is a nonsingular diagonal matrix, then the convergence test (3.3) is not scale invariant unless (3.3) is modified to use a norm scaled by the problem data.

The scaling properties of the convergence test (3.2) are shared by (3.7), but this is not the case for (3.6). Indeed, if we consider the change of scale where $\hat{f} = \alpha f$ and $\hat{c} = \beta c$, and assume that $\hat{\pi}(x) = (\alpha/\beta)\pi(x)$, then $\hat{c}(x)\hat{\pi}(x) = \alpha c(x)\pi(x)$. Hence, (3.6) shows that $\tau_c$ must be scaled by both $\alpha/\beta$ and $\alpha$. Since $\tau_c$ cannot absorb two different changes of scale, the convergence test (3.6) is not scale invariant.

## 5 Benchmarking with COPS

We use performance profiles [3] and COPS [4] (Version 3.0) to evaluate the effect of the optimality measures in Table 3.1. The COPS benchmark collection provides a selection of difficult nonlinearly constrained optimization problems from applications in optimal design, fluid dynamics, parameter estimation, mesh smoothing, computational chemistry, and optimal control, among others. Moreover, each application has a short description of the formulation of the application as an optimization problem. Figure 5.1 has plots for the solutions to four problems in COPS.

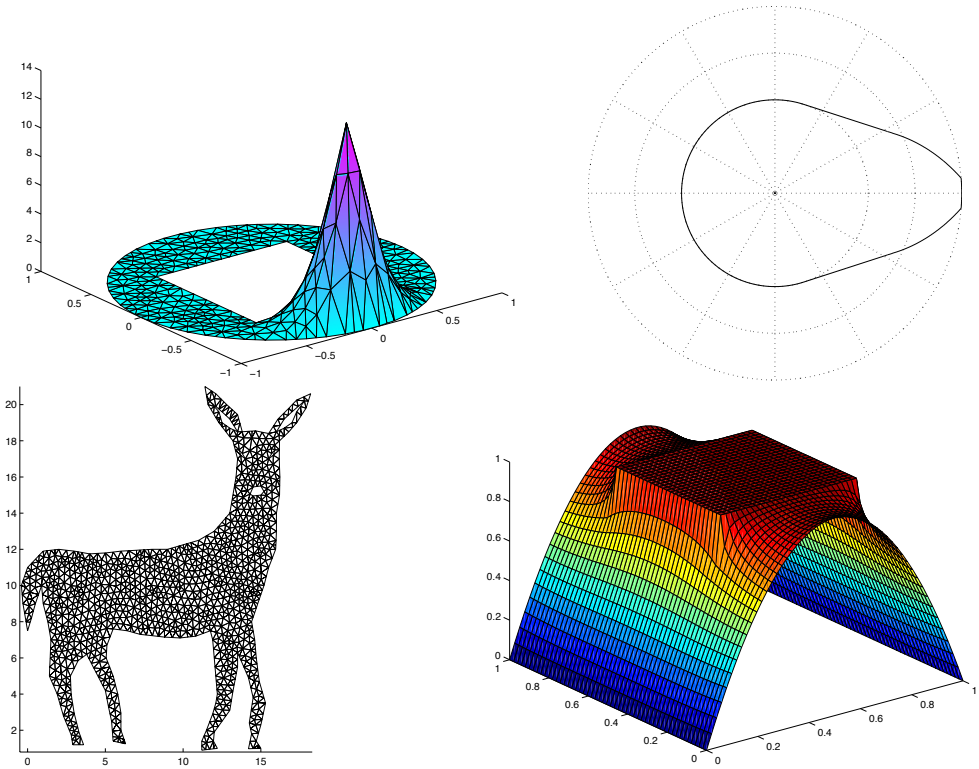

Figure 5.1: Solutions for the Henon (upper left), cam shape (upper right), tetrahedral mesh smoothing (lower left), and minimal surface with obstacle (lower right) COPS problems.

Version 3.0 of COPS has 22 different applications. For each of these applications we use three to five instances of the application obtained by varying a parameter in the application, for example, the number of grid points in a discretization. Table 5.1 gives the quartiles for four problem parameters: the number of variables, the number of equality and inequality constraints, and the total number of constraints (including bounds).

Our benchmarking results are done with a set of problems and solvers. We have used COPS, but we could have also used a selection of the engineering problems provided by Vanderbei [11]. Also note that timing data refers to a particular computing environment

Table 5.1: Problem data (minimum, quartiles, maximum) for the COPS benchmark

| Problem Parameter | min | $q_1$ | $q_2$ | $q_3$ | max |
|---|---|---|---|---|---|
| Variables | 98 | 1146 | 2500 | 4398 | 19241 |
| Equality constraints | 0 | 0 | 100 | 1995 | 7995 |
| Inequality constraints | 0 | 0 | 0 | 41 | 20093 |
| Total constraints | 21 | 1177 | 2402 | 4001 | 20496 |

(machine, compiler, libraries). The usual cautions apply, and thus our conclusions could change if the problems, solvers or computing environment changes. On the other hand, the use of performance profiles tends to minimize the effect of these issues, as noted in [3].

We also note that the solvers for constrained optimization problems invariably have different requirements. Some of the solver use second-order information, while others (for example, MINOS and SNOPT) only use first-order information. The use of second-order information can reduce the number of iterations, but the cost per iteration usually increases. In addition, obtaining second-order information is more costly and may not even be possible. Memory requirements can also play an important role. In particular, solvers that use direct linear equation solvers are often more efficient in terms of computing time, provided there is enough memory. Moreover, some of the solvers are designed for problems with a modest number of degrees of freedom.

The script for generating the timing data sends a problem to each solver successively, so as to minimize the effect of fluctuation in the machine load. The script tracks the wall-clock time from the start of the AMPL process to the end of the solve. Any process that runs 30 minutes is declared unsuccessful. We cycle through all the problems, recording the wall-clock time as well as the combination of AMPL system time (to interpret the model and compute varying amounts of derivative information required by each solver) and solver time for each model variation. We have verified that the AMPL time results we present can be reproduced to within 10 percent accuracy.

## 6 Computational Experiments

We now investigate how performance profiles behave when the convergence test (3.1) is enforced on all the solvers. We also describe some of the computational experiments that we have done with an analyzer that computes the optimality measures in Table 3.1 for optimization problems in the AMPL or GAMS modeling language.

Figure 6.1 displays performance profiles for the two experiments performed with the COPS 3.0 test set [4]. The following solvers were used for the experiments:

KNITRO 3.0 [13], ASL (20020905) LOQO 6.02 [12], ASL (20020221)
MINOS 5.5 [10], ASL (20020614) SNOPT 6.1 [5], ASL (20020614)

All of the computations were performed on an Intel Pentium 4 1.8 GHz CPU with 512 MB of RAM and a 256 KB cache, running Red Hat Linux 7.3. Furthermore, a time limit of 30 minutes was imposed on the solvers for each problem in the test set. A failure is reported when the time limit expires.

The first experiment does not check the optimality measures, trusting the optimization solver to report optimality. In the second experiment, if the convergence test proposed in Section 3,

$$\nu_f(x) \le \tau_f, \quad \nu_s(x, \tau_f) \le \tau_s, \qquad \tau_f = \tau_s = 10^{-6}, \tag{6.1}$$

is not satisfied, then the convergence tolerances used by the solvers are reduced and the problem is re-solved. This procedure is stopped when the tolerances provided to the solver reach $10^{-16}$.

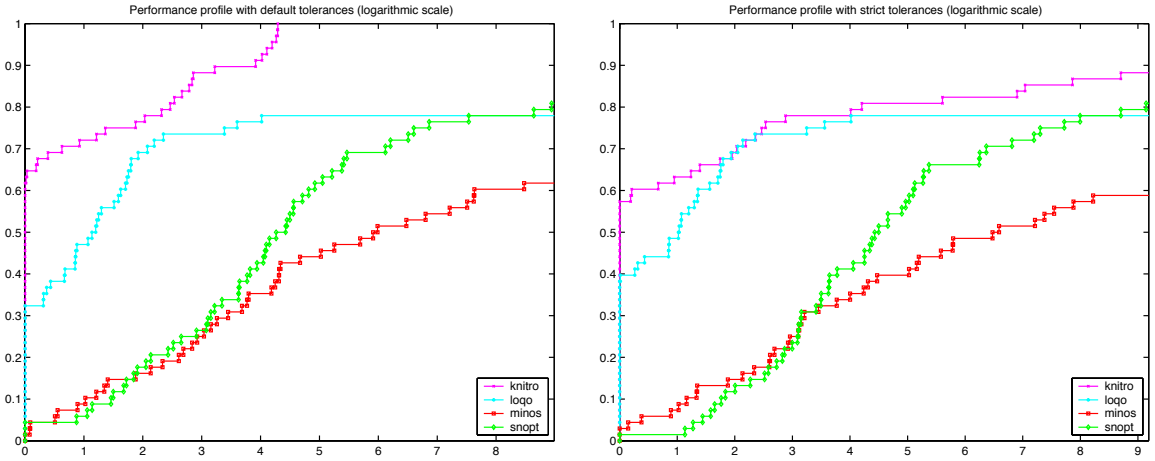


Figure 6.1: Performance profiles ($\log_2$ scale) when no optimality checks are made (left) and when the convergence test (6.1) is enforced (right).

The analyzer computes the optimality measures in Table 3.1 for optimization problems in the AMPL or GAMS modeling language. The main computational task in determining the optimality measures is to compute multipliers by setting up and solving the linear program (2.7) with MINOS. All the data for (2.7) is written to a file with at least fifteen digits of accuracy. The optimality and feasibility tolerances for MINOS are set to $10^{-14}$ when computing the multipliers. In all tests, MINOS indicates that an optimal solution to the linear program was found.

The solvers reported optimal solutions that satisfied the convergence test (6.1) on 109 of the 272 problem instances (40%). In two cases, one of the solvers reported that a nonoptimal solution had been found, when in fact our optimality measures were within the desired tolerances.

The major conclusion that can be drawn from Figure 6.1 is that performance profiles do indeed change when a consistent convergence test is used. The trends in the plots remain

the same, but the magnitude of the differences, especially at the beginning of the plots can change significantly.

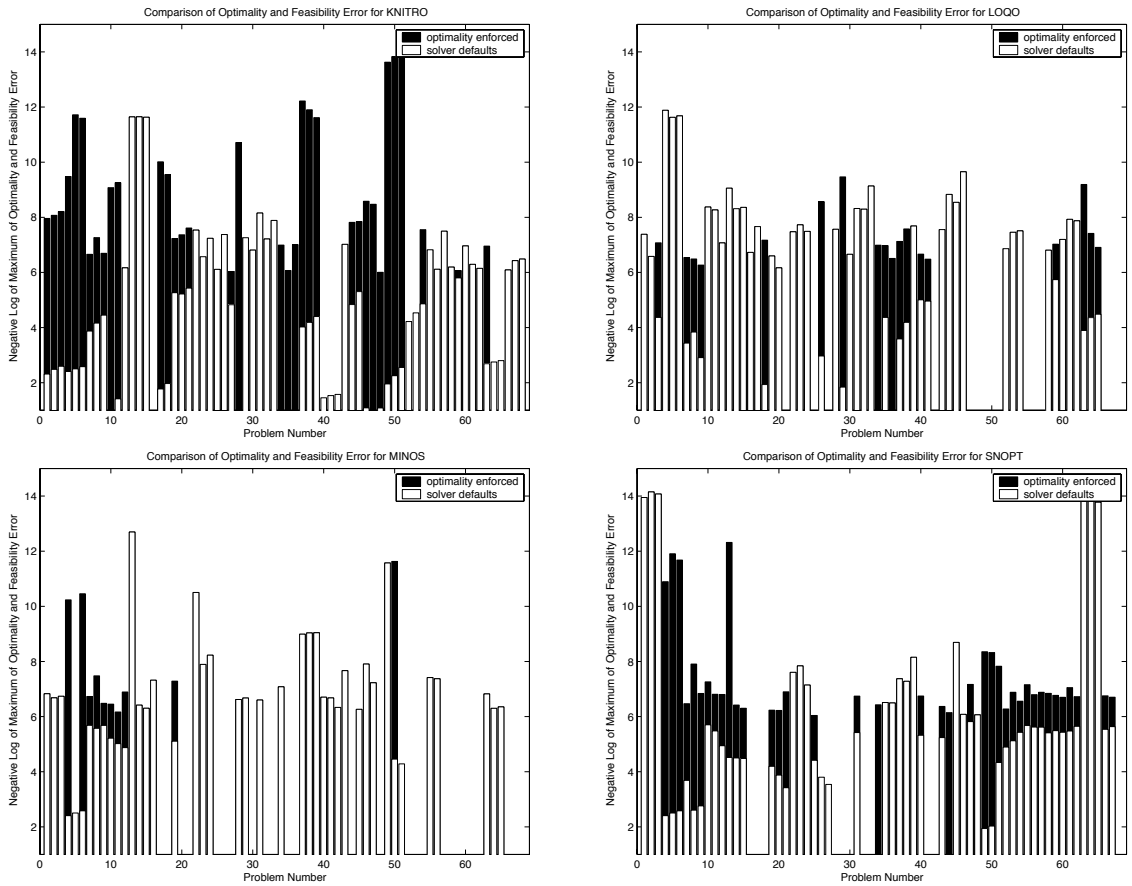


Figure 6.2: Graph of the performance metric (6.2) for KNITRO (upper left), LOQO (upper right), MINOS (lower left), and SNOPT (lower right) when no optimality checks are made (white) and when the convergence test (6.1) is enforced (black).

The effect of convergence criteria on solver performance can also be seen in Figure 6.2. In this figure we plot the performance metric

$$p(x) = -\log_{10}(\max(\nu_f(x), \nu_s(x, \tau_f))) \tag{6.2}$$

for all the solvers. The white bars are for the first experiment where the convergence test (6.1) is not enforced, while the black bars are for the second experiment where the convergence test is enforced.

The height of the bars in Figure 6.2 give the level of accuracy reached. The scaling of the measures $\nu_f$ and $\nu_s$ shows that we can expect to have the performance metric $p(x) \in [0, 16]$ on computations with 16 decimal digits. Problems where $p(x)$ is near zero have not been solved accurately. If $p(x) < 6$, then the convergence test (6.1) is not satisfied.

If only a white bar is shown for a problem, the solver either satisfied the convergence test with the default tolerances and no refinement was needed, or reported a failure. If the solver reports a failure, then the white bar will not reach a tolerance of $\tau_f = \tau_s = 10^{-6}$. In these cases, the iterative reduction in the tolerances was stopped. Those models with no bar present either encountered the time limit imposed during the testing or reported a failure.

If a black bar can be seen in Figure 6.2, then for this problem the default tolerances had to be reduced to meet the convergence test (6.1). As can be seen, all solvers failed to satisfy (6.1) initially in several cases but managed to satisfy the convergence test as the tolerances were reduced.

Figure 6.2 clearly shows that the default convergence test for MINOS tends to agree with (6.1) in most cases. This code was able to satisfy the convergence test with the default tolerances for most of the problems. On the other hand, these results show that KNITRO tended to perform poorly when measured with the metric (6.2).

The main reason why solvers fail to satisfy (6.1) with their default tolerances is that, as noted at the end of Section 3, solvers tend to scale $\tau_f$ and $\tau_s$ based on the size of problem data, for example, $x$, the multipliers $\pi(x)$, or the constraints $c(x)$. This scaling can increase the values of $\tau_f$ and $\tau_s$, and thus lead to a weaker convergence test. Assume, for example, that $\tau_P = \tau_D = 10^{-6}$ in the convergence tests of SNOPT and KNITRO. If $\|x\| = 10^4$ and $\|c(x_0)_+\|_\infty = 10^4$, then both solver set $\tau_f = 10^{-2}$. This explains why some of the white bars are below the $10^{-3}$ level.

Another reason for the large values of $\nu_f$ and $\nu_s$ obtained by the solvers with their default tolerances is that $\nu_f$ and $\nu_s$ examine the accuracy in all of the components, while other measures examine the accuracy in the largest components. Consider, for example, a case where

$$\nabla f(x) = \begin{pmatrix} \alpha \\ 1 \\ \vdots \\ 1 \end{pmatrix}, \qquad \nabla c(x)\pi(x) = \begin{pmatrix} \alpha(1+\tau_s) \\ 1+\alpha\tau_s \\ \vdots \\ 1+\alpha\tau_s \end{pmatrix},$$

for some $\alpha \geq 1$. In this case,

$$\|\nabla f(x) - \nabla c(x)\pi(x)\|_\infty \leq \tau_s \min\Big\{\|\nabla f(x)\|_\infty, \|\nabla c(x)\pi(x)\|_\infty\Big\},$$

and thus the relative error between $\nabla f(x)$ and $\nabla c(x)\pi(x)$ is small. However, it is also clear that the relative error between components $2, \ldots, n$ can be large. In fact,

$$\nu_s(x, \tau_f) = \frac{\alpha\tau_s}{2+\alpha\tau_s},$$

and thus the error measured by $\nu_s$ can be arbitrarily close to one.

## 7 Concluding Remarks

We have shown that the convergence test (3.1) is scale invariant when absolute tolerances are set to zero and behaves satisfactorily at any point where the constraints satisfy the Mangasarian-Fromovitz constraint qualification. We have also demonstrated that this test does not need to use the multipliers given by (2.7), but can use the projection of any set of multipliers into the cone (2.5) associated with $\mathcal{A}_\tau(x)$. This approach avoids an explicit test on complementarity.

Our computational experiments have shown that the use of this convergence test on the benchmarking process can have a significant effect on performance profiles. These experiments have also shown that an additional advantage of the measures associated with this convergence test is that solutions of low accuracy will have either $\nu_f$ or $\nu_s$ close to one, while high accuracy solutions will satisfy (6.1) for the appropriate values of $\tau_f$ and $\tau_s$. This is clearly seen in Figure 6.2.

## References


[1] H. Y. Benson, D. F. Shanno, and R. J. Vanderbei, *A comparative study of large-scale nonlinear optimization algorithms*, Tech. Rep. ORFE-01-04, Princeton University, 2001.

[2] E. G. Birgin, R. A. Castillo, and J. M. Martínez, *Numerical comparison of augmented Lagrangian algorithms for nonconvex problems*, preprint, University of São Paulo, 2003.

[3] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Programming, 91 (2002), pp. 201–213.

[4] E. D. Dolan, J. J. Moré, and T. S. Munson, *Benchmarking optimization software with COPS 3.0*, Technical Memorandum ANL/MCS-TM-273, Argonne National Laboratory, Argonne, Illinois, 2004.

[5] P. E. Gill, W. Murray, and M. A. Saunders, *User's guide for SNOPT 5.3: A Fortran package for large-scale nonlinear programming*, Report NA97-5, University of California, San Diego, 1997.

[6] ——, *SNOPT: An algorithm for large-scale constrained optimization*, SIAM J. Optim., 12 (2002), pp. 979–1006.

[7] C.-J. Lin and J. J. Moré, *Newton's method for large bound-constrained optimization problems*, SIAM J. Optim., 9 (1999), pp. 1100–1127.

[8] H. D. Mittlemann and A. Pruesner, *A server for automated performance analysis of benchmarking data*, preprint, Performance World, 2003. Available at `www.gamsworld.org/performance`.

[9] J. L. Morales, J. Nocedal, R. A. Waltz, and H. Liu, *Assesing the potential of interior point methods for nonlinear optimization*, in High Performance Algorithms and Software for Nonlinear Optimization, L. T. Biegler, O. Ghattas, M. Heinkenschloss, and B. Van Bloemen Waanders, eds., Springer-Verlag, 2003, pp. 167–183.

[10] B. A. Murtagh and M. A. Saunders, *MINOS 5.5 user's guide*, Report SOL 83-20R, Stanford University, 1983, revised July 1998.

[11] R. J. Vanderbei, *Nonlinear optimization models.* See `www.sor.princeton.edu/~rvdb/ampl/nlmodels`.

[12] ——, *LOQO user's manual – Version 4.05*, 2000.

[13] R. Waltz and J. Nocedal, *KNITRO user's manual – Version 3.1*, Tech. Rep. 5, Northwestern University, Evanston, 2003.

The submitted manuscript has been created by the University of Chicago as Operator of Argonne National Laboratory ("Argonne") under Contract No. W-31-109-ENG-38 with the U.S. Department of Energy. The U.S. Government retains for itself, and others acting on its behalf, a paid-up, nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.