A Worst-Case Example Using Linesearch Methods for Numerical Optimization with Inexact Gradient Evaluations.

Richard G. Carter

December 1991

Argonne Laboratory Preprint MCS-P283-1291

A Worst-Case Example Using Linesearch Methods for Numerical Optimization with Inexact Gradient Evaluations. *

Richard G. Carter †

Abstract

Two approaches often used to improve the robustness of numerical optimization algorithms are *linesearch* and *trust region* methods. Trust region methods have previously been shown to be extremely forgiving of high levels of noise and inaccuracy in gradient evaluations. We present a worst-case example demonstrating that linesearch methods can be very fragile with respect to such noise.

1 Introduction

Given the unconstrained minimization problem

$$\text{minimize } f(x) \ , \ f : \Re^n \to \Re \ \ , \tag{1}$$

we consider iterative numerical algorithms such as Newton's method or quasi-Newton methods. These algorithms compute a local quadratic model about a given iterate x_k and generate new iterates $x_{k+1} = x_k + s_k$ using this model. For instance, if the quadratic model is

$$\psi_k(x_k + s) = f(x_k) + g_k^T s + \frac{1}{2} s^T B_k s \tag{2}$$

(where g_k approximates $\nabla f(x_k)$, the gradient of f at x_k, and B_k is a symmetric positive definite matrix approximating $\nabla^2 f(x_k)$, the Hessian of f at x_k), then the simplest quasi-Newton method would be to take $x_{k+1} = x_k + d_k$, where d_k is the step to the minimizer of the quadratic model

$$d_k \equiv -B_k^{-1} g_k. \tag{3}$$

In practice, this approach usually works well when x_0 is sufficiently close to the problem solution, but may not work at all otherwise. To make such methods more robust they are usually modified by the imposition of either a *linesearch* or a *trust region* methodology.

*This research was supported by the Army Research Office under grant DAALO3-89-C-0038.

†Army High Performance Computing Research Center, Institute of Technology, University of Minnesota, 1100 Washington Avenue South, Minneapolis, Minnesota 55415

In linesearch methods, rather than taking $x_{k+1} = x_k + d_k$, we instead set $x_{k+1} = x_k + \alpha_k d_k$ for some positive α_k. The steplength α_k is computed by a one-dimensional search procedure (hence the name *linesearch*) that attempts to approximately minimize $f(x_k + \alpha_k d_k)$ for positive α_k. A broad variety of search procedures and step acceptance criteria have been proposed; a good introduction to these techniques can be found in [4]. These procedures assume that d_k is a strict descent direction for f (trivially true if g_k is exact and B_k is positive definite) and will iterate until an α_k is found that provides a sufficient reduction in the function. Convergence proofs typically rely on the assumptions that $g_k \equiv \nabla f(x_k)$ and that in the limit the search directions do not become orthogonal to the steepest descent directions $-\nabla f(x_k)$.

Trust region methods do not confine their search path to the single direction d_k. A full description of the trust region approach is beyond the scope of this paper (the reader is referred to [4] and [7]), but we note that it has been well established (see, for instance, [7], [11], [2], and [1]) that trust region methods are extremely robust with respect to gradient errors.

In this paper we present a class of examples demonstrating how linesearch methods can fail if even tiny amounts of error are present in the gradient. Specifically, given any nonzero ζ, we can generate a sufficiently bad example such that even though the relative error in the gradient is less than ζ, the search direction d_0 computed by the linesearch technique at the initial iteration is an ascent direction on the function f. In contrast, convergence results have been shown [2] for trust region methods with ζ in excess of 0.5.

2 A Simple Quadratic Example

Select $\epsilon \in (0, 1)$, and define

$$f(x) \equiv \frac{1}{2} x^T A x \quad , \tag{4}$$

where A is a positive definite matrix with condition number $1/\epsilon^2$ defined by

$$A \equiv \frac{1}{2} \begin{pmatrix} \epsilon + \epsilon^{-1} & \epsilon - \epsilon^{-1} \\ \epsilon - \epsilon^{-1} & \epsilon + \epsilon^{-1} \end{pmatrix} \quad \text{so that} \quad A^{-1} \equiv \frac{1}{2} \begin{pmatrix} \epsilon^{-1} + \epsilon & \epsilon^{-1} - \epsilon \\ \epsilon^{-1} - \epsilon & \epsilon^{-1} + \epsilon \end{pmatrix}. \tag{5}$$

Further select $\gamma \in (0, 1]$, and define

$$x_0 \equiv \frac{\sqrt{2}}{2} \begin{pmatrix} 1 - \gamma\epsilon \\ 1 + \gamma\epsilon \end{pmatrix} \quad , \quad g_0 \equiv \frac{\sqrt{2}}{2} \begin{pmatrix} -\gamma - \epsilon \\ \gamma - \epsilon \end{pmatrix} \quad , \text{ and } \quad B_0 \equiv A. \tag{6}$$

Then we have

$$f(x_0) = \frac{\epsilon}{2}(1 + \gamma^2) \quad , \quad \nabla f(x_0) = \frac{\sqrt{2}}{2} \begin{pmatrix} -\gamma + \epsilon \\ \gamma + \epsilon \end{pmatrix} \quad \text{and} \quad \nabla^2 f(x_0) = A. \tag{7}$$

Using these definitions, one can easily establish that the "ideal" Newton direction is

$$d_0^{\text{Ideal}} \equiv - \nabla^2 f(x_0)^{-1} \nabla f(x_0) = - \frac{\sqrt{2}}{2} \begin{pmatrix} 1 - \gamma\epsilon \\ 1 + \gamma\epsilon \end{pmatrix} = -x_0 \ , \tag{8}$$

hence $x_0 + d_0^{\text{Ideal}}$ is the exact solution. However, the Newton direction actually computed using the inexact gradient g_0 is

$$d_0 \equiv - B_0^{-1} g_0 = - \frac{\sqrt{2}}{2} \begin{pmatrix} -1 - \gamma\epsilon \\ -1 + \gamma\epsilon \end{pmatrix} . \tag{9}$$

Thus, the one-dimensional cross section of f in the direction d_0 is

$$f(x_0 + \alpha d_0) = f(x_0) + \alpha\epsilon(1 - \gamma^2) + \alpha^2 \frac{\epsilon}{2}(1 + \gamma^2). \tag{10}$$

Notice that $\|\nabla f(x_0)\| = \|g_0\|$ and $\|d_0\| = \|d_0^{\text{Ideal}}\|$. More specifically, g_0 is a rotation of $\nabla f(x_0)$ through the angle Θ_g while d_0 is a rotation of d_0^{Ideal} through the angle Θ_s, with

$$\Theta_g = \cos^{-1}\left(\frac{\gamma^2 - \epsilon^2}{\gamma^2 + \epsilon^2}\right) \quad \text{and} \quad \Theta_s = \cos^{-1}\left(\frac{\gamma^2\epsilon^2 - 1}{\gamma^2\epsilon^2 + 1}\right). \tag{11}$$

Two final quantities of interest are the angle between d_0 and the steepest descent direction $-\nabla f(x_0)$:

$$\Theta_d = \cos^{-1}\left(\frac{\epsilon^2(\gamma^2 - 1)}{(1 + \gamma^2\epsilon^2)^{1/2}(\epsilon^4 + \gamma^2\epsilon^2)^{1/2}}\right) \tag{12}$$

and the relative error in the gradient:

$$\zeta \equiv \frac{\|g_0 - \nabla f(x_0)\|}{\|g_0\|} = \frac{2\epsilon}{(\gamma^2 + \epsilon^2)^{1/2}}. \tag{13}$$

Given these equations, we can generate a number of interesting examples by selecting appropriate values of ϵ and γ. Consider, for instance, $\gamma = 1$. First note that for *any* choice of ϵ, Θ_d is $\frac{\pi}{2}$ so that d_0 is orthogonal to the steepest descent direction. Hence the directional derivative of f

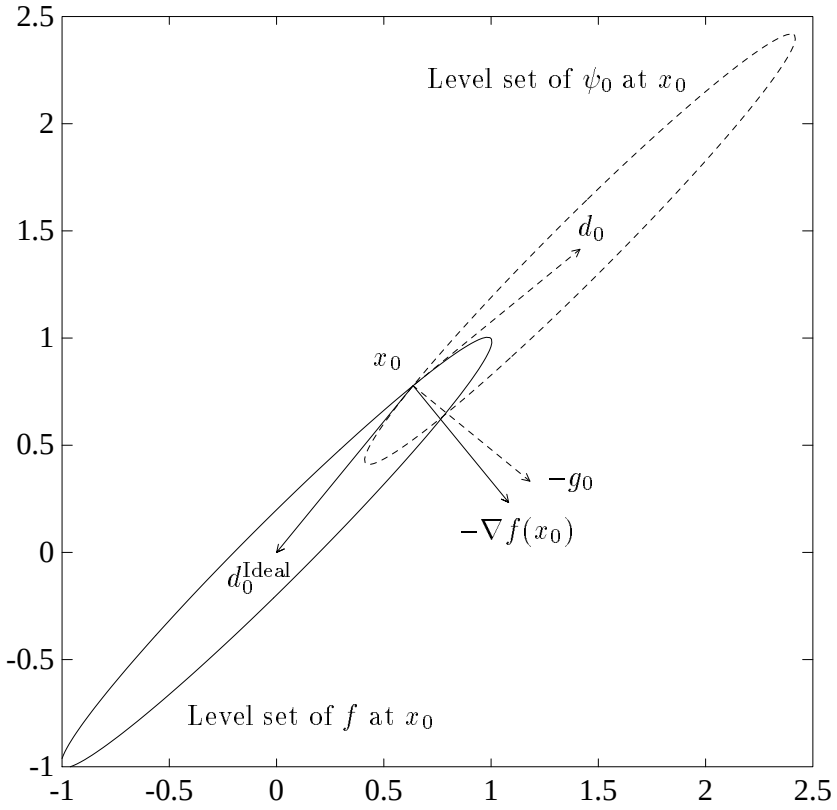

Figure 1: Level sets of f and ψ_0 with $\gamma = 1$, $\epsilon = 0.1$

in the direction d_0 is zero even though the computed directional derivative $g_0^T d_0/\|d_0\|$ is negative. Moreover, for *any* $\alpha \neq 0$, we have $f(x_0 + \alpha d_0) > f(x_0)$, so descent can never be achieved in the d_0 direction even if nonpositive α values are allowed in the linesearch.

Figure 1 shows the level sets of both f and ψ_0 at x_0 for $\gamma = 1$ and $\epsilon = 10^{-1}$, along with the true and approximate gradients and the computed and ideal Newton steps. We see that a very slight change in the gradient approximation results in a dramatic change in the search direction d_0.

If we select $\epsilon = 10^{-3}$, simple calculations show that the relative error in g_0 is only 0.2% and that the angle between g_0 and $\nabla f(x_0)$ is just over 0.1 degrees. Yet not only is d_0 orthogonal to $\nabla f(x_0)$, but it is also almost diametrically opposite in direction from d_0^{Ideal}! *

As ϵ approaches zero, the matrix A becomes successively more ill-conditioned, and errors in g_0 are magnified by successively greater factors when used in the computation of the search direction d_0, as the following table shows.

*In fact, for $\gamma = 1$ we can write $\Theta_s = \pi - \Theta_g$ and $\frac{\Theta_s}{\Theta_g} = \frac{\pi}{\Theta_g} - 1$.

Table 1. Behavior of d_0 as A becomes ill-conditioned, with $\gamma = 1$

ϵ	ζ	Θ_g (degrees)	Θ_s (degrees)	$\frac{\Theta_s}{\Theta_g}$
10^{-1}	0.199	11.4	168.579	0.148D+02
10^{-2}	0.200D-01	1.15	178.854	0.156D+03
10^{-3}	0.200D-02	0.115	179.885	0.157D+04
10^{-4}	0.200D-03	0.115D-01	179.989	0.157D+05
10^{-5}	0.200D-04	0.115D-02	179.999	0.157D+06
10^{-6}	0.200D-05	0.115D-03	180.000	0.157D+07

We see that for an even moderately ill-conditioned matrix, a small error in g_0 can result in a search direction d_0 almost diametrically opposite the correct search direction d_0^{Ideal}. For all of these examples with $\gamma = 1$, there exists no value of α for which descent in f is achieved; hence any linesearch will necessarily fail.

Even more pathological examples can be generated by selecting different values of γ. For instance, if $\gamma = \frac{1}{2}$, then Θ_d is greater than 90 degrees, and d_0 is actually a strict *ascent* direction on f. That is, the directional derivative of f in the d_k direction, $\nabla f(x_0)^T d_0/\|d_0\|$, is strictly positive (even though the computed directional derivative, $g_0^T d_0/\|d_0\|$, is strictly negative). Again, this is true even for very small values of relative gradient error. Investigation of such examples is left as an exercise for the reader.

3 Conclusions

Much debate has occurred over the years between advocates of linesearch techniques and devotees of trust region methodologies. Some arguments center upon issues such as simplicity or elegance and are therefore unanswerable. Other issues are more tangible: linesearch codes are often cited as superior with respect to scale invariance and with respect to linear-algebra-cost-per-iteration, while trust region methods are often regarded as superior for nonconvex problems. However, trust region methods can be made scale invariant with the proper preconditioning [7] and can be implemented if desired with very inexpensive linear algebra (e.g. [8],[3], [10]), while modified linesearch techniques exist which at least partially address the issue of negative curvature (e.g. [6],[5]). In the opinion

of the author, none of the arguments raised in the past (with the possible exception of the issue of nonconvexity) are overwhelmingly convincing to objective observers.

In contrast the situation for once seems quite clear if one considers the relative merits of trust region versus linesearch approaches using the criteria of robustness with respect to inexact gradients. Robust convergence results have been shown for trust region algorithms even in the presence of significant gradient error. However, when computing a search direction in a linesearch method, a very small amount of error in the computed gradient may result in a computed search direction almost diametrically opposite the desired direction. This search direction may be orthogonal to the steepest descent direction or may even be a strict ascent direction on f. Linesearch techniques will invariably fail if this occurs. While the worst-case examples presented here represent particularly poor combinations of starting point and gradient error, *linesearch algorithms are nevertheless in principle highly vulnerable to the slightest inaccuracies or noise in gradient evaluations.* A numerical comparison of actual performance of linesearch versus trust region approaches in the presence of gradient noise for a limited number of test problems can be found in [9].

References

[1] R. Carter, *Numerical experience with a class of algorithms for nonlinear optimization using inexact function and gradient information*, Tech. Report 89-46, Institute for Computer Applications in Science and Engineering, 1989. (SIAM J. Sci. Statist. Comput., to appear).

[2] ——, *On the global convergence of trust region algorithms using inexact gradient information*, SIAM J.Numer. Anal., 28 (1991), pp. 251–265.

[3] J. Dennis Jr. and H. Mei, *Two new unconstrained optimization algorithms which use function and gradient values*, J. Optim. Theory Appl., 28 (1979), pp. 453–482.

[4] J. Dennis Jr. and R. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[5] E. E. Eskow and R. Schnabel, *Software for a new modified Choleski factorization*, Tech. Report CU-CS-443-89, Dept. of Computer Science, University of Colorado at Boulder, 1989.

[6] P. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, 1981.

[7] J. Moré, *Recent developments in algorithms and software for trust region methods*, in Mathematical Programming: State of the Art, A. Bachem, M. Grőtschel, and B.Korte, eds., Springer-Verlag, Berlin, 1983, pp. 258–287.

[8] M. Powell, *A hybrid method for nonlinear equations*, in Numerical Methods for Nonlinear Algebraic Equations, P. Rabinowitz, ed., Gordon and Breach, London, 1970, pp. 87–114.

[9] G. Shubin and P. Frank, *A comparison of the implicit gradient approach and the variational approach to aerodynamic design optimization*, Tech. Report AMS-TR-163, Applied Mathematics and Statistics Dept., Boeing Computer Services, 1991.

[10] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[11] Ph. L. Toint, *Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space*, IMA Journal of Numerical Analysis, 8 (1988), pp. 231–252.