AMO - Advanced Modeling and Optimization, Volume 11, Number 1, 2009

Global Convergence using De-linked Goldstein or Wolfe Linesearch Conditions

Bruce Christianson
School of Computer Science
University of Hertfordshire, Hatfield AL10 9AB, England
b.christianson@herts.ac.uk

Abstract.
Goldstein or Wolfe conditions can be imposed on a linesearch to ensure convergence of an iterative nonlinear optimization algorithm to a stationary point. However it is actually not necessary to find a single step which satisfies both Goldstein (or both Wolfe) conditions simultaneously in order to ensure global convergence. De-linking the conditions can make it significantly easier to find an acceptable stepsize, which is neither too short nor too long. Although this fact has been known for a long time, the practice seems to have fallen out of fashion. However In this note we give a short, self-contained proof of global convergence for de-linked Goldstein and Wolfe conditions, and advocate their use. In particular, we argue that the increasingly widespread availability of second order adjoints via Automatic Differentiation tools means that the cost of a conventional safe line search is often unacceptably high for algorithms such as Truncated Newton. The de-linked approach advocated here is used with the Goldstein conditions in the OPTIMA Truncated Newton code.

Keywords: Truncated Newton; Line-search; Wolfe conditions; Goldstein conditions.

1 The Goldstein Conditions

The Goldstein conditions [6] for a stepsize a are

$$(G1) \quad f(x_k + as_k) < f(x_k) + m_1 a(s_k, g_k)$$
$$(G2) \quad f(x_k + as_k) > f(x_k) + m_2 a(s_k, g_k)$$

Here f is the function being minimized, x_k is the current search position, s_k is a search direction returned by the direction finder starting at x_k, a is a stepsize found by the linesearch, $g_k = f'(x_k)$, (s_k, g_k) denotes inner product, and m_1, m_2 are constants with $0 < m_1 < m_2 < 1$. Usually $m_1 < 0.5$ and $m_2 = 1 - m_1$. Note that $(s_k, g_k) < 0$ iff s_k is a descent direction. It is well known that, under suitable conditions (for example, those set out below) choosing a_k to satisfy both G1 and G2 and setting $x_{k+1} = x_k + a_k s_k$ as the new search position suffices to ensure convergence of $||g_k||$ to zero. However such global convergence results can also be established without requiring the two conditions both to hold for a single value of a. Results of this kind were once extensively used, but seem to have fallen out of fashion in recent decades.

Theorem 1
Suppose that f is bounded below and has Lipshitz continuous derivative f' on the basin $y : f(y) \leq f(x_0)$. Let $R > 1$ be a positive constant.
At each optimization step k suppose that s_k is chosen to be a descent direction, so that $(s_k, g_k) < 0$, and a_k, b_k are chosen with $0 < b_k < Ra_k$ and such that the stepsize a_k satisfies G1 and the stepsize b_k satisfies G2. Set $x_{k+1} = x_k + a_k s_k$.
Then either $g_k = 0$ for some k or else

$$\sum_{k=0}^{\infty} [\cos^2 \theta_k \cdot ||g_k||^2] < \infty$$

where θ_k is the angle between s_k and g_k.
Proof
Let m be $\min[m_1, 1 - m_2]$. Then by G2 we have

$$f(x_k + b_k s_k) - f(x_k) > (1 - m) b_k (s_k, g_k)$$

whence by the MVT we have, for some c with $0 < c < b_k$

$$s_k \cdot f'(x_k + c s_k) > (1 - m)(s_k, g_k)$$

so

$$s_k \cdot [f'(x_k + c s_k) - f'(x_k)] > -m||s_k||\ ||g_k|| \cos \theta_k.$$

Let K be a Lipshitz constant for f'. Then $||f'(x_k + c s_k) - f'(x_k)|| < cK\ ||s_k||$ and so

$$cK||s_k||^2 > -m||s_k||\ ||g_k|| \cos \theta_k.$$

Also $c < b_k < Ra_k$ so

$$a_k ||s_k|| > -(m/RK)||g_k|| \cos \theta_k.$$

Note that both sides are positive, because s_k is a descent direction so $\cos \theta_k < 0$. Meanwhile from G1 we have

$$f(x_k) - f(x_k + a_k s_k) > -m a_k (s_k, g_k) = -m a_k ||s_k|| ||g_k|| \cos \theta_k$$
$$> (m^2/RK)||g_k||^2 \cos^2 \theta_k > 0$$

so the $f(x_k)$ are monotone decreasing and bounded below by L say (since f is bounded below by hypothesis) whence

$$f(x_0) - L \geq \sum_{k=0}^{\infty} [f(x_k) - f(x_{k+1})] > (m^2/RK) \sum_{k=0}^{\infty} [\cos^2 \theta_k ||g_k||^2].$$

QED
In particular, if the s_k are chosen so that $\cos \theta_k$ is bounded away from zero, then $\sum ||g_k||^2$ converges, whence $||g_k||$ tends to zero faster than $\sqrt{n}$.

Note that we have not proved that x_k converges in norm, but the basin $y : f(y) \leq f(x_0)$ is compact, so a subsequence of x_k will be norm convergent to x_* with $f'(x_*) = 0$.

2 The Wolfe Conditions

The Wolfe conditions [8] may be used with the linesearch instead of the Goldstein conditions. The Wolfe conditions are:

$$(W1) \quad f(x_k + as_k) < f(x_k) + m_1 a(s_k, g_k)$$

$$(W2) \quad s_k \cdot f'(x_k + as_k) > m_2(s_k, g_k)$$

As with the Goldstein conditions, m_1, m_2 are constants with $0 < m_1 < m_2 < 1$ and usually $m_1 < 0.5, m_2 = 1 - m_1$. Note that W1 is the same as G1.

Corollary 2
Under the conditions of Theorem 1, if we choose a_k, b_k to satisfy W1, W2 instead of G1, G2 respectively then the same conclusion holds.
Proof
Let m be $\min[m_1, 1 - m_2]$ and proceed as in the proof of Theorem 1. Instead of applying the MVT to G2 use W2 directly to get

$$s_k \cdot f'(x_k + b_k s_k) > (1 - m)(s_k, g_k)$$

then proceed as before with b_k in place of c.
QED

Note that although in practice b_k is usually chosen so that $b_k \geq a_k$ this condition is not required by either proof.

3 A Simple Linesearch

Use of the de-linked Goldstein conditions allows a very simple and quick linesearch to be implemented.

Algorithm 3 (De-linked Goldstein).
Let a be the initial stepsize (usually $a = 1$) and let $R > 1$ be a positive constant. Now perform

```
b:=a;
while not G1(a) do b:=a; a := b/R enddo;
while not G2(b) do a:=b; b := R*a enddo;
```

The first iteration must terminate because f is differentiable at x_k. The second iteration must terminate because f is bounded below. Since if G1 is false then G2 is true, we have the postcondition that a satisfies G1 and b satisfies G2, and either $b = a$ or $b = Ra$. Note that the two `while` loops can be placed in either order, and that at most one of them will ever be performed.

The OPTIMA Truncated Newton code [5] has from its initial implementation used this approach, with $R = 2$, to modify a suggested step, based on a quadratic model, in such a way as to ensure that it is of appropriate length for the non-quadratic objective function f, while maintaining a guarantee of global convergence. We now explain the arguments for preferring this linesearch approach for use with Truncated Newton.

First, recall that it is not worthwhile for the linesearch to minimize f accurately along the line in direction s_k through x_k. To perform such an *exact* linesearch would be computationally very expensive, and would not help to minimize f.

To see that the extra cost of an exact search may not be very beneficial in terms of overall convergence, recall that if the quadratic approximation to f at x_k is good for the proposed step s_k then

$$f(x_k + s_k) \approx f(x_k) + \frac{1}{2}(s_k, g_k)$$

so conditions G1 and G2 will both hold for $a = 1$. Thus, if we are adjusting the stepsize at all, then the quadratic approximation is not good for the step we are considering. Since Truncated Newton is a second order (i.e. quadratic) algorithm, we therefore need to be prepared to move to a point where the Hessian, and hence the directional Hessian in the search direction itself, may be significantly different. Hence it is generally better, when we are far from the solution, to make the line search satisfy a global convergence condition (such as de-linked Goldstein) *with as little computational effort as possible* and to invest the effort in subsequent TN iterations.

The increasing availability of tools, such as the NAGWare Fortran compiler, which support higher-order Automatic Differentiation has proved valuable in making accurate and computationally inexpensive gradient and Hessian information more readily available for use in calculating good search directions. However a side-effect is that function calls in line searches become, in relative terms, much more computationally expensive than before. With the use of second order reverse or adjoint mode Automatic Differentiation [2, 7], a directional second derivative can be accurately evaluated for the same computational cost as about six function evaluations, irrespective of the number of independent variables (i.e. regardless of the size of x.) Thus an exact linesearch may turn out to cost significantly more than an entire inner TN iteration. Far away from the minimum, where gradients are not small and quadratic models are not good, the number of inner iterations per outer iteration is usually not great, even for large scale problems, and so linesearches which require many function evaluations can easily become a significant proportion of the total computational cost.

From these two considerations it follows that, if we cannot accept an initially proposed step, it is desirable to find an acceptable steplength using as few trial points as possible. Satisfying global convergence conditions rapidly and moving on allows

computational effort to be invested in TN iterations rather than in linesearches.

The disadvantages of exact line searches are well-known and most minimization algorithms rely on a version of Theorem 1 (or Corollary 2) with $a_k = b_k$ and use a so-called weak search which seeks to satisfy both G1 and G2 (or W1 and W2). Most of these weak searches are of the backtracking type, originally proposed by Armijo for modified steepest descent [1, Corollary 2]. Armijo's algorithm omits any iteration to enforce G2, and instead ensures global convergence by imposing an *a priori* condition upon the choice of initial step. The (often unmet) requirement to prove that a particular direction finder satisfies such a condition thus represents a theoretical obstacle to promiscuous use of a backtracking linesearch. Failure to take a sufficiently long step when far from the solution is also a well-known practical problem.

Use of the de-linked Goldstein conditions in the manner set out in Algorithm 3 resolves both of these problems. Furthermore it typically requires fewer trial points than an approach that attempts to satisfy the Goldstein or Wolfe conditions jointly. For example, compare Algorithm 3 with the following algorithm, taken from [4, Section 3.4].

Algorithm 4 (Conventional Wolfe).
Let $R > 1$ and $0 < r \leq 0.5$ be positive constants. Let t be the initial stepsize (usually $t = 1$) and perform

```
a:=0; b:= ∞;
while not (a=b) do
 if W1(t) then a:=t else b:=t endif;
 if W1(t) and W2(t) then b:=t
 else
  choose a new t with a < t < b;
  if b = ∞ then t := max (t, R*a)
  else t:= max ((1-r)*a + r*b, min (t, r*a + (1-r)*b))
  endif
 endif
enddo
```

For the first iteration, t is chosen to be the initial stepsize. In later iterations, t may be chosen by polynomial extrapolation or interpolation, but must be adjusted to ensure first that b eventually becomes finite, and subsequently that $b - a$ decreases geometrically with each iteration. The length $b - a$ decreases by a factor of $1 - r$ in the worst case, but even in the best case the factor is at least r, which can be unfortunate when t is close to a.

It can be shown [4, Theorem 3.7] that this iteration will always terminate after a finite number of steps, so that a will satisfy both Wolfe conditions.

Remember that we are not seeking to minimize the function exactly along the line, but merely to satisfy the Wolfe conditions jointly, and for this purpose Algorithm 4

is about the most efficient that may be devised. The advantages of using a de-linked test as in Algorithm 3 are clear.

We prefer de-linked Goldstein to de-linked Wolfe for use with Truncated Newton: this is because G2 for some b_k with $b_k < Ra_k$ implies W2 for some c_k with $c_k < b_k < Ra_k$ by the MVT. Far away from the minimum, it is usually better to err on the side of taking a longer step which satisfies G1.

4 Conclusions

This paper re-states and clarifies a useful fact which seems to have been overlooked in recent years – namely that the line search in a minimization algorithm need not cause the new point to satisfy both Goldstein conditions G1 and G2 (or both Wolfe conditions W1 and W2). Instead these conditions can be de-linked and an acceptable step can be found using the simple and inexpensive Algorithm 3. The possibility of using a de-linked stopping rule is implicit in the well-known and often cited paper by Armijo [1]. However, although his linesearch algorithm ensures that the stepsize is not too long, it relies upon the search direction finder to ensure that the stepsize is not too short. This means that a backtracking linesearch cannot be combined with an arbitrary direction finder, whereas Algorithm 3 can be.

Algorithm 3 has been proposed several times before, the 1988 paper by Dixon & Price [5] being just one example. However more recent texts such as [4] continue to suggest more expensive searches (e.g. Algorithm 4 above) which require the steplength to satisfy the conditions jointly. A simple line search similar to Algorithm 3 is outlined in [3, Chapter 8]; but this is followed by the cautionary remark that additional steps are needed to ensure that the final point satisfies stopping tests for both loops.

For these reasons, it seems appropriate once again to draw the theoretical and practical attractions of using a de-linked approach to the attention of a wider readership.

References

[1] L. Armijo, Minimization of Functions having Lipschitz Continuous First Partial Derivatives, Pacific Journal of Mathematics, 16(1), 1–3, 1966.

[2] M. Bartholomew-Biggs, S. Brown, B. Christianson and L. Dixon, Automatic Differentiation of Algorithms, Journal of Computational and Applied Mathematics, 124 171–190, 2000.

[3] M. Bartholomew-Biggs, Nonlinear Optimization with Engineering Applications, Springer, 2008.

[4] J.F. Bonnans, J.C. Gilbert, C. Lamarechal and C.A. Sagastizabel, Optimisation Numerique, Springer, Berlin, 1997.

[5] L.C.W. Dixon and R.C. Price, Numerical Experience with the Truncated Newton Method for Unconstrained Optimization, Journal of Optimization Theory and Applications, 56, 245–255, 1988.

[6] A.A. Goldstein, Constructive Real Analysis, Harper and Row, London, 1967.

[7] U. Naumann, M. Maier, J. Riehme, D. Gendler and B. Christianson, Compiler-Generated First- and Second-Order Adjoints for Matrix-Free Unconstrained Nonlinear Optimization, International Journal of Applied Mathematics and Computer Science, 2009, to appear.

[8] P. Wolfe, Convergence Conditions for Ascent Methods, SIAM Review, 11, 226–235, 1969 and 13, 185–188, 1971.