A note on robust descent in differentiable optimization*

Jean-Pierre Dussault†

November 5, 2015

Abstract

In this note, we recall two solutions to alleviate the catastrophic cancellations that occur when comparing function values in descent algorithms. The automatic finite differencing approach [4] was shown useful to trust region and line search variants. The main original contribution is to successfully adapt the line search strategy [6] for use within trust region like algorithms.

Keywords: Unconstrained nonlinear optimization, descent algorithms, numerical accuracy

Introduction

We consider descent algorithms for solving

$$\min_{x\in\mathbb{R}} f(x) \tag{1}$$

where it is understood that we are seeking a *local* minimum of f, and that f is $\mathcal{C}^2(\mathbb{R}^n)$.

Two main algorithm categories have been developed for differentiable unconstrained optimization: line search methods and trust region algorithms.

In line search methods, a descent direction $d : \nabla f(x)d < 0$ and a step t are computed ensuring $f(x + td) < f(x)$. Actually, more is required than plain descent, for instance the so-called strong Wolfe conditions. Sometimes, a simple Armijo backtracking procedure is applied to ensure plain descent using a sufficient descent d.

In trust region methods (or the ARC$_q$variant), a quadratic model $q_x(d) = f(x)+\nabla f(x)d+\frac{1}{2}d^t\nabla^2 f(x)d$ is used to compute a direction ensuring $q_x(d) < q_x(0) = f(x)$. Then, the algorithm is built using the ratio

$$r = \frac{\Delta f(x)}{\Delta q_x(d)} \stackrel{\text{def}}{=} \frac{f(x+d) - f(x)}{q_x(d) - q_x(0)}.$$

*This research was partially supported by NSERC grant OGP0005491

†Professeur titulaire, département d'Informatique, Université de Sherbrooke, Sherbrooke (Québec), Canada J1K 2R1. e-mail: Jean-Pierre.Dussault@USherbrooke.ca. This work was completed while in sabbatical leave at INSA, Rennes

In both approaches, descent must be assessed, i.e. $f(x+d) < f(x)$ is enforced. Non-monotone approaches have been devised to ensure descent periodically instead of at every step. Descent has to be enforced as well in those non-monotone variants.

Since $f(x+d) - f(x) \sim \mathcal{O}(\|d\|^2)$, when d is smaller than the square root of the machine precision, usually because x is close to a minimizer or sometimes for some other reason, $\Delta f(x)$ may not be reliably computed, being the order of the machine precision obtained by subtracting two close quantities. This phenomenon is refereed to as catastrophic cancellation.

This phenomenon was observed in [4] and [6] and two different strategies were developed to alleviate this catastrophic cancellation. In [4], automatic finite differentiation was used to clean the descent test, an approach useful and tested both for trust region algorithms and line search variants. In [6], a clever reformulation of the descent test was used to improve the reliability of their line search procedure.

We present in this note the adaptation of the reformulation of [6] for trust region like algorithms.

1 Reformulation of the descent

The descent condition $f(x+d) - f(x) < 0$ may be expressed using the function $\phi(\alpha) = f(x+\alpha d)$ as $\phi(\alpha) - \phi(0)$. When used in the Armijo criterion, we need to assess that $\phi(\alpha) - \phi(0) \leq \delta\alpha\phi'(0)$. When ϕ is quadratic (f is quadratic), $\phi(\alpha) = q(\alpha) = \frac{1}{2}a_2\alpha^2 + a_1\alpha + a_0$ with $a_2 = d^t\nabla^2 f(x)d$, $a_1 = \nabla f(x)d$ and $a_0 = f(x)$. The Armijo condition may be rewritten $\frac{1}{2}a_2\alpha + a_1 \leq \delta a_1$ which yields $(\frac{1}{2}a_2\alpha + a_1) + \frac{1}{2}a_1 \leq \delta a_1$, i.e. as expressed in [6],

$$a_2\alpha + a_1 = q'(\alpha) \leq (2\delta - 1)q'(0) = (2\delta - 1)a_1.$$

Equivalently, we may write $\alpha\frac{q'(\alpha)+q'(0)}{2} \leq \delta\alpha q'(0)$. Thus, for a quadratic function f, the descent condition may be expressed as $f(x+d) - f(x) = \frac{q'(1)+q'(0)}{2} < 0$.

The descent condition when f is not quadratic may be approximated by the formula

$$0 > f(x+d) - f(x) \approx \frac{\nabla f(x+d)d + \nabla f(x)d}{2} \tag{2}$$

and the approximation should be accurate when d is small.

1.1 Accuracy of formula (2)

Since the successive derivatives of order p of ϕ are $\mathcal{O}(\|d\|^p)$, assuming $f \in \mathcal{C}^3(\mathbb{R}^n)$, we may write the Taylor expansion $\phi(\alpha) = q(\alpha) + \mathcal{O}(\alpha\|d\|)^3$. Therefore,

$$\phi(\alpha) - \phi(0) = q(\alpha) - q(0) + \mathcal{O}(\|\alpha d\|^3) = \alpha\frac{q'(\alpha) + q'(0)}{2} + \mathcal{O}(\|\alpha d\|^3).$$

Now, $\phi'(0) = q'(0)$ and $\phi'(\alpha) = \nabla f(x+\alpha d)d$ while $q'(\alpha) = (\nabla f(x) + \alpha d^t\nabla^2 f(x))d$ and

$$\nabla f(x+\alpha d) = \nabla f(x) + d^t\nabla^2 f(x) + \mathcal{O}(\|\alpha d\|^2)$$

so that $\phi'(\alpha) = q'(\alpha) + \mathcal{O}(\|\alpha d\|^3)$ and

$$\alpha\frac{\phi'(\alpha)+\phi'(0)}{2} = \alpha\frac{q'(\alpha)+q'(0)}{2} + \mathcal{O}(\|\alpha d\|^3).$$

Therefore,

$$\left|(f(x+d)-f(x)) - \left(\frac{\nabla f(x+d)d + \nabla f(x)d}{2}\right)\right| = \mathcal{O}(\|d\|^3).$$

As stressed in [6], the formula $\frac{\nabla f(x+d)d+\nabla f(x)d}{2}$ is only an approximation of $f(x+d)-f(x)$ but since $(f(x+d)-f(x)) = \mathcal{O}(\|d\|^2)$ while its error is $\mathcal{O}(\|d\|^3)$, the approximation is asymptotically accurate. Moreover, when d becomes small, roughly of the order of the square root of the machine precision, the expression of the true value suffers from catastrophic cancellations but formula (2) still can be evaluated accurately.

The use of automatic finite differencing as advocated in [4] is preferable since its precision is not dependent on d becoming small; on the down side, it involves elaborate software development while the approximation just described is readily and easily added to any existing code.

2 Implementation within a trust region framework

The actual use of formula involves trade off related to the magnitude of d with respect to the machine precision. Inspired by [6], we recommend to use the alternate formula when Δf becomes small, $|\Delta f(x)| \leq \epsilon |f(x)|$ and when Δq becomes small, $\Delta q(x) \leq \epsilon$. The resulting algorithm is only slightly modified, as seen on algorithm 1.

```
Model_Algorithm(x, β, f, m)
  { Given: x; }
  {    objective function f; }
  {    model m; }
  {    initial value for β. }
  repeat
    d ← Solve_Model(m, x, β)
    Δf ← f(x) − f(x + d)
        { Here is the numerical switch to the approximate formula }
        if (Δq < ε ∨ |Δf| ≤ ε|f(x)| ) then
            Δf ← (∇f(x)d + ∇f(x + d)d)/2
    Δq ← q(0) − q(d)
    r ← Δf/Δq
    if (r < 0.25 ) then β ← β/2   {Unsuccessful}
    else
        x ← x + d                 {Successful}
        if (r > 0.75 ) then
            β ← 2 ∗ β             {Very successful}
  until ( termination_criterion)
  Result ← x
```

Algorithm 1: Trust region

3 Numerical observations

The improvement resulting from the use of formula (2) is apparent on specific instances where the catastrophic cancellations indeed result in bad behavior. On most instances, the behavior of the improved variant is exactly the same as the vanilla plain version. Critical instances may benefit from the improved version either by reaching a tighter tolerance, or by reducing the iteration count to reach the same stopping criterion. We present observations to illustrate different cases. The examples are taken from the Lukšan *et al* collection [7] of scalable versions (some are modified) of CUTEst problems. The problem's dimensions are $n = 100$ and the stopping criterion for all examples is $\|\nabla f(x)\| \leq 10^{-8}$. This is a quite tight stopping criterion used to illustrate the difficulties for 100-dimensional problems. For larger instances (1024), cancellations occur when aiming to a 10^{-6} precision and for even larger instances, cancellation occurs for even looser stopping criteria.

The switching ϵ in algorithm 1 is $10^4 \epsilon_{machine}$ and our machine precision is $\epsilon_{machine} = 2.220446049250313 \times 10^{-16}$.

The variant for solving the model is ARC_qK described in details in [5], a new implementation of a variant ARC_q [3] of ARC [1, 2].

3.1 arwhead: failure avoided

This first example does not reach the 10^{-8} stopping criterion since numerical cancellation force the direct descent formula to return 0.0 while the improved variant achieves the prescribed tolerance this very iteration.

Δf	formula (2)	$\|d\|$
0.0	$3.243324790241839 \times 10^{-17}$	$2.3019020478444516 \times 10^{-9}$

3.2 curly: failure avoided

For the curly problem, three iterations with improved formula are observed. The last ratio for the improved formula is 0.9999999942893429, a very good agreement with the prediction.

Δf	formula (2)	$\|d\|$	$\Delta f/\Delta q$	$\Delta f(2)/\Delta q$
2.04×10^{-8}	2.03×10^{-8}	0.00022	1.00	0.99
3.63×10^{-12}	4.06×10^{-12}	1.13×10^{-6}	0.89	0.99
0.0	2.05×10^{-14}	2.31×10^{-7}	0.0	0.99

3.3 errinros-mod: converged faster

For the problem errinros-mod, the improved variant required 273 functions, 185 gradients and 2284 products hessian/vector evaluations while the vanilla plain version required 280 functions, 185 gradients and 2291 products hessian/vector evaluations. Both variants achieved the stopping criterion $\|\nabla f(x)\| \leq 10^{-8}$.

Conclusion

Numerical evaluation of descent is prone to numerical cancellations. The modified formula derived from [6] and adapted here to trust region algorithms proves useful to alleviate this cancellation phenomenon. Improved descent evaluation should be part of any robust implementation.

References

[1] Coralia Cartis, Nicholas I.M. Gould, and Philippe L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Mathematical Programming*, 127(2):245–295, 2011.

[2] Coralia Cartis, Nicholas I.M. Gould, and Philippe L. Toint. Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function- and derivative-evaluation complexity. *Mathematical Programming*, 130(2):295–319, 2011.

[3] Jean-Pierre Dussault. Simple unified convergence proofs for trust region and a new ARC variant. Technical report, Université de Sherbrooke, 2015. http://www.optimization-online.org/DB_FILE/2015/06/4939.pdf.

[4] Jean-Pierre Dussault and Benoit Hamelin. Robust descent in differentiable optimization using automatic finite differences. *Optimization Methods and Software*, 21(5):769–777, 2006.

[5] Jean-Pierre Dussault and Dominique Orban. A scalable implementation of adaptative cubic regularization methods using shifted linear systems. Technical report, GERAD G-2015-109, 2015.

[6] William W. Hager and Hongchao Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on Optimization*, 16(1):170–192, 2005.

[7] Ladislav Lukšan, Ctirad Matonoha, and Jan Vlček. Modified cute problems for sparse unconstrained optimization. Technical report, Institute of Computer Science, 2010.