New York University

Master's Thesis

A Large-Scale Constrained Optimizer for NonSmooth Functions

Author:
Wilmer Henao

Supervisor:
Michael L. Overton

A thesis submitted in fulfilment of the requirements for the degree of Master of Science in Scientific Computing

in the

Courant Institute of Mathematical Sciences
Department of Mathematics

November 2013

Declaration of Authorship

I, Wilmer Henao, declare that this thesis titled, 'A Large-Scale Constrained Optimizer for NonSmooth Functions' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

NEW YORK UNIVERSITY

Abstract

Faculty Name
Department of Mathematics

Master of Science in Scientific Computing

A Large-Scale Constrained Optimizer for NonSmooth Functions

by Wilmer Henao

The Thesis Abstract is written here ...

Acknowledgements

I would like to than my advisor Michael Overton for all the hours of hard work and for all the great recommendations and changes that suggested for the creation of this thesis.

I also would like to thank Allan Kaku and Anders Skaaja for earlier contributions to other versions of the code.. . .

Contents

List of Figures

List of Tables

Abbreviations

LAH	**L**ist **A**bbreviations **H**ere

Physical Constants

Speed of Light $c = 2.997\ 924\ 58 \times 10^{8}\ \mathrm{ms}^{-\mathrm{S}}$ (exact)

Symbols

a	distance	m
P	power	W (Js^{-1})
ω	angular frequency	$rads^{-1}$

For/Dedicated to/To my...

Chapter 1

Introduction

The most important goal in this thesis is to find a solution of the minimization problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & l_i \le x_i \le u_i, \\ & i = 1, \ldots, n. \end{aligned} \tag{1.1}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$. And l_i and $u_i \in \mathbb{R}$

In the particular case when $n = 0$, the problem would be called "unconstrained" and several techniques have already been developed to handle this type of problems [1]. In our particular cases n is supposed to be finite but very large, so storing and calculating a Hessian matrix is prohibitively expensive. In this thesis $f(x)$ is a nonsmooth function, that is, the function $f(x)$ itself is continuous but its gradient $\nabla f(x)$ is not.

For the particular case when n is a small number, several methods have been developed in order to solve optimization problems of nondifferentiable functions in lower dimensions [2]. In the case of smooth functions, it is possible to use Newton iteration algorithms and achieve quadratic convergence, the problem with Newton algorithms is that they require second derivatives to be provided[1]. In the 1950's and several years after that, several quasi-newton methods were proposed where the second derivative Hessian matrix is "approximated" step by step [1]. These approximations or "updates" are updated after every iteration and the way in which this update is calculated defines a new method depending on the particular needs. This thesis will only be concerned with the *BFGS*. [2] which can achieve super linear convergence, has proven to work in most practical

[1]the main issue with the second derivative is that it requires a total of $n \times n$ partial derivatives. Which is totally impractical for medium and for some small-size problems

[2]BFGS stands for the last names of its authors Broyden, Fletcher, Goldfarb and Shanno

purposes and posseses very nice self correcting features [3], in other words, it doesn't matter that one update incorrectly estimates the curvature in the objective function, It will always correct itself in just a few steps. This self-correcting property is very desired in the nonsmooth case, since changes in curvature could be large near the optimal point. $BFGS$ is not the only update method available, there are many more updates typically used. In particular we have the SR1(Symmetric Rank-1), which has the problem that it may not maintain a positive definiteness of the Hessian in the case when f is convex, although it otherwise generates "good Hessian approximations"[4] Another update worth mentioning is the DFP[3] which together with BFGS spans the Broyden class of updates.

Finally, this thesis will also assume that the Hessian matrix is not sparse. In this case, there are other algorithms that may be more suitable [5, 6], some of them have even been implemented in fortran [7].

In this sense, chapter 2 will extend on $BFGS$ and on the basic reasons and steps to move into a large scale $L-BFGS$. During the solution of an unconstrained problem, a template version of $C++$ code was created which enhances the code originally written by Allan Kaku and Anders Skaaja [8] while maintaining Allan's high-precision libraries. A link to this $BFGSL-BFGS$ software is provided in the thesis website

The next chapter is chapter 3 and this chapter introduces the algorithm by Nocedal, and what changes were introduced into the code in order to produce a better and stable version in the nonsmooth case. The website also contains a $FORTRAN$ implementation derived from the original implementation by Nocedal.

[3]DFP also stands for its authors Davidon, Fletcher and Powell. Davidon is credited with the first quasi-newton algorithm while he was working at Argonne National Laboratories

Chapter 2

BFGS and L-BFGS and implementations in the unconstrained case

The *BFGS* method is a line search method. In this sense it is very important to understand where it comes from.

2.1 Line Search Methods:

Line search methods are iterative methods where every step the researcher has to decide a direction to move or "search direction" and also, how much to move in that direction or "step length". In general all line search methods are characterized by the equation

$$x_{k+1} = x_k + \alpha_k d_k \tag{2.1}$$

where α_k is the step length and d_k is the search direction.

The main difference between line search methods will be in the selection of α_k and d_k and depending on this selection we will be able to achieve a convergence that will typically be somewhere between linear and quadratic.

In general it is desirable that the search direction is a descent direction in every step. So one usually also checks that the property $d_k^T \nabla f_k = |d_k||\nabla f_k| \cos\theta < 0$ is satisfied.

In general the search direction will have the form

$$\mathrm{d}_k = -B_k^{-1}\nabla f_k \tag{2.2}$$

And the choice of matrix B_k will define what type of method we are using. This implies that together with the search direction condition with $d_k^T \nabla f_k < 0$ we need that:

$$\text{-}\nabla f_k B_k^{-1} \nabla f_k < 0 \tag{2.3}$$

And this would require that B_k be positive definite.

Other conditions that are of the utmost importance in this thesis are the conditions on step length. These conditions are so important in this thesis that they will be reviewed in a separate chapter.

Next a presentation of some of the most important line search methods

2.1.1 Steepest Descent

The most naive choice of search direction is the negative of the gradient $\nabla f_k(x)$, which is equivalent to choosing $B_k = I$. This method is called steepest descent and its main advantage is obviously that it only requires ∇f_k as an input.

If the condition of the function's Hessian matrix is low, then steepest descent will converge very fast, but if this condition is high, the method will take a lot of iterations. In the worst cases, steepest descent can show a zig-zagging pattern that slows down convergence.

2.1.2 Convergence of search direction methods

Finally there is the question of convergence and for the case where $f(x)$ is a smooth function, line search convergence is guaranteed for the steepest descent method via Zoutendijk's theorem [9] in [**?**]. This theorem quantifies the impact of the step length α_k on convergence and guarantees convergence for steepest descent. It also, produces a set of criteria for other methods so that they too are globally convergent as long as they don't deviate too far away from steepest descent. Unfortunately there is no such theorem for the nonsmooth case, and we are only sure of convergence from practical results.

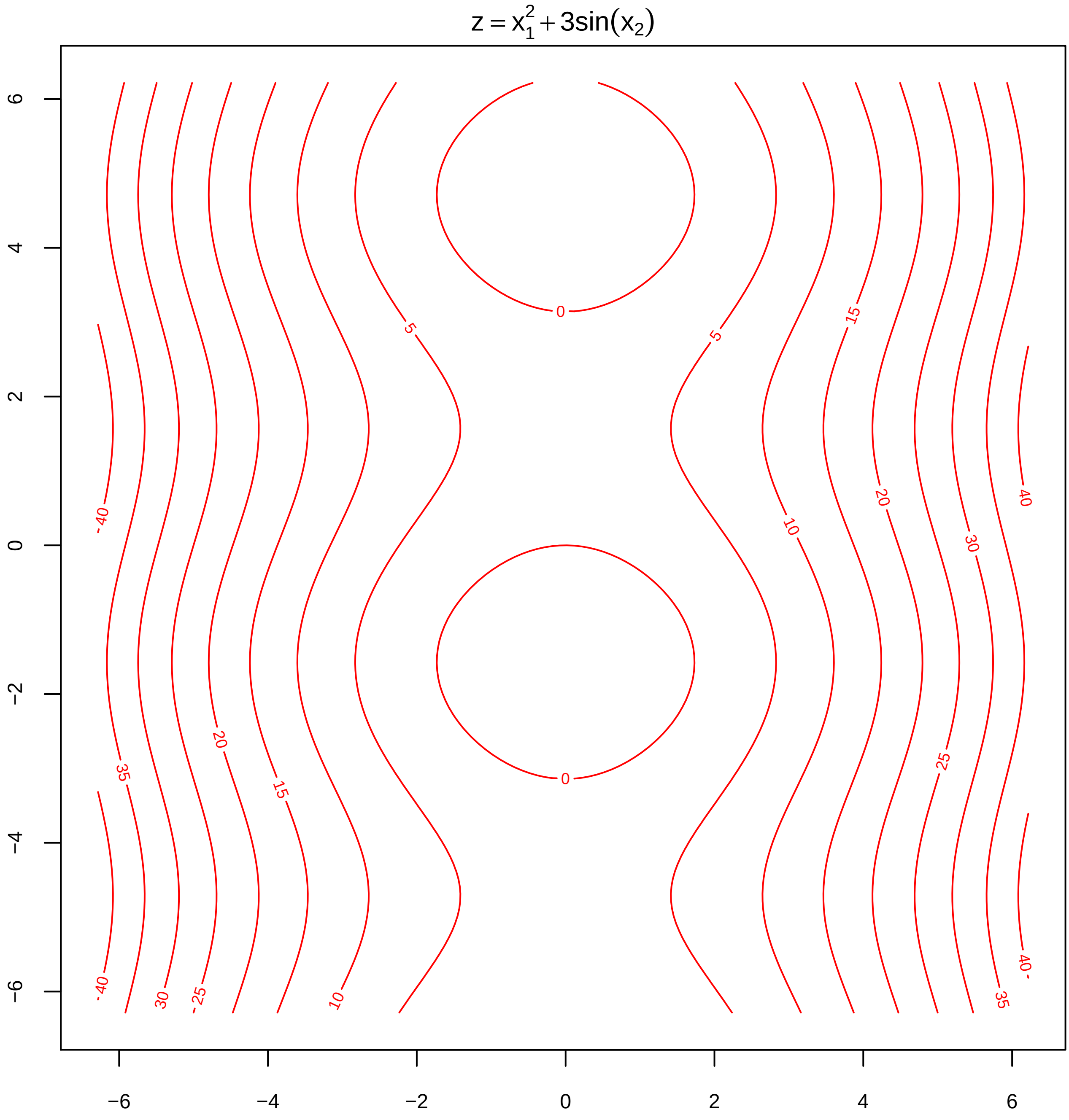

FIGURE 2.1: Steepest Descent

Chapter 3

Constraints and L-BFGS-B: Algorithm and implementations

Appendix A

Appendix Title Here

Write your Appendix content here.

Bibliography

[1] J.E. Dennis and R.B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice Hall, SIAM Publications, Englewood Cliffs, NJ, 1983, 1993.

[2] Krzysztof C. Kiwiel. *Methods of Descent for Nondiferentiable Optimization.* Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1985.

[3] Jorge Nocedal. Theory of algorithms for unconstrained optimization. *Acta Numerica*, 1:199–242, January 1992.

[4] Stephen J. Wright Jorge Nocedal. *Numerical Optimization.* Springer Series in Operations Research, 2nd edition, 2006.

[5] Roger Fletcher, Andreas Grothey, and Sven Leyffer. Computing sparse hessian and jacobian approximations with optimal hereditary properties. Technical report, Large-Scale Optimization with Applications, Part II: Optimal Design and Control, 1996.

[6] R. Fletcher. An optimal positive definite update for sparse hessian matrices. *SIAM Journal on Optimization*, 5:192–218, 1995.

[7] Prof. Dr. Ph. L. Toint Dr. A. R. Conn, Dr. N. I. M. Gould. *LANCELOT: A fortran package for large-scale nonlinear optimization (Release A).* Springer Series in Computational Mathematics, Berlin Heidelberg, 1992.

[8] Allan Kaku. Implementation of high-precision arithmetic in the bfgs method for nonsmooth optimization. Master's thesis, New York University, Courant Institute of Mathematical Sciences, 251 Mercer Street, New York, NY 10012, 2011.

[9] G. Zoutendjik. *Integer and Nonlinear Programming*, chapter Nonlinear Programming, computational methods, pages 37–86. North-Holland Publishing company, Amsterdam-London, 1970.