How *Grossone* Can Be Helpful to Iteratively Compute Negative Curvature Directions

Renato De Leone[1], Giovanni Fasano[2](✉), Massimo Roma[3], and Yaroslav D. Sergeyev[4]

[1] Scuola di Scienze e Tecnologie, Università di Camerino, Camerino, Italy
renato.deleone@unicam.it

[2] Dipartimento di Management, Università Ca' Foscari Venezia, Venice, Italy
fasano@unive.it

[3] Dipartimento di Ingegneria Informatica, Automatica e Gestionale 'A. Ruberti', SAPIENZA, Università di Roma, Rome, Italy
roma@dis.uniroma1.it

[4] Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica, Università della Calabria, Rende, Italy
yaro@dimes.unical.it

Abstract. We consider an iterative computation of negative curvature directions, in large scale optimization frameworks. We show that to the latter purpose, borrowing the ideas in [1,3] and [4], we can fruitfully pair the Conjugate Gradient (CG) method with a recently introduced numerical approach involving the use of grossone [5]. In particular, though in principle the CG method is well-posed only on positive definite linear systems, the use of grossone can enhance the performance of the CG, allowing the computation of negative curvature directions, too. The overall method in our proposal significantly generalizes the theory proposed for [1] and [3], and straightforwardly allows the use of a CG-based method on indefinite Newton's equations.

Keywords: Negative curvature directions
Second order necessary optimality conditions · Grossone
Conjugate gradient method

1 Introduction

When considering the solution of the unconstrained optimization problem

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

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a nonlinear smooth function and n is large, specific methods should be applied in case stationary points satisfying second order necessary optimality conditions are sought. Moreover, additional cares on the algorithms

R. Battiti et al. (Eds.): LION 12 2018, LNCS 11353, pp. 180–183, 2019.
https://doi.org/10.1007/978-3-030-05348-2_16

adopted are definitely mandatory, since standard stationarity conditions in general do not ensure the convexity of the objective function in a neighborhood of critical points. In this regard, the computation of negative curvature directions for the objective function is an essential tool, to guarantee convergence to second order points.

Observe that convergence towards second order stationary points, where the Hessian matrix is positive semidefinite, requires to efficiently explore the eigenspaces of the Hessian matrix at the solution. Thus, the optimization method adopted to solve (1) should be able to cope also with nonconvexities of the objective function. In particular, as showed in [3], this can be accomplished by suitable Krylov-subspace methods, such that at each iteration j, a pair of directions (s_j, d_j) is computed satisfying specific properties. The vector s_j must be a direction which approximately solves Newton's equation $\nabla^2 f(x_j)\ s = -\nabla f(x_j)$ at x_j. Its purpose is essentially to ensure convergence to stationary points. On the other hand, the direction d_j is a so called *negative curvature direction*, which is used to possibly force convergence to stationary points which satisfy second order necessary optimality conditions [9]. This implies that the sequence $\{d_j\}$ is expected to satisfy the next assumption (which might be possibly weakened).

Assumption 1. *The vectors in the sequence* $\{d_j\}$ *are bounded and*

$$d_j^T \nabla^2 f(x_j) d_j < 0$$

$$d_j^T \nabla^2 f(x_j) d_j \longrightarrow 0 \qquad \Longrightarrow \qquad \min\left[0,\ \lambda^{\,min}(\nabla^2 f(x_j))\right] \longrightarrow 0,$$

being $\lambda^{\,min}(\nabla^2 f(x_j))$ *the smallest eigenvalue of the Hessian matrix* $\nabla^2 f(x_j)$.

Roughly speaking, the conditions in Assumption 1 imply that the negative curvature directions $\{d_j\}$ need to eventually approximate an eigenvector associated with the smallest negative eigenvalue of the Hessian matrix. In [3] indications on the computation of the pair (s_j, d_j) were given, though the computation of d_j involved the use of Planar-CG methods, which impose an heavy computation in case some (so called) planar iterations are performed. This approach proved to be effective but required a complex analysis involving different articulated subcases. Here, we aim at describing a strong simplification in the computation the directions $\{d_j\}$, by using a novel approach which extends some ideas in [4].

2 Our Proposal

To this purpose, we use the Krylov-subspace method in [1], for indefinite Newton's equations. This method alternates standard CG iterations and *planar* iterations, each of them being equivalent to a double CG iteration. This method is used to satisfy the next lemma (the proof follows from Theorem 3.2 in [3]).

Lemma 1. *Given problem (1), suppose at iteration* n *(being* n *the space dimension) of the Krylov-subspace method used to solve Newton's equation* $\nabla^2 f(x_j)s = \nabla f(x_j)$*, the decompositions*

$$\nabla^2 f(x_j) R_j = R_j T_j, \qquad T_j = L_j B_j L_j^T$$

are available, where $R_j \in \mathbb{R}^{n\times n}$ is orthogonal, $T_j \in \mathbb{R}^{n\times n}$ has the same eigenvalues of $\nabla^2 f(x_j)$, with at least one negative eigenvalue, and $L_j, B_j \in \mathbb{R}^{n\times n}$ are nonsingular. Let z be the unit eigenvector corresponding to the smallest eigenvalue of B_j, and $\bar{y} \in \mathbb{R}^n$ be the bounded solution of the linear system $L_j^T y = z$. Then, the vector $d_j = R_j\bar{y}$ satisfies Assumption 1.

The main drawback of the latter approach is that the eigenvector z of B_j and the solution of the linear system $L_n^T y = z$ should be of easy computation, which is hardly guaranteed uniquely using the instruments in [1] and [3]. To fill this gap, let us consider the following matrices, obtained applying the method in [1]:

$$
L_n = \left(\begin{array}{ccccccccc}
1 & & & & & & & & \\
-\sqrt{\beta_1} & \ddots & & & & & & & \\
 & \ddots & 1 & & & & & & \\
 & & -\sqrt{\beta_{k-1}} & 1 & & & & 0 & \\
 & & & 0 & 1 & & & & \\
 & & & -\sqrt{\beta_k\beta_{k+1}} & 0 & 1 & & & \\
 & 0 & & & & -\sqrt{\beta_{k+2}} & \ddots & & \\
 & & & & & & \ddots & 1 & \\
 & & & & & & & -\sqrt{\beta_{n-1}} & 1
\end{array}\right),
$$

$$
B_n = \left(\begin{array}{ccccccc}
1/a_1 & & & & & & \\
 & \ddots & & & & 0 & \\
 & & 1/a_{k-1} & & & & \\
 & & & 0 & \sqrt{\beta_k} & & \\
 & & & \sqrt{\beta_k} & e_{k+1} & & \\
 & 0 & & & & \ddots & \\
 & & & & & & 1/a_n
\end{array}\right),
$$

where $\{a_i\}$, $\{\beta_i\}$, e_{k+1} are suitable scalars, and we assume (for the sake of simplicity) that the method performed all CG iterations, with the exception of only one planar iteration (namely the k-th iteration - see [1] and [2]). Then, our novel approach proposes to introduce the numeral *grossone*, as in [5–8], and follow some guidelines from [4], so that we can compute the lower block triangular matrix

$$
\bar{L}_n = \left(\begin{array}{ccc|c|ccc}
1 & & & & & & \\
-\sqrt{\beta_1} & \ddots & & & & & \\
 & \ddots & 1 & & & & \\
\hline
 & & -\sqrt{\beta_{k-1}} & V_k C_k^{-1} & & & \\
 & & 0 & & & & \\
\hline
 & & & \begin{pmatrix} -\dfrac{\beta_k\sqrt{\beta_{k+1}}}{\sqrt{\beta_k+\lambda_k^2}} & -\dfrac{\beta_k\sqrt{\beta_{k+1}}}{\sqrt{\beta_k+\lambda_{k+1}^2}} \end{pmatrix} C_k^{-1} & 1 & & \\
 & 0 & & & -\sqrt{\beta_{k+2}} & \ddots & \\
 & & & & & \ddots & 1 \\
 & & & & & -\sqrt{\beta_{n-1}} & 1
\end{array}\right)
$$

and the diagonal matrix

$$\bar{B}_n = \begin{pmatrix} 1/a_1 & & & & & & \\ & \ddots & & & & & \\ & & 1/a_{k-1} & & & 0 & \\ & & & \frac{1}{\alpha_k s\textcircled{1}} & & & \\ & 0 & & & \frac{s\textcircled{1}}{\alpha_{k+1}} & & \\ & & & & & \ddots & \\ & & & & & & 1/a_n \end{pmatrix},$$

such that $L_n B_n L_n^T = \bar{L}_n \bar{B}_n \bar{L}_n^T$ and the symbol ① indicates *grossone*. Moreover,

$$C_k = \begin{pmatrix} 1/\sqrt{\lambda_k \alpha_k s\textcircled{1}} & 0 \\ 0 & \sqrt{s\textcircled{1}/(\lambda_{k+1}\alpha_{k+1})} \end{pmatrix} \in \mathbb{R}^{2\times 2},$$

being λ_k, λ_{k+1} the two eigenvalues (with $\lambda_k \lambda_{k+1} < 0$) of the 2×2 matrix

$$\begin{pmatrix} 0 & \sqrt{\beta_k} \\ \sqrt{\beta_k} & e_{k+1} \end{pmatrix}, \tag{2}$$

and the columns of the orthogonal matrix $V_k \in \mathbb{R}^{2\times 2}$ correspond to the normalized eigenvectors of the matrix in (2). It is shown in [10] that the latter arrangement can easily allow the computation of the sequence $\{d_j\}$ of negative curvature directions complying with Assumption 1 and Lemma 1.

References

1. Fasano, G.: Conjugate Gradient (CG)-type method for the solution of Newton's equation within optimization frameworks. Optim. Methods Softw. **19**(3–4), 267–290 (2004)
2. Fasano, G.: Planar-conjugate gradient algorithm for large scale unconstrained optimization, part 1: theory. J. Optim. Theory Appl. **125**(3), 523–541 (2005)
3. Fasano, G., Roma, M.: Iterative computation of negative curvature directions in large scale optimization. Comput. Optim. Appl. **38**(1), 81–104 (2007)
4. De Leone, R., Fasano, G., Sergeyev, Y.D.: Planar methods and grossone for the Conjugate Gradient breakdown in nonlinear programming. Comput. Optim. Appl. **71**(1), 73–93 (2018)
5. Sergeyev, Y.D.: Numerical infinities and infinitesimals: methodology, applications, and repercussions on two Hilbert problems. EMS Surv. Math. Sci. **4**, 219–320 (2017)
6. De Leone, R.: Nonlinear programming and grossone: quadratic programming and the role of constraint qualifications. Appl. Math. Comput. **318**, 290–297 (2018)
7. Gaudioso, M., Giallombardo, G., Mukhametzhanov, M.: Numerical infinitesimals in a variable metric method for convex nonsmooth optimization. Appl. Math. Comput. **318**, 312–320 (2018)
8. Sergeyev, Y.D., Kvasov, D.E., Mukhametzhanov, M.: On strong homogeneity of a class of global optimization algorithms working with infinite and infinitesimal scales. Commun. Nonlinear Sci. Numer. Simul. **59**, 319–330 (2018)
9. Moré, J., Sorensen, D.: On the use of directions of negative curvature in a modified Newton method. Math. Program. **16**, 1–20 (1979)
10. De Leone, R., Fasano, G., Roma, M., Sergeyev, Y.D.: Iterative ①-based computation of negative curvature directions in large scale optimization (submitted)