# QuickeNing: A Generic Quasi-Newton Algorithm for Faster Gradient-Based Optimization

**Hongzhou Lin** * **Julien Mairal** * **Zaid Harchaoui** †
firstname.lastname@inria.fr

## Abstract

We propose a technique to accelerate gradient-based optimization algorithms by giving them the ability to exploit L-BFGS heuristics. Our scheme is (i) generic and can be applied to a large class of first-order algorithms; (ii) it is compatible with composite objectives, meaning that it may provide exactly sparse solutions when a sparsity-inducing regularization is involved; (iii) it admits a linear convergence rate for strongly-convex problems; (iv) it is easy to use and it does not require any line search. Our work is inspired in part by the Catalyst meta-algorithm [15], which accelerates gradient-based techniques in the sense of Nesterov; here, we adopt a different strategy based on L-BFGS rules to learn and exploit the local curvature. In most practical cases, we observe significant improvements over Catalyst for solving large-scale high-dimensional machine learning problems.

## 1 Introduction

Convex composite optimization arises in many scientific fields, such as image and signal processing, or machine learning. It consists of minimizing a real-valued function composed of two convex terms:

$$\min_{x \in \mathbb{R}^d} \Big\{ f(x) \triangleq f_0(x) + \psi(x) \Big\}, \tag{1}$$

where $f_0$ is smooth with Lipschitz continuous gradient, and $\psi$ is not necessarily differentiable. To solve (1), significant efforts have been devoted to (i) extending techniques for smooth optimization to deal with composite terms [2, 19]; (ii) exploiting the underlying structure of the problem—is $f$ a finite sum of independent terms [1, 5, 6, 17, 24, 25]? is $\psi$ separable in different blocks of coordinate [21, 22, 26]? (iii) exploiting the local curvature of the objective to achieve faster convergence than gradient-based approaches when the dimension $d$ is very large [16, 20]. Yet, solving all these problems at the same time remains challenging: this is precisely the focus of this paper.

To tackle (1), first-order methods are often used, but it is also known that Quasi-Newton approaches are sometimes very effective in the smooth case [24]. Since the dimension $d$ is large, limited-memory variants such as L-BFGS are necessary to achieve high scalability [16, 20]. The theoretical guarantees offered by L-BFGS are somewhat weak, meaning that it does not outperform first-order methods in terms of worst-case convergence rate. Yet, it remains one of the greatest practical success of smooth optimization, and adapting it to composite and structured problems is of utmost importance.

For instance, proximal Quasi-Newton methods have been proposed [3, 13], but they typically require computing many times the proximal operator of $\psi$ with respect to a non-isotropic metric, which may be as computationally demanding as solving the original problem. More related to our work, L-BFGS is combined with SVRG for minimizing smooth finite sums in [11]. The goal of our paper is more general since it is not limited to SVRG, but it can be applied to a large-class of first-order techniques for composite optimization, including for instance other incremental algorithms [5, 6, 17, 24, 25].

More precisely, our main contribution is a generic meta-algorithm which consists of applying a modified L-BFGS scheme with inexact (but accurate enough) gradients to the Moreau-Yosida

*Thoth team, Inria Grenoble, Laboratoire Jean Kuntzmann, CNRS, Univ. Grenoble Alpes, France.
†University of Washington, Seattle, WA 98195, USA.

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

regularization of the objective. The resulting approach turns out to be (i) generic, as stated previously; (ii) despite the smoothing of the objective, the sub-problems that we solve are composite ones, which may lead to exactly sparse iterates when a sparsity-inducing regularization is involved; (iii) it admits a worst-case linear convergence rate for strongly-convex problems, which is typically the best obtained for L-BFGS schemes in the literature; (iv) it is easy to use and does not require any line search algorithm, which is sometimes the computational bottleneck of classical Quasi-Newton methods.

The idea of combining second order or quasi-Newton methods with Moreau-Yosida regularization is in fact relatively old [4, 9, 10, 18]. Our approach revisits this principle with a limited-memory variant (to deal with large dimension $d$), with an alternative strategy to line search schemes (which is useful when $f$ is a large sum of $n$ functions), and with a global complexity analysis that is more relevant than focusing on convergence rates regardless of the cost per iteration.

## 2 The Moreau-Yosida Regularization

The Moreau-Yosida regularization of the objective is defined as

$$F(x) = \min_{z \in \mathbb{R}^p} \left\{ f(z) + \frac{\kappa}{2} \|z - x\|^2 \right\}, \tag{2}$$

where $\kappa > 0$. The image $p(x)$ of $x$ under the proximal mapping is defined as the solution of (2). The Moreau-Yosida regularization admits classical properties, which we present below; see also [14].

**Proposition 1 (Basic properties of the Moreau-Yosida regularization)**

1. *$F$ is convex and minimizing $f$ and $F$ are equivalent in the sense that*

$$\min_{x \in \mathbb{R}^p} F(x) = \min_{x \in \mathbb{R}^p} f(x),$$

*and the solution set of the two above problems coincide with each other.*

2. *$F$ is continuously differentiable even when $f$ is not and*

$$\nabla F(x) = \kappa(x - p(x)),$$

*Moreover the gradient $\nabla F$ is Lipschitz continuous with constant $L_F = \kappa$.*

3. *When $f$ is $\mu$-strongly convex, $F$ is $\mu_F$-strongly convex with constant $\mu_F = \mu\,\kappa/(\mu + \kappa)$ .*

Interestingly, these observations yield a simple strategy for minimizing any convex function $f$, by simply minimizing $F$ with an algorithm that is able to handle *smooth* functions. Such an approach is appealing but it raises several difficulties: computing the gradient of $F$ requires the exact solution $p(x)$ of (2), for which no closed-form is available in general and it is thus necessary to use an approximate solution. This implies defining an inexactness criterion that is easy to check and to control the accuracy of the gradient approximation to ensure convergence; see, e.g. [7]. Catalyst [15] falls in this class of algorithms, applying an accelerated first-order method with inexact gradients to $F$.

## 3 The QuickeNing Algorithm and its Convergence Analysis

In this paper, we consider using L-BFGS with inexact gradients [8], which are computed with Algorithm 1 using a given optimization method $\mathcal{M}$. We remark that the criterion $h(z) - h^* \leq \varepsilon$ is one of the weakest in the literature about inexact gradient-based approaches [7, 12, 23], and it is probably the most useful one: the condition $h(z) - h^* \leq \varepsilon$ can often be checked by computing duality gaps in practice and gradient-based methods often admit convergence rates that allow us to control the computational complexity for solving the sub-problems (3) up to accuracy $\varepsilon$.

**Algorithm 1** Procedure GradientEstimate

**input** Current point $x$ in $\mathbb{R}^p$; accuracy $\varepsilon$; smoothing parameter $\kappa > 0$; optimization method $\mathcal{M}$.

1: Compute the approximate proximal mapping using an optimization method $\mathcal{M}$:

$$z \approx \operatorname*{arg\,min}_{z \in \mathbb{R}^d} \left\{ h(z) \triangleq f(z) + \frac{\kappa}{2} \|z - x\|^2 \right\}, \tag{3}$$

such that $h(z) - h^* \leq \varepsilon$ where $h^* = \min_{z \in \mathbb{R}^d} h(z)$; define $F_a = h(z)$.

2: Compute the approximate gradient of $F$ at $x$: $g = \kappa(x - z)$.

**output** approximate gradient $g$, objective value $F_a$, and proximal mapping $z$.

We present our modified L-BFGS heuristic in Algorithm 2, which provides a positive definite matrix $H$ given two vectors $s$ and $y$ representing the difference of two consecutive iterates and gradients, respectively. The matrix $H$ is not explicitly stored, but formed by a "generating list" of at most $l$ pairs $(s_i, y_i)_{i=1\dots,j}$ using the classical L-BFGS formula, see [20]. Here, the inexactness of the gradient requires changing the L-BFGS rule in order to guarantee the positive definiteness of $H$. This is typically achieved by skipping some L-BFGS updates that would make $H$ non-positive definite [8]. Our convergence analysis suggests a skipping rule, which corresponds to basic strong-convexity and Lipschitz gradient inequalities when using exact gradients with $c_1 = c_2 = 1$. The QuickeNing scheme is then presented in Algorithm 3, and its main properties are discussed below.

**Algorithm 2** Modified L-BFGS update with skipping criterion

**input** current L-BFGS matrix $H$ formed from a generating list $\{(s_i, y_i)\}_{i=1\dots j}$ and initial diagonal matrix $H_0$; new candidate pair $(s, y)$; L-BFGS parameters $0 < c_1, c_2 \leq 1$; memory parameter $l$;

**if** $c_1\mu_F\|s\|^2 \leq y^T s$ and $\frac{c_2}{L_F}\|y\|^2 \leq y^T s$ **then**
  add $(s, y)$ to the generating list, and remove the oldest pair if the cardinal exceeds $l$.
**else**
  keep the generating list unchanged.
**end if**

**output** new matrix $H$ (generating list and $H_0$).

**Algorithm 3** The QuickeNing meta-algorithm

**input** Initial point $x_0$ in $\mathbb{R}^p$; decreasing sequence $(\varepsilon_k)_{k\geq 0}$; number of iterations $K$; smoothing parameter $\kappa > 0$; L-BFGS parameters $0 < c_1, c_2 \leq 1$; optimization method $\mathcal{M}$;

Initialization: $(g_0, F_0, z_0) = \mathsf{GradientEstimate}(x_0, \varepsilon_0)$; BFGS matrix $H_0 = (1/\kappa)I$.
**for** $k = 0, \dots, K-1$ **do**
  Perform the Quasi-Newton step: $x_{\text{test}} = x_k - H_k g_k$.
  Estimate the new gradient and the Moreau-Yosida function value

$$(g_{\text{test}}, F_{\text{test}}, z_{\text{test}}) = \mathsf{GradientEstimate}(x_{\text{test}}, \varepsilon_{k+1}).$$

  **if** sufficient approximate decrease is obtained $F_{\text{test}} \leq F_k - \frac{1}{4\kappa}\|g_k\|^2 + \varepsilon_k$, **then**
    accept the new iterate: $(x_{k+1}, g_{k+1}, F_{k+1}, z_{k+1}) = (x_{\text{test}}, g_{\text{test}}, F_{\text{test}}, z_{\text{test}})$.
  **else**
    update the current iterate with the proximal mapping: $x_{k+1} = z_k$.

$$(g_{k+1}, F_{k+1}, z_{k+1}) = \mathsf{GradientEstimate}(x_{k+1}, \varepsilon_{k+1}).$$

  **end if**
  update $H_{k+1} = \mathsf{L\text{-}BFGS}(H_k, x_{k+1} - x_k, g_{k+1} - g_k)$.
**end for**

**output** last proximal mapping $z_K$ (solution).

**Handling composite objective functions.** When sparsity of the solution is desired, the $\ell_1$-norm is typically used. In such case, one may argue that our scheme operates on the smoothed objective $F$, leading to iterates $(x_k)_{k\geq 0}$ that may have small coefficients, but not exact zeroes. Yet, our approach also provides iterates $(z_k)_{k\geq 0}$ that are computed using the original optimization method $\mathcal{M}$ that we wish to accelerate. When $\mathcal{M}$ handles composite problems without smoothing, typically when $\mathcal{M}$ is a proximal block-coordinate, or incremental method, the iterates $(z_k)_{k\geq 0}$ may be sparse. For this reason, our theoretical analysis studies the convergence of the sequence $(f(z_k))_{k\geq 0}$ to the solution $f^*$.

**On the absence of line-search scheme.** A key property of QuickeNing is the absence of line-search, which is usually necessary to ensure the convergence of L-BFGS algorithms [20]. In the context of the Moreau-Yosida regularization, any line-search would be prohibitive, since it would require to evaluate the function $F$ multiple times, hence solving the subproblems (3) as many times. Here, we introduce a simple strategy that selects $x_{k+1} = z_k$ when the sufficient descent condition $F_{\text{test}} \leq F_k - \frac{1}{4\kappa}\|g_k\|^2 + \varepsilon_k$ is not satisfied. $z_k$ is indeed a good candidate since it corresponds to performing one step of the inexact proximal point algorithm whose convergence properties are well understood [12, 15]. Finally, the next proposition provides some convergence guarantees.

**Proposition 2 (Complexity Analysis of QuickeNing for $\mu$-strongly convex $f$)**
*Assume that $\mathcal{M}$ is always able to produce a sequence of iterates $(w_t)_{t\geq 0}$ for solving (3) such that*

$$h(w_t) - h^* \leq A(1-\tau_{\mathcal{M}})^t (h(w_0) - h^*) \text{ for some constants } A, \tau_{\mathcal{M}} > 0.$$

*By choosing the sequence $\varepsilon_k = C(1-\rho)^{k+1}/3$ with $C \geq (f(x_0) - f^*)$ and $\rho < \mu/(4(\mu+\kappa))$,*

$$f(z_k) - f^* \leq C(1-\rho)^{k+2} / \left(\tfrac{\mu}{4(\mu+\kappa)} - \rho\right),$$

*and each sub-problem (3) is solved up to the desired accuracy in at most a constant number $T_{\mathcal{M}}$ of iterations of $\mathcal{M}$, where $T_{\mathcal{M}} = \tilde{O}(1/\tau_{\mathcal{M}})$, where $\tilde{O}$ hides some logarithmic quantities in $\mu$, $L$ and $\kappa$.*

The proof follows in part that of Catalyst [15], but requires significant modifications to accommodate the L-BFGS metric. As in Catalyst, the proposition also holds for optimization methods $\mathcal{M}$ that provide a convergence rate in terms of dual certificate $h(w_t) - g(w_t)$, where $g(w_t)$ is a lower bound on $h^*$; this is the case for SDCA/MISO/Finito [6, 15, 25]. Like classical L-BFGS algorithms, the theoretical complexity is a worst case and does not outperform other first-order methods. In Catalyst, the theoretical analysis is used to set up the parameter $\kappa$; here, the analysis suffers from a mismatch between theory and practice (as any L-BFGS method), and we recommend instead to set up $\kappa$ as in Catalyst [15], which seems to provide good results in practice (see next section).

## 4 Numerical Illustrations

We now present preliminary experiments to compare the performance of QuickeNing applied to the method MISO [17], Catalyst-MISO [15], SAGA [5] (using parameter step-size $1/3L$), which is adaptive to unknown strong convexity, and the L-BFGS implementation developed by Mark Schmidt for smooth objectives, which has been widely used in other comparisons [24]. As an illustration, we consider two datasets, covtype and alpha, and two formulations: the logistic regression problem with $\ell_2$-regularization and the Elastic-net [27] which consists of a least-square objective with the non-smooth regularization $\psi(x) = \lambda\|x\|_1 + \gamma\|x\|^2$. Both of them are strongly convex, with constant lower-bounded by the $\ell_2$-regularization parameter. The parameters $\kappa, \varepsilon_k$ of QuickeNing are all set up according to the rules presented in the previous section and we choose $c_1 = c_2 = 0.5$.

Speed comparison results in terms of gradient evaluations, which dominate the cost of all algorithms, are presented in Figure 1. Our conclusions from this first experiment are encouraging: (i) L-BFGS was always significant behind other approaches that exploit the finite sum structure of the objective; (ii) QuickeNing and SAGA perform equally well on alpha, probably due to some hidden strong convexity in the loss, which Catalyst fails to exploit; (iii) for covtype, QuickeNing was significantly faster than other approaches, especially SAGA that seems to suffer from very ill-conditioned problems.

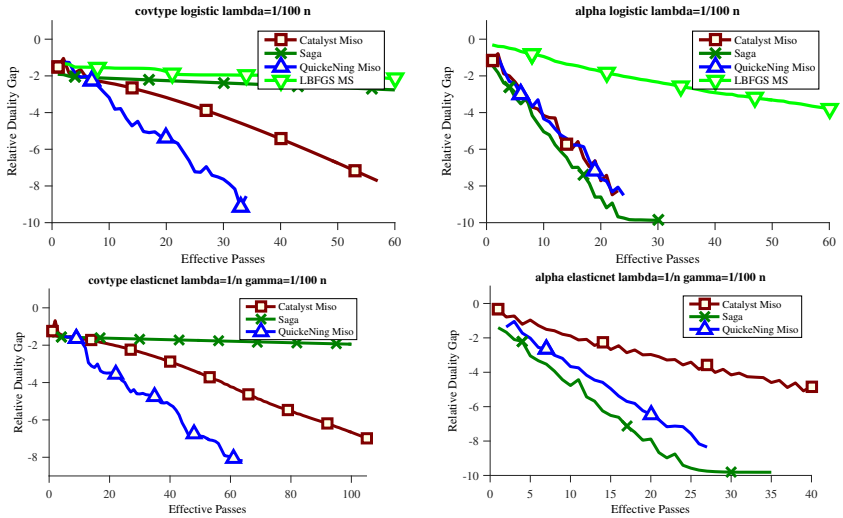


Figure 1: Relative duality gap for different number of passes performed over dataset covtype and alpha. The legend for all curves is on the top right.

**Acknowledgments**

This work was supported by ANR (MACARON project ANR-14-CE23-0003-01), the program "Learning in Machines and Brains" (CIFAR), the project Titan (CNRS-Mastodons), and the MSR-Inria joint centre.
A longer version of this paper is available at https://arxiv.org/abs/1610.00960.

## References


[1] Z. Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *preprint ArXiv:1603.05953*, 2016.

[2] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[3] R. H. Byrd, J. Nocedal, and F. Oztoprak. An Inexact Successive Quadratic Approximation Method for Convex L-1 Regularized Optimization. *Mathematical Programming*, 157(2):375–396, 2016.

[4] X. Chen and M. Fukushima. Proximal quasi-Newton methods for nondifferentiable convex optimization. *Mathematical Programming*, 85(2):313–334, 1999.

[5] A. Defazio, F. Bach, and S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[6] A. Defazio, J. Domke, and T. S. Caetano. Finito: A faster, permutable incremental gradient method for big data problems. In *International Conference on Machine Learning (ICML)*, 2014.

[7] O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1-2):37–75, 2014.

[8] M. P. Friedlander and M. Schmidt. Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing*, 34(3):A1380–A1405, 2012.

[9] M. Fuentes, J. Malick, and C. Lemaréchal. Descentwise inexact proximal algorithms for smooth optimization. *Computational Optimization and Applications*, 53(3):755–769, 2012.

[10] M. Fukushima and L. Qi. A globally and superlinearly convergent algorithm for nonsmooth convex minimization. *SIAM Journal on Optimization*, 6(4):1106–1120, 1996.

[11] R. M. Gower, D. Goldfarb, and P. Richtárik. Stochastic block BFGS: Squeezing more curvature out of data. *preprint arXiv:1603.09649*, 2016.

[12] O. Güler. New proximal point algorithms for convex minimization. *SIAM Journal on Optimization*, 2(4):649–664, 1992.

[13] J. Lee, Y. Sun, and M. Saunders. Proximal Newton-type methods for convex optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[14] C. Lemaréchal and C. Sagastizábal. Practical aspects of the Moreau–Yosida regularization: Theoretical preliminaries. *SIAM Journal on Optimization*, 7(2):367–385, 1997.

[15] H. Lin, J. Mairal, and Z. Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[16] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[17] J. Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

[18] R. Mifflin. A quasi-second-order proximal bundle algorithm. *Mathematical Programming*, 73(1):51–72, 1996.

[19] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[20] J. Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980.

[21] M. Razaviyayn, M. Hong, and Z.-Q. Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013.

[22] P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

[23] S. Salzo and S. Villa. Inexact and accelerated proximal point algorithms. *Journal of Convex Analysis*, 19(4):1167–1192, 2012.

[24] M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *preprint arXiv:1309.2388*, 2013.

[25] S. Shalev-Shwartz and T. Zhang. Proximal stochastic dual coordinate ascent. *preprint arXiv:1211.2717*, 2012.

[26] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, 2001.

[27] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.