# A secant-based Nesterov method for convex functions

**Razak O. Alli-Oke and William P. Heath**

**Optimization Letters**

Optimization Letters manuscript No. OPTL-D-15-00085R1

# A secant-based Nesterov method for convex functions.

Razak O Alli-Oke · William P Heath

Received: 13 March 2015 / Accepted: 15 December 2015

**Abstract** A simple secant-based fast gradient method is developed for problems whose objective function is convex and well-defined. The proposed algorithm extends the classical Nesterov gradient method by updating the estimate-sequence parameter with secant information whenever possible. This is achieved by imposing a secant condition on the choice of search point. Furthermore, the proposed algorithm embodies an "update rule with reset" that parallels the restart rule recently suggested in O'Donoghue and Candes (2013). The proposed algorithm applies to a large class of problems including logistic and least-square losses commonly found in the machine learning literature. Numerical results demonstrating the efficiency of the proposed algorithm are analyzed with the aid of performance profiles.

**Keywords** Convex optimization · Secant Methods · Fast gradient methods · Nesterov gradient method.

## 1 Introduction

This paper considers the unconstrained optimization of convex function $f$:

$$\mathcal{UCOP} : \qquad \min_{x\in\mathbb{R}^n} f(x) \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable convex function. The domain of $f$, $\mathbf{dom}\, f$, is the convex set $\mathbb{R}^n$ and $x$ is a real vector. A necessary and sufficient condition for a point $x^*$ to be a minimizer of $f$ is $\nabla f(x^*) = 0$ [7]. Furthermore, it is assumed that $f$ is bounded below and there exists a unique minimizer $x^* : f(x^*) = f^*$ with $f^* \leq f(x)\ \ \forall\, x \in \mathbb{R}^n$ [27].

This work was funded by EPRSC under the grant EP/H016600/1.

The authors are with the Control Systems Center, School of Electrical and Electronic Engineering,
Sackville Street Building, University of Manchester, Manchester, M13 9PL, UK.
E-mail: razak.alli-oke@manchester.ac.uk, william.heath@manchester.ac.uk

The idea of enhancing or accelerating gradient methods directly has been intensively researched [3, 12, 23, 24, 29, 32] since the pioneering works of Shah et. al. [33] and Polyak [31]. Accelerated gradient methods are easy to implement and offer much lower memory requirement as compared to higher-order methods such as Newton's method. Accelerated gradient methods compute future iterates by relying only on the local gradient and a history of past iterates. Accelerated gradient schemes can be thought of as momentum methods, in that the step taken at the current iteration depends on the previous iterations, and where the momentum grows from one iteration to the next [28]. Accelerated gradient methods, unlike gradient-descent methods, are not guaranteed to monotonically decrease the objective value. In other words, accelerated gradient methods are nonmonotone gradient methods that utilize the momentum from the previous iterates.

Particular schemes include the Barzilai-Borwein gradient method [2], the backpropagation method with momentum [5, 35] - a well-known algorithm in the neural network community - and a fast gradient method developed by Nesterov [24]. All the aforementioned accelerated gradient methods use only the previous iterate and as such they can be considered special cases of two-step iterative algorithms,

$$x_{k+1} = x_k - \alpha_k \nabla f(y_k) + \eta_k (x_k - x_{k-1}) \quad : \quad \alpha_k > 0\,,\ \eta_k \geq 0\,;\ \ y_k = \sum_{i=0}^{i=k} \tau_i x_i\ ,\ \tau_i \in \mathbb{R}\,,$$

with appropriate choice of $\alpha_k$ , $\eta_k$ and $y_k$ . The nonmonotonicity of the accelerated gradient methods are beneficial and contribute to their increased convergence rate [1, 10, 15]. However they are susceptible to severe bumps in the objective values that may be detrimental and may lead to wasted iterations as noted in [28]. This is the case in the Nesterov gradient method when the momentum factor has exceeded a critical value. This can happen when the condition number ( i.e. $q^{-1} = \frac{\mathrm{L}}{\mu}$, where L, $\mu$ are as defined in § 2 ) is underestimated [28]. Moreover, accelerating the gradient method with the precise $q$ in a well-conditioned region can also lead to wasted iterations [28]. The Lipschitz constant L can be estimated in a straightforward manner using backtracking (e.g. [22, pg. 162-163], [4, pg. 195]); however obtaining a nontrivial lower bound for the convexity parameter $\mu$ is much more challenging. In [25], a backtracking approach is taken to estimate a nontrivial convexity parameter. The use of fixed restart proportional to the condition number has also been considered (see e.g. [4, 13, 18]). A heuristic adaptive restart technique was recently introduced in [28] based on the idea of restarting the momentum factor to zero when a heuristic gradient condition is satisfied. The origin of momentum restart can in fact be traced back to the late 80's (see e.g. [36]). O'Donoghue and Candes [28] demonstrate dramatic speed up in the convergence rate of accelerated gradient methods by adaptively restarting the momentum factor with zero when a heuristic gradient condition is satisfied. They show that their restart scheme recovers the optimal complexity $\mathcal{O}(\sqrt{q^{-1}}\ \text{In}\ \frac{1}{\epsilon})$ for strongly-convex quadratic functions. A significant improved performance of accelerated gradient methods combined with this heuristic adaptive restart of [28] was also reported in [11,17,19].

In this paper, we provide a theoretical justification for the heuristic restart condition of [28] by extending the Nesterov gradient method (NGM) to utilize available secant information. The proposed algorithm is based on updating the estimate-sequence parameter with secant information whenever possible. Furthermore, the proposed algorithm embodies an "update rule with reset" that parallels the restart rule suggested in [28]. Numerical examples indicate that the proposed algorithm significantly outperforms the adaptive restart [28]. The rest of this paper is organized as follows: In sections 3 and 4, the Nesterov gradient method and the quasi-Newton method are discussed. The proposed **Secant-Based-NGM** is described in section 5 and the global convergence for all convex functions will also be established therein.

## 2 Notation

A continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ has a Lipschitz continuous gradient on $\mathbb{R}^n$ with constant L if there exist a constant $\mathrm{L} > 0$ such that $\|\nabla h(x) - \nabla h(y)\| \leq \mathrm{L}\|x - y\|\,,\ \forall\, x,\, y \in \mathbb{R}^n$. A continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ is convex with parameter $\mu$ if there exists a constant $\mu \geq 0$ such that $h(x) \geq h(y) + \nabla h(y)(x - y) + \frac{\mu}{2}\|x - y\|^2,\ \ \forall\, x,\, y \in \mathbb{R}^n$. Subsequently, the term trivial convexity parameter means a lower bound of convexity parameter while nontrivial convexity parameter refers to the greatest lower bound of the convexity parameter. Denote $\mathcal{F}^{1,1}_{\mathrm{L}}(\mathbb{R}^n)$ as the class of convex functions with L-Lipschitz-continuous gradient. Let the function $f \in \mathcal{F}^{1,1}_{\mathrm{L}} : \mathbb{R}^n \to \mathbb{R}$ denote a convex function bounded below and with a unique minimizer. If $\mu > 0$, then function $f$ is strongly-convex, i.e. $f \in \mathcal{S}^{1,1}_{\mu,\mathrm{L}}(\mathbb{R}^n) \subset \mathcal{F}^{1,1}_{\mathrm{L}}(\mathbb{R}^n)$. The optimal values of $f(x)$ and $\Phi_k(x)$ are denoted by $f^*$ and $\Phi^*_k$ respectively. Let $\nabla$ denote the gradient operator and it is defined by $\nabla f(x) = \Big[\frac{df(x)}{dx_1}, \cdots, \frac{df(x)}{dx_n}\Big]^T$.

**Assumption 1.** *A trivial convexity parameter $\mu \geq 0$ and the gradient's Lipschitz constant* L *are known.*

## 3 Nesterov Gradient Method

This section reviews the fast gradient method due to Nesterov [24]. Consider the following approximations of $f(x)$ at $x_k$:

$$\phi^1_k(x) = f(x_k) + \nabla f(x_k)^T(x - x_k) + \frac{1}{2\alpha}\|x - x_k\|^2\,. \tag{2}$$

$$\phi^2_k(x) = f(x_k) + \nabla f(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T\nabla^2 f(x_k)(x - x_k)\,. \tag{3}$$

The Nesterov gradient method [24] attempts to use approximations which are better than $\phi^1_k(x)$ but less expensive than $\phi_k^2(x)$ by defining an estimate-sequence (see Definition 1). Provided this estimate-sequence satisfies Nesterov's Principle (see below), then convergence to $f^*$ is guaranteed (see Lemma 1).

**Definition 1** ( [24]). *A pair of sequences $\{\Phi_k(x)\}_{k=0}^{\infty}$, $\{\lambda_k\}_{k=0}^{\infty}$ is called an estimate-sequence of a function $f(x)$ if $\lambda_k \to 0$ and for any $x \in \mathbb{R}^n$ and for all $k \geq 0$, we have:*

$$\Phi_k(x) \leq (1-\lambda_k)f(x) + \lambda_k \Phi_0(x)\,, \tag{4}$$

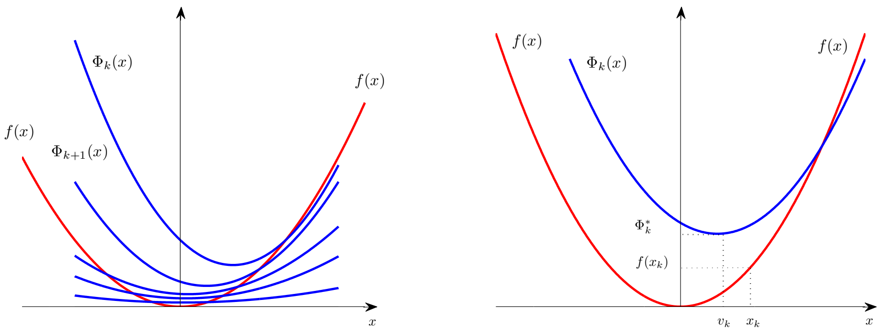
*where $\lambda_k > 0$ and $\Phi_k(x)$ is some local function.* ■

The Nesterov gradient method is based on the principle of utilizing a sequence of local functions $\Phi_k(x)$ whose limit approaches the greatest global lower bound of $f(x)$.

**Nesterov's Principle:** This principle requires that the estimate-sequence (see Definition 1) defined by the local functions $\Phi_k(x)$ is constructed such that

$$f(x_k) \leq \Phi_k^*\,, \qquad \Phi_k^* = \min_x \Phi_k(x)\,. \quad ■ \tag{5}$$

As graphically illustrated in Fig. 1, Nesterov's principle ensures that the local functions $\Phi_k(x)$ constituting the estimate-sequence have a continuum of minima that approaches the minimum of $f(x)$ as $\lambda_k \to 0$. This convergence property of Nesterov's principle is made precise in Lemma 1.



(a) Local function $\Phi_k(x) \to f(x)$ as the parameter $\lambda_k \to 0$. See (4)

(b) Imposed requirement on the choice of local function $\Phi_k(x)$. See (5)

**Fig. 1** Nesterov's optimal concept.
An illustration of Nesterov's principle $\Big( f(x_k) \leq \Phi_k^*$ and $\Phi_k(x) \to f(x)$ as $\lambda_k \to 0 \Big)$.

**Lemma 1** ( [24]). *If a local function $\Phi_k(x)$ is chosen such that Nesterov's principle is satisfied, then*

$$f(x_k) - f(x^*) \leq \lambda_k[\Phi_0(x^*) - f(x^*)]\,, \qquad \forall k > 0\,. \quad ■$$

Thus, for any scheme that satisfies Nesterov's principle $\Big[$i.e. (4), (5)$\Big]$, the convergence rate of its minimization process is directly related to the rate of convergence of the $\lambda_k$ sequence.

The Nesterov scheme is one approach that ensures satisfaction of the Nesterov's principle $\Big[$i.e. (4), (5)$\Big]$. The following Lemma 2, gives a recursive rule that satisfies (4) i.e. Definition 1.

**Lemma 2** ( [24]). *Let scalars* $\lambda_0 = 1$, $\beta_k \in (0,\, 1)$, $\sum\limits_{k=1}^{\infty} \beta_k = \infty$ . *The following recursive rules,*

$$\lambda_{k+1} = (1 - \beta_k)\lambda_k \,, \tag{6}$$

$$\Phi_{k+1}(x) \leq (1 - \beta_k)\Phi_k(x) + \beta_k f(x)\,, \tag{7}$$

*are sufficient to constitute an estimate-sequence* $\{\Phi_k(x)\}_{k=0}^{\infty}$ , $\{\lambda_k\}_{k=0}^{\infty}$ *in the sense of Definition 1.* ■

The Nesterov scheme uses Lemma 2 to construct the estimate-sequence defined in Lemma 3. Thereafter, the acceleration parameter $\beta_k$ and search point $y_k$ are carefully chosen such that (5) is satisfied.

### 3.1 Nesterov's choice of Recursive rule

The Nesterov choice of recursive rule for the local function $\Phi_k(x)$ satisfies the requirements of Lemma 2. In this section, the recursive rule for $\Phi_k(x)$ used in the Nesterov scheme is given and illustrated in Fig. 2.

**Definition 2** ( [24]). *Define* $\overline{\Phi_{k+1}}(x)$ *as*

$$\overline{\Phi_{k+1}}(x) = (1 - \beta_k)\Phi_k(x) + \beta_k[\, f(y_k) + \nabla f(y_k)^T(x - y_k) + \frac{\mu_k}{2}\|x - y_k\|^2\,] \tag{8}$$

*for a given sequence* $\{y_k\}_{k=0}^{\infty}$ *and with* $\mu_k \in [0,\, \mu]$ . ■

**Remark 1.** *The recursion* $\Phi_{k+1}(x) = \overline{\Phi_{k+1}}(x)$ *satisfies Lemma 2. In particular, Nesterov's choice of recursive rule corresponds to* $\Phi_{k+1}(x) = \overline{\Phi_{k+1}}(x)$ *with* $\mu_k = \mu$.

Consequently, Nesterov's choice of local function $\Phi_{k+1}(x)$ is a convex combination of the previous local function $\Phi_k(x)$ and the greatest global lower bound of $f(x)$ . This choice is graphically illustrated in Fig. 2.

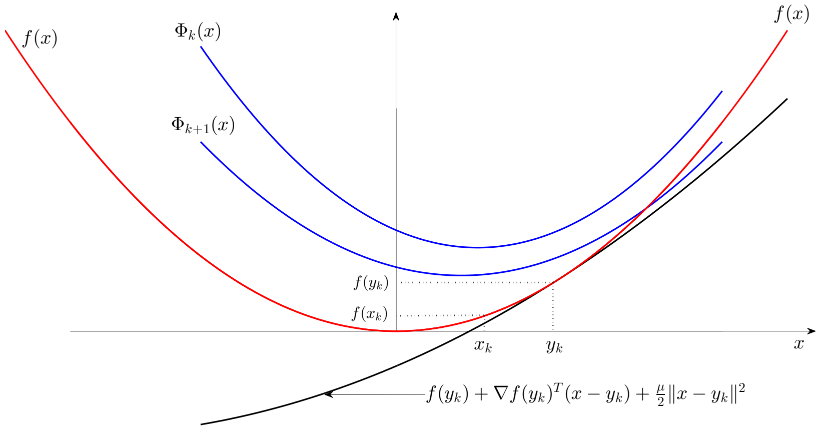


**Fig. 2** Updating the local function $\Phi_k(x)$ for some $y_k$.
Local function $\Phi_{k+1}(x)$ is obtained as a convex combination of $\Phi_k(x)$ and the greatest global lower bound of $f(x)$ at $y_k$. (see Remark 1).

### 3.2 Nesterov's choice of estimate-sequence

In this section, a simple quadratic form is chosen as the initial local function $\Phi_0(x)$ . This simple choice allows the requirements of Lemma 2 to be satisfied easily. Thus the recursion of the sequences defined in Lemma 3 defines an estimate-sequence that satisfies Definition 1.

**Lemma 3** ( [24]). *Let scalars* $\beta_k \in (0,1)$, $\gamma_k > 0$, $\mu \geq 0$, *and* $v_k\,, y_k \in \mathbb{R}^n$ . *Let* $\Phi_0(x) = \Phi_0^* + \frac{\gamma_0}{2}\|x - v_0\|^2$ . *The recursive rules in Lemma 2* $\Big[$*i.e.* (6), (7)$\Big]$ *hold for*

$$\Phi_k(x) = \Phi_k^* + \frac{\gamma_k}{2}\|x - v_k\|^2 \tag{9}$$

*provided the sequences* $\{\gamma_k, v_k, \Phi_k^*\}_{k=0}^{\infty}$ *are defined as*

$$\begin{aligned}
\gamma_{k+1} &= (1-\beta_k)\gamma_k + \beta_k\mu\,, && (10)\\
v_{k+1} &= \frac{1}{\gamma_{k+1}}[(1-\beta_k)\gamma_k v_k + \beta_k\mu y_k - \beta_k\nabla f(y_k)]\,, && (11)\\
\Phi_{k+1}^* &= (1-\beta_k)\Phi_k^* + \beta_k f(y_k) - \frac{\beta_k^2}{2\gamma_{k+1}}\|\nabla f(y_k)\|^2 + \\
&\qquad \frac{\beta_k(1-\beta_k)\gamma_k}{\gamma_{k+1}}\,[\,\frac{\mu}{2}\|y_k - v_k\|^2 + \nabla f(y_k)^T(v_k - y_k)\,]\,. && (12)
\end{aligned}$$

The variables $\gamma_k$ and $\beta_k$ shall subsequently be referred to as the estimate-sequence parameter and acceleration parameter respectively. The estimate-sequence is as desired but still the local condition (5) at the next iterate, $f(x_{k+1}) \leq \Phi_{k+1}^*$ , needs to be ensured. This is subsequently achieved in § 3.3 by carefully choosing the accelerating parameter $\beta_k$ and the search point $y_k$ such that (5) is satisfied.

### 3.3 Nesterov's choice of $\beta_k$ and search point $y_k$

Suppose that $f(x_k) \leq \Phi_k^*$ . Denote $\zeta(\beta_k) = \Big[f(x_k) - f(y_k) + \frac{\beta_k\gamma_k}{\gamma_{k+1}}\,[\,\frac{\mu}{2}\|y_k - v_k\|^2 + \nabla f(y_k)^T(v_k - y_k)\,]\Big]$ . Then $\Phi_{k+1}^*$ (12) can be written as

$$\Phi_{k+1}^* \geq f(y_k) - \frac{\beta_k^2}{2\gamma_{k+1}}\|\nabla f(y_k)\|^2 + (1-\beta_k)\zeta(\beta_k)\,. \tag{13}$$

The choice of $y_k$ and $\beta_k$ that satisfies Nesterov's Principle $\big($i.e. Lemma 3 and $f(x_{k+1}) \leq \Phi_{k+1}^*\,\big)$ is obtained as follows. Take $x_{k+1} = y_k - \alpha_k\nabla f(y_k)$, $\alpha_k = \frac{1}{\mathrm{L}}$ . Then, we have $f(x_{k+1}) \leq f(y_k) - \frac{1}{2\mathrm{L}}\|\nabla f(y_k)\|^2$ . Hence, to satisfy $f(x_{k+1}) \leq \Phi_{k+1}^*$ , the search point $y_k$ is chosen as

$$y_k = \frac{\beta_k\gamma_k v_k + \gamma_{k+1}x_k}{\gamma_k + \beta_k\mu} \;:\; \zeta(\beta_k) \geq \Big[\frac{\beta_k\gamma_k}{\gamma_{k+1}}\,[\,\frac{\mu}{2}\|y_k - v_k\|^2]\Big]\,, \tag{14}$$

and compute $\beta_k$:

$$\beta_k^2 \mathrm{L} = \gamma_{k+1} = (1-\beta_k)\gamma_k + \beta_k \mu \,. \tag{15}$$

**Remark 2.** *The search point* $y_k$ (14) *can be written as*

$$y_k = x_k - \rho_k \nabla \Phi_k(x)\,, \qquad \rho_k = \frac{\beta_k}{\gamma_k + \beta_k \mu}\,, \quad \nabla \Phi_k(x) = \gamma_k (x_k - v_k)\,. \tag{16}$$

The Nesterov scheme is graphically illustrated as shown below in Fig. 3.

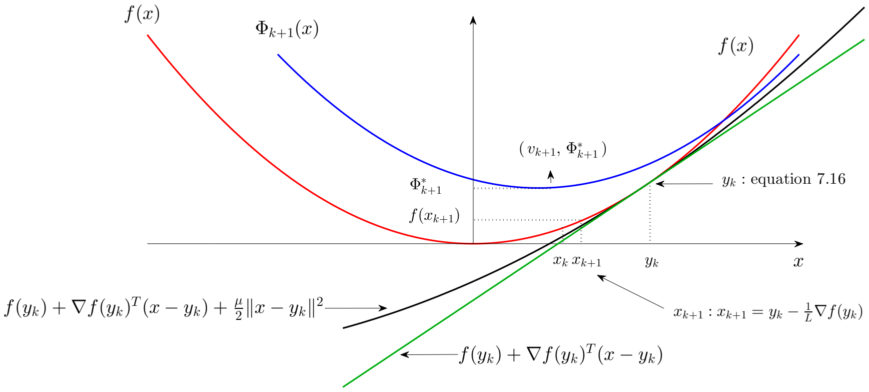

$f(x)$
$\Phi_{k+1}(x)$
$f(x)$
$(v_{k+1}, \Phi^*_{k+1})$
$\Phi^*_{k+1}$
$y_k$ : equation 7.16
$f(x_{k+1})$
$x_k\ x_{k+1}$
$y_k$
$x$
$f(y_k) + \nabla f(y_k)^T (x - y_k) + \frac{\mu}{2}\|x - y_k\|^2$
$x_{k+1} : x_{k+1} = y_k - \frac{1}{L}\nabla f(y_k)$
$f(y_k) + \nabla f(y_k)^T (x - y_k)$

**Fig. 3** Ensuring local condition (5) at the next iterate, $f(x_{k+1}) \le \Phi^*_{k+1}(x)$. After obtaining $\Phi_{k+1}(x)$ as shown in Fig. 2, the choice of $\beta_k$, $y_k$ and $x_{k+1}$ ensures (5).

**Algorithm 1a** ( [24]). The basic Nesterov gradient method is outlined as follows:

**Algorithm 1a** Basic Nesterov gradient method [24].

```
Given a starting point $x_0 \in \mathbf{dom}\, f$, $\gamma_0 > 0$ and $v_0 = x_0$ .
repeat until stopping criterion is satisfied
  1. Compute $\beta_k \in (0,\, 1)$ from $\beta_k^2 \mathrm{L} = (1-\beta_k)\gamma_k + \beta_k\mu$ .   (15)
  2. Compute $\gamma_{k+1}$ :  $\gamma_{k+1} = (1-\beta_k)\gamma_k + \beta_k\mu$ .   (10)
  3. Compute search point: $y_k = x_k - \rho_k\gamma_k(x_k - v_k)$ .   (16)
  4. Compute the Nesterov iterate: $x_{k+1} = y_k - \alpha_k \nabla f(y_k)$ , with $\alpha_k = \frac{1}{\mathrm{L}}$ .
  5. Compute $v_{k+1}$ :  $v_{k+1} = \frac{1}{\gamma_{k+1}}[(1-\beta_k)\gamma_k v_k + \beta_k\mu y_k - \beta_k \nabla f(y_k)]$ .   (11)
end (repeat)
```

**Algorithm 1a** can be simplified by eliminating variables $v_k$ and $\gamma_k$ . With this elimination of $v_{k+1}$ and the estimate-sequence parameter $\gamma_{k+1}$ , **Algorithm 1a** simplifies to **Algorithm 1b**.

**Algorithm 1b** ( [24]). The simplified Nesterov gradient method is outlined as follows:

```
Algorithm 1b Simplified Nesterov gradient method [24].

Given a starting point $x_0 \in \mathbf{dom}\, f$, $\beta_0 \in (0,\, 1)$ , $y_0 = x_0$ and $q = \frac{\mu}{\mathrm{L}}$ .
repeat until stopping criterion is satisfied
    1. Compute the Nesterov iterate: $x_{k+1} = y_k - \alpha_k \nabla f(y_k)$ , with $\alpha_k = \frac{1}{\mathrm{L}}$ .
    2. Compute $\beta_{k+1} \in (0,\, 1)$ from $\beta_{k+1}^2 = (1-\beta_{k+1})\beta_k^2 + q\beta_{k+1}$ .
    3. Compute $\theta_{k+1}$ : $\theta_{k+1} = \dfrac{\beta_k(1-\beta_k)}{\beta_{k+1} + \beta_k^2}$ .
    4. Compute $y_{k+1} = x_{k+1} + \theta_{k+1}(x_{k+1} - x_k)$ .
end (repeat)
```

**Remark 3.** *The choice of* $\beta_0 = \sqrt{\dfrac{\mu}{\mathrm{L}}}$ *corresponds to* $\gamma_0 = \mu$ *while the corresponding* $\beta_0$ *for the case of* $\gamma_0 = \mathrm{L}$ *can be obtained from* (15). *It is important to emphasize that* $\beta_0 \neq 1$ *since* (15) *cannot hold when* $\beta_0 = 1$ . *Hence the choice* $\beta_0 \in (0,\, 1)$ . *Furthermore, if* $\beta_0 = \sqrt{\dfrac{\mu}{\mathrm{L}}}$ , *then* $\beta_k = \sqrt{\dfrac{\mu}{\mathrm{L}}}$ *and* $\theta_k = \dfrac{\sqrt{\mathrm{L}} - \sqrt{\mu}}{\sqrt{\mathrm{L}} + \sqrt{\mu}}$ *for all* $k$ . *Were* $\beta_k$ *be chosen as* 0 *for all* $k \geq 0$ , **Algorithm 1** *would reduce to a fixed-step gradient-descent method. Subsequently, the variable* $\theta_k$ *shall be referred to as the momentum parameter.*

**Theorem 1** ( [24]). *Let* $\Phi_0(x) = \Phi_0^* + \frac{\gamma_0}{2}\|x - v_0\|^2$ . *Suppose* $v_0 = x_0$ . *If a scheme satisfies Nesterov's principle* $\Big[$*i.e.* (4), (5)$\Big]$, *then*

$$f(x_k) - f(x^*) \leq \lambda_k \left[\, f(x_0) - f(x^*) + \frac{\gamma_0}{2}\|x - x_0\|^2 \,\right], \qquad \forall k > 0\,,$$

*where* $\lambda_0 = 1$ *and* $\lambda_k = \prod_{i=0}^{k-1}(1-\beta_i)$ . ■

**Remark 4.** *Take* $\gamma_0 = \mathrm{L}$ *in* ***Algorithm 1a*** *( or the corresponding* $\beta_0$ *in* ***Algorithm 1b*** *). Let* $v_0 = x_0$ . *Then the Nesterov scheme satisfies the premises of Theorem 1. Since* $\gamma_k > 0$ *for all* $k$ , *then the Nesterov gradient method* **Algorithm 1** *generates a sequence* $\{x_k\}_{k=0}^{\infty}$ *such that*

$$f(x_k) - f^* \leq \frac{4\mathrm{L}}{(k+2)^2} \times \|x_0 - x^*\|^2\,. \tag{17}$$

*Furthermore, since* $\gamma_k \geq \mu > 0$ *for all* $k$ , *then*

$$f(x_k) - f^* \leq \min\left\{\, \mathrm{L}(1 - \sqrt{\frac{\mu}{\mathrm{L}}})^k\,,\ \frac{4\mathrm{L}}{(k+2)^2} \,\right\} \times \|x_0 - x^*\|^2\,. \tag{18}$$

## 4 Quasi-Newton Method

In the quasi-Newton [27] method, the local function is a local quadratic model about $x_k$ where $B_k$ is the Hessian-approximate at $x_k$ i.e.

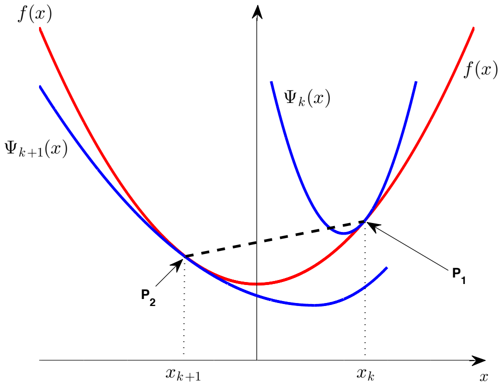


**Fig. 4** Quasi-Newton method: Updating local function $\Psi_k(x)$.
Illustrating the secant line $\mathrm{P}_1$-$\mathrm{P}_2$. The imposed requirement on $B_{k+1}$ is that $\nabla\Psi_{k+1}(x_k) = \nabla f(x_k)$.

$$\Psi_k(x) = f(x_k) + \nabla f(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T B_k (x - x_k)\,. \tag{19}$$

Define the point where $\Psi_k(x_k) = f(x_k)$ and $\nabla\Psi_k(x_k) = \nabla f(x_k)$ as point $\mathrm{P}_1$ (see Fig. 4). Suppose the new iterate $x_{k+1}$ has been generated by minimizing $\Psi_k(x)$ i.e.

$$x_{k+1} = x_k + \alpha_k d_k = x_k - B_k^{-1}\nabla f(x_k) = x_k + s_k\,. \tag{20}$$

We wish to construct $\Psi_{k+1}(x)$ of the form

$$\Psi_{k+1}(x) = f(x_{k+1}) + \nabla f(x_{k+1})^T(x - x_{k+1}) + \frac{1}{2}(x - x_{k+1})^T B_{k+1}(x - x_{k+1})\,. \tag{21}$$

Similarly, define the point where $\Psi_{k+1}(x_{k+1}) = f(x_{k+1})$ and $\nabla\Psi_{k+1}(x_{k+1}) = \nabla f(x_{k+1})$ as $\mathrm{P}_2$ (see Fig. 4). Requirements can be imposed on $B_{k+1}$ based on our knowledge of the previous step. For a reliable Hessian estimate $B_{k+1}$ , it is reasonable to expect that in addition to $\nabla\Psi_{k+1}(x_{k+1}) = \nabla f(x_{k+1})$ , that it is desired to have $\nabla\Psi_{k+1}(x_k) = \nabla f(x_k)$ . Therefore, it follows that

$$B_{k+1}(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k)\,. \tag{22}$$

This imposed condition (22) is known as the secant condition and can be written as:

$$B_{k+1}s_k = y_k\,. \tag{23}$$

Alternatively, the secant condition can be derived from the mean-value theorem for vector-valued functions which implies that (23) is satisfied by the mean Hessian in the interval $[x_k,\, x_{k+1}]$ [6]. The pair $(s_k, y_k)$ is said to be the secant pair associated with the secant condition (23). The matrix $B_k$ is updated ( see [27] ) using symmetric rank-one updates (SR1) or symmetric rank-two updates (e.g. Powell-Symmetric-Broyden(PSB) and Davidon-Flectcher-Powell(DFP) updates).

## 5 A Secant-Based Nesterov Gradient Method

A careful study of (13) reveals that $f(x_k) \leq \Phi_k^*$ in the Nesterov scheme provided that $\beta_k^2 \mathrm{L} - \gamma_{k+1} \leq 0$ . This implies that the accelerating parameter $\beta_k$ can be chosen such that $\beta_k \in (0,\, \beta_k^{'}]$ where $\beta_k^{'}$ is the computed solution of step 1 of **Algorithm 1a**. A natural question that arises is " *how should* $\beta_k \in (0,\, \beta_k^{'}]$ *be chosen*? ". In this section, a new accelerated gradient method ( **Secant-Based-NGM** ) is proposed by extending the classical Nesterov gradient method to utilize available secant information whenever possible. **Secant-Based-NGM** is based on updating the estimate-sequence parameter $\gamma_k$ by imposing a secant condition on the choice of search point $y_k$ . This approach exploits the curvature information at the $k^{\text{th}}$ iterate when determining the accelerating parameter $\beta_k$ . As a result, the computed momentum parameter $\theta_k$ is varied in accordance to the changing curvature of the objective function $f(x)$ . The global convergence of the proposed **Secant-Based-NGM** is also established for all convex functions.

### 5.1 Recursive Rule Revisited

**Definition 3.** *Let* $l_k(x) : l_k(x) = f(y_k) + \nabla f(y_k)^T (x - y_k) + \dfrac{\mu_k}{2} \|x - y_k\|^2$ *for a given sequence* $\{y_k\}_{k=0}^{\infty}$ *and with* $\mu_k \in [0,\, \mu]$ .

Recall from Lemma 2 that it is sufficient for the recursive rule for $\Phi_k(x)$ to satisfy,

$$\Phi_{k+1}(x) \leq (1 - \beta_k)\Phi_k(x) + \beta_k f(x)\,. \tag{24}$$

Recall from Remark 1 that $\Phi_{k+1}(x) = \overline{\Phi_{k+1}}\,(x)$ satisfies (24) where,

$$\overline{\Phi_{k+1}}\,(x) = (1 - \beta_k)\Phi_k(x) + \beta_k l_k(x)\,. \tag{25}$$

In the case $\mu_k \neq 0$ , it then follows from (25) that

$$\overline{\Phi_{k+1}}\,(x) = (1 - \beta_k)\left[\, \Phi_k^* + \frac{\gamma_k}{2}\|x - v_k\|^2 \,\right] + \beta_k \left[\, l_k^* + \frac{\mu_k}{2}\|x - z_k\|^2 \,\right], \tag{26}$$

where $z_k = y_k - \dfrac{1}{\mu_k}\nabla f(y_k)$ , $l_k^* = f(y_k) - \dfrac{1}{2\mu_k}\|\nabla f(y_k)\|^2$ . In the case $\mu_k = 0$ , it follows from (25) that,

$$\overline{\Phi_{k+1}}(x) = (1-\beta_k)\left[\,\Phi_k^* + \frac{\gamma_k}{2}\|x-v_k\|^2\,\right] + \beta_k\left[\,\hat{l_k}^* + \nabla f(y_k)^T x\,\right], \tag{27}$$

where $\hat{l_k}^* = f(y_k) - \nabla f(y_k)^T y_k$ . Subsequently, we do not explore the flexibility in $\mu_k$ and take that $\mu_k = \mu$ as used in the original scheme of Nesterov. Furthermore, the proposed algorithm chooses $\Phi_{k+1}(x)$ such that $\Phi_{k+1}(x) \le \overline{\Phi_{k+1}}(x)$ i.e.

$$\Phi_{k+1}(x) \le (1-\beta_k)\Phi_k(x) + \beta_k[\,f(y_k) + \nabla f(y_k)^T(x-y_k) + \frac{\mu}{2}\|x-y_k\|^2\,]\,. \tag{28}$$

### 5.2 Construction of **Secant-Based-NGM**

**Basic Secant-Based-NGM** extends the classical Nesterov scheme by utilizing the secant information in updating the estimate-sequence parameter $\gamma_k$ . The two subsequent lemmas are used to arrive at an inequality (36) that gives an upper bound to $\gamma_k$ .

**Lemma 4.** *Given $a,\, b > 0$ and $x_1,\, x_2 \in \mathbb{R}^n$ , there exists $x_3 \in \mathbb{R}^n$ and $d \ge 0$ such that $a\|x-x_1\|^2 + b\|x-x_2\|^2 = (a+b)\|x-x_3\|^2 + d$ for all $x \in \mathbb{R}^n$ .*

**Proof**:

Take

$$x_3 = \frac{ax_1 + bx_2}{a+b} \quad \text{and} \quad d = \frac{ab(x_1-x_2)^T(x_1-x_2)}{a+b}\,. \tag{29}$$

It then follows that

$$a\|x-x_1\|^2 + b\|x-x_2\|^2 = (a+b)\|x-x_3\|^2 + d\,. \tag{30}$$

Hence, $d = 0$ if and only if $x_1 = x_2$ . Thus $d \ge 0$ if $a,\, b > 0$ . ■

**Lemma 5.** *Given $a,\, b > 0$ and $x_1,\, x_2 \in \mathbb{R}^n$ , there exists $x_3 \in \mathbb{R}^n$ and $\hat{d} \in \mathbb{R}$ such that $a\|x-x_1\|^2 + bx_2^T x = a\|x-x_3\|^2 + \hat{d}$ for all $x \in \mathbb{R}^n$ .*

**Proof**:

Take

$$x_3 = \frac{2ax_1 - bx_2}{2a} \quad \text{and} \quad \hat{d} = \frac{4abx_1^T x_2 - b^2 x_2^T x_2}{4a}\,. \tag{31}$$

It then follows that $a\|x-x_1\|^2 + bx_2^T x = a\|x-x_3\|^2 + \hat{d}$. It is clear from (31) that $\hat{d} \in \mathbb{R}$ if $a,\, b > 0$ . ■

The subsequent Lemma 6 gives the needed freedom in updating the $\gamma_k$ sequence and also gives the upper bound to the acceptable value of $\gamma_k$ .

**Lemma 6.** *Let scalars $\beta_k \in (0,1)$, $\gamma_k > 0$, $\mu \geq 0$, and $v_k, y_k \in \mathbb{R}^n$. Let $\Phi_0(x) = \Phi_0^* + \frac{\gamma_0}{2}\|x - v_k\|^2$. The recursive rules in Lemma 2 $\Big[$i.e. (6), (7)$\Big]$ hold for*

$$\Phi_k(x) = \Phi_k^* + \frac{\gamma_k}{2}\|x - v_k\|^2 \tag{32}$$

*provided the sequences $\{\gamma_k, v_k, \Phi_k^*\}_{k=0}^{\infty}$ are defined as*

$$\gamma_{k+1}^F = (1-\beta_k)\gamma_k + \beta_k\mu\,, \tag{33}$$

$$v_{k+1} = \frac{1}{\gamma_{k+1}^F}[(1-\beta_k)\gamma_k v_k + \beta_k\mu y_k - \beta_k\nabla f(y_k)]\,, \tag{34}$$

$$\begin{aligned}\Phi_{k+1}^* = (1-\beta_k)\Phi_k^* + \beta_k f(y_k) - \frac{\beta_k^2}{2\gamma_{k+1}^F}\|\nabla f(y_k)\|^2 + \\ \frac{\beta_k(1-\beta_k)\gamma_k}{\gamma_{k+1}^F}\,[\,\frac{\mu}{2}\|y_k - v_k\|^2 + \nabla f(y_k)^T(v_k - y_k)\,]\,,\end{aligned} \tag{35}$$

$$\gamma_{k+1} \leq \gamma_{k+1}^F\,. \tag{36}$$

**Proof**:

From (25), $\overline{\Phi_{k+1}}(x)$ is defined as $\overline{\Phi_{k+1}}(x) = (1-\beta_k)\Phi_k(x) + \beta_k l_k(x)$. Furthermore, choose $v_{k+1}$ as the unconstrained minimum of $\overline{\Phi_{k+1}}(x)$. Since $\gamma_{k+1}^F = (1-\beta_k)\gamma_k + \beta_k\mu$, we can obtain $v_{k+1}$ as,

$$v_{k+1} = \frac{1}{\gamma_{k+1}^F}[(1-\beta_k)\gamma_k v_k + \beta_k\mu y_k - \beta_k\nabla f(y_k)]\,. \tag{37}$$

Next, the inequality (36) is established for the cases when the convexity parameter $\mu \neq 0$ and $\mu = 0$.

Case 1: $\mu \neq 0$

By Lemma 4, it follows from (26) that $\overline{\Phi_{k+1}}(x) = (1-\beta_k)\Phi_k^* + \beta_k l_k^* + d + \frac{\gamma_{k+1}^F}{2}\|x - v_{k+1}\|^2$ for some $d \geq 0$. The case of $d = 0$ occurs if and only if we have coincident minimizers i.e. $v_{k+1} = v_k = z_k$ ( see [12] ). Nesterov's choice then corresponds to $\Phi_{k+1}(x) = \overline{\Phi_{k+1}}(x)$ with $\gamma_{k+1} = \gamma_{k+1}^F$ and $\Phi_{k+1}^* = (1-\beta_k)\Phi_k^* + \beta_k l_k^* + d$. However, we choose

$$\gamma_{k+1} \leq \gamma_{k+1}^F\,. \tag{38}$$

Consequently, we then have $\Phi_{k+1}(x) \leq \overline{\Phi_{k+1}}(x)$ which still satisfies (24) and more importantly we still have $\Phi_{k+1}(v_{k+1}) = \overline{\Phi_{k+1}}(v_{k+1})$. Also note that the unconstrained minimum of this choice of $\Phi_{k+1}(x)$ still remains as $v_{k+1}$ (34). This inequality (36) is crucial in the sense that it gives an upper bound to $\gamma_k$ for all $k$ and allows the use of the secant information in updating the $\gamma_k$ sequence whenever possible.

Case 2: $\mu = 0$

By Lemma 5, it follows from (27) that $\overline{\Phi_{k+1}}(x) = (1-\beta_k)\Phi_k^* + \beta_k\hat{l_k}^* + \hat{d} + \frac{\gamma_{k+1}^F}{2}\|x - v_{k+1}\|^2$ for some

$\hat{d} \in \mathbb{R}$. Thus the inequality (36) still holds for this case using similar arguments as in the case of $\mu \neq 0$.

It remains to compute $\Phi^*_{k+1}$. It follows from (32) that at $x = v_{k+1}$, we have that $\Phi^*_{k+1} = \Phi_{k+1}(v_{k+1})$. Since $\Phi_{k+1}(v_{k+1}) = \overline{\Phi_{k+1}}(v_{k+1})$, then by (25) it follows that $\Phi^*_{k+1}$ can be computed as

$$\Phi^*_{k+1} = (1-\beta_k)\Phi_k(v_{k+1}) + \beta_k\, l_k(v_{k+1})\,. \tag{39}$$

Substitute for $\Phi_k(v_{k+1})$ in (39) using (32), then (39) becomes

$$\begin{aligned}\Phi^*_{k+1} = (1-\beta_k)\Phi^*_k + \frac{(1-\beta_k)\gamma_k}{2}\|v_{k+1}-v_k\|^2 + \\ \beta_k f(y_k) + \beta_k \nabla f(y_k)(v_{k+1}-y_k) + \frac{\beta_k\mu}{2}\|v_{k+1}-y_k\|^2\,.\end{aligned} \tag{40}$$

It follows from (37) that

$$v_{k+1} - y_k = \frac{(1-\beta_k)\gamma_k}{\gamma^F_{k+1}}(v_k - y_k) - \frac{\beta_k}{\gamma^F_{k+1}}\nabla f(y_k)\,, \tag{41}$$

and that

$$v_{k+1} - v_k = \frac{\beta_k\mu}{\gamma^F_{k+1}}(y_k - v_k) - \frac{\beta_k}{\gamma^F_{k+1}}\nabla f(y_k)\,. \tag{42}$$

Substituting (41) and (42) into (40), then the result for $\Phi^*_{k+1}$ follows,

$$\begin{aligned}\Phi^*_{k+1} = (1-\beta_k)\Phi^*_k + \beta_k f(y_k) - \frac{\beta_k^2}{2\gamma^F_{k+1}}\|\nabla f(y_k)\|^2 + \\ \frac{\beta_k(1-\beta_k)\gamma_k}{\gamma^F_{k+1}}\left[\,\frac{\mu}{2}\|y_k - v_k\|^2 + \nabla f(y_k)^T(v_k - y_k)\,\right].\quad \blacksquare\end{aligned} \tag{43}$$

**Remark 5.** *Lemma 3 is a special case of Lemma 6 if* $\gamma_{k+1}$ *is chosen as* $\gamma_{k+1} = \gamma^F_{k+1}$.

The secant information is used in updating $\gamma_k$ by requiring that $\nabla\Phi_{k+1}(y_k) = \nabla f(y_k)$. It then follows that $\hat{\gamma}_{k+1}(y_k - v_{k+1}) = \nabla f(y_k)$ where $\hat{\gamma}_{k+1}$ is a possible update of $\gamma_k$. Using a symmetric rank-1 update, it then follows that

$$\hat{\gamma}_{k+1} = \frac{\nabla f(y_k)^T(y_k - v_{k+1})}{(y_k - v_{k+1})^T(y_k - v_{k+1})}\,. \tag{44}$$

However, this $\gamma_k$ updating is subject to the constraint (36). An effective way of enforcing (36) is that larger values of $\hat{\gamma}_{k+1}$ correspond to updates closer to $\gamma^F_{k+1}$ and vice-versa. To this end, we define $\gamma^{\mathrm{E}}_{k+1}$,

$$\gamma^{\mathrm{E}}_{k+1} = \frac{\hat{\gamma}_{k+1}}{\hat{\gamma}_{k+1} + \gamma^F_{k+1}} \times \gamma^F_{k+1}\,. \tag{45}$$

where the effective curvature $\gamma^{\mathrm{E}}_{k+1}$ is to be used in defining an update rule for $\gamma_k$ in the proposed algorithm. The search point $y_k$ still has to satisfy (16) and the accelerating parameter $\beta_k$ computed from $\beta_k^2\mathrm{L} = \gamma^F_{k+1}$

as in the classical Nesterov gradient method. However, updating $\gamma_k$ based on (44) and (45) ensures that the computed ($\beta_k \in (0,\, \beta_k^{'}]$) satisfies $\beta_k^2 \mathrm{L} - \gamma_{k+1}^{FF} \leq 0$ ( where $\gamma_{k+1}^{FF} = (1-\beta_k)\gamma_k^F + \beta_k \mu$ , see introduction of § 5 ). Moreover, $\beta_k$ is determined by the curvature information at the $k^{\text{th}}$ iterate ( see step 1 of **Algorithm 2a** ).

This computation of $\hat{\gamma}_{k+1}$ comes at an extra cost of 2 vector-vector multiplication. However, as shown in the numerical results, the benefits of computing $\gamma_{k+1}$ outweigh the extra cost of its computation. The update $\gamma_{k+1}$ can now be appended to the classical Nesterov gradient method in a straight-forward manner as shown below in step 6 of **Basic Secant-Based-NGM** below.

**Algorithm 2a** (Basic secant-based algorithm)**.** The outline is as follows:

```
Basic Secant-Based-NGM.

Given a starting point $x_0 \in \mathbf{dom}\, f$, $\gamma_0 > 0$ and $v_0 = x_0$ .
repeat until stopping criterion is satisfied
  1. Compute $\beta_k \in (0,\, 1)$ from $\beta_k^2 \mathrm{L} + \beta_k(\gamma_k - \mu) - \gamma_k = 0$ .
  2. Compute $\gamma_{k+1}^F$ : $\gamma_{k+1}^F = (1-\beta_k)\gamma_k + \beta_k \mu$ .   (33)
  3. Compute search point: $y_k = x_k - \rho_k \gamma_k (x_k - v_k)$ .
  4. Compute the Nesterov iterate: $x_{k+1} = y_k - \alpha_k \nabla f(y_k)$ , with $\alpha_k = \frac{1}{\mathrm{L}}$ .
  5. Compute $v_{k+1}$ : $v_{k+1} = \frac{1}{\gamma_{k+1}^F}[(1-\beta_k)\gamma_k v_k + \beta_k \mu y_k - \beta_k \nabla f(y_k)]$ .   (44)
  6. Compute $\hat{\gamma}_{k+1}$ : $\hat{\gamma}_{k+1} = \frac{\nabla f(y_k)^T (y_k - v_{k+1})}{(y_k - v_{k+1})^T (y_k - v_{k+1})}$ .   (34)
  7. Compute $\gamma_{k+1}$ : $\gamma_{k+1} = \widehat{\min}_\mu\, (\, \gamma_{k+1}^{\mathrm{E}} \,,\, \gamma_{k+1}^F \,)$ .   ··· update rule
  8.     If $\hat{\gamma}_{k+1} < 0$ , then set $\gamma_{k+1} = \beta_k^2 \mu$ .   ··· reset rule
end (repeat)
```

**Remark 6.** *The $\widehat{\min}_\mu$ operator rule in step 7 is given by*

$$c = \widehat{\min}_\mu\,(\,a,\ b\,) : \begin{cases} c = a & \text{if } \gamma_{k+1}^F > \mu, \\ c = \min\,(\,\hat{\gamma}_{k+1}\,,\ b\,) & \text{if } \gamma_{k+1}^F < \mu \text{ and } \hat{\gamma}_{k+1} > \hat{\gamma}_k \\ c = b & \text{if otherwise.} \end{cases}$$

*The extra condition in the case of $\gamma_{k+1}^F < \mu$ serves to penalize oscillation in the trajectory of $\hat{\gamma}_{k+1}$ .*

**Remark 7.** *In the case $\mu \neq 0$ , see Remark 8 for the appropriate reset rule.*

Just as with the classical Nesterov gradient method, $v_{k+1} = x_{k+1} + \dfrac{1-\beta_k}{\beta_k}(x_{k+1} - x_k)$ and the variable $v_k$ can therefore be eliminated. With this elimination of $v_{k+1}$ , **Basic Secant-Based-NGM** simplifies to **Simplified Secant-Based-NGM**.

**Algorithm 2b** (Simplified secant-based algorithm)**.** The outline is as follows:

**Simplified Secant-Based-NGM**.

```
Given a starting point $x_0 \in \mathbf{dom}\, f$, $\beta_0 \in (0,\, 1)$ and $y_0 = x_0$ .
repeat until stopping criterion is satisfied
    1. Compute Nesterov iterate: $x_{k+1} = y_k - \alpha_k \nabla f(y_k)$ , with $\alpha_k = \frac{1}{\mathrm{L}}$ .
    2. Compute $\gamma^F_{k+1} = \beta_k^2 \mathrm{L}$   ;   $\tau_k = \dfrac{1-\beta_k}{\beta_k}$ .
    3. Compute $y_v = [\alpha_k \nabla f(y_k) - \tau_k (x_{k+1} - x_k)]$   ;   $\hat{\gamma}_{k+1} = \dfrac{y_v^T \nabla f(y_k)}{y_v^T y_v}$ .
    4. Compute $\gamma_{k+1}$ : $\gamma_{k+1} = \widehat{\min}_\mu\, (\, \gamma^{\mathrm{E}}_{k+1} \,,\, \gamma^F_{k+1}\,)$ .                    $\cdots$ update rule
    5.      If $\hat{\gamma}_{k+1} < 0$ , then set $\gamma_{k+1} = \beta_k^2 \mu$ .                    $\cdots$ reset rule
    6. Compute $\beta_{k+1} \in (0,\, 1)$ from $\beta_{k+1}^2 \mathrm{L} + \beta_{k+1}(\gamma_{k+1} - \mu) - \gamma_{k+1} = 0$ .
    7. Compute $\theta_{k+1}$ : $\theta_{k+1} = \rho_{k+1} \gamma_{k+1} \tau_k$ ,   where   $\rho_{k+1} = \dfrac{\beta_{k+1}}{\gamma_{k+1} + \beta_{k+1}\mu}$ .
    8. Compute $y_{k+1} = x_{k+1} + \theta_{k+1}(x_{k+1} - x_k)$ .
end (repeat)
```

**Fact**: Let $\gamma_{k+1}$ , $\gamma^F_{k+1}$, $\beta_k$ and $\theta_{k+1}$ be as previously defined. If $\mu = 0$ and $\gamma_{k+1} = \dfrac{\gamma^F_{k+1}}{n}$ for some $n \gg 1$ , then the ratio $\dfrac{\beta_{k+1}}{\beta_k} \approx \dfrac{1}{\sqrt{n}}$ and $\theta_{k+1} \approx \dfrac{1-\beta_k}{\sqrt{n}}$ . The proof follows from the definitions of $\gamma^F_{k+1}$ , $\beta_{k+1}$ and $\theta_{k+1}$ in steps $(\,2, 6, 7)$ respectively.

**Remark 8.** *In the case $\mu = 0$ , then the reset rule in step 5 can replaced with $\gamma_{k+1} = \dfrac{\gamma^F_{k+1}}{\mathrm{L}^2}$ and this corresponds to a momentum factor $\theta_{k+1} <\approx \dfrac{1}{\mathrm{L}}$ . However, to account for the case where $\mathrm{L} \not\gg 1$ , we have used $\gamma_{k+1} = min\,(\, \dfrac{\gamma^F_{k+1}}{\mathrm{L}^2} \,,\, \epsilon\gamma^F_{k+1}\,)$ where $\epsilon = 10^{-6}$ .*

In what follows, the gradient restart condition and restart rule of [28] is contrasted with a gradient condition (46) and the proposed "update rule with reset" respectively. With the substitution of $v_{k+1}$ in **Basic Secant-Based-NGM**, the reset condition $\hat{\gamma}_{k+1} < 0$ in step 5 of **Simplified Secant-Based-NGM** is then equivalent to

$$\alpha_k \|\nabla f(y_k)\|^2 - \tau_k \nabla f(y_k)^T (x_{k+1} - x_k)) < 0\,. \tag{46}$$

This gradient condition (46) is more conservative than the gradient-scheme restart condition suggested in [28] especially when the iterates are far away from the optimum point. Thus the gradient condition (46) is less frequently satisfied. The advantage of the conservativeness of (46) is reinforced by the observation in [28] that "*... restarting far from the optimum can slow down the early convergence slightly, until the quadratic region is reached and the algorithm enters the rapid linear convergence phase.*".

The restart rule of [28] is given as

1. setting $\beta_{k+1}$ as 1.
2. setting the momentum factor $\theta_{k+1}$ as 0.

Firstly, it should be noted that the reset $\beta_{k+1}$ should be in the interval $(0,\, 1)$ since $\beta_k \in (0,\, 1)$ for all $k \geq 0$ ( see Remark 3 ). Moreover, an arbitrary choice of $\beta_{k+1} \in (0,\, 1)$ may correspond to a $\gamma_{k+1}$ that violates the inequality $\gamma_{k+1} \leq \gamma^F_{k+1}$ (36). Furthermore, the proposed **Simplified Secant-Based-NGM** proceeds with the computed $\beta_{k+1}$ unlike the restart rule 1 of [28]. It is emphasized that the proposed **Simplified Secant-Based-NGM** does not reset the momentum factor to zero ( see Remark 8 ) unlike the restart rule 2 of [28]. Thus the **Secant-Based-NGM** is not a momentum-restart algorithm, and the proposed "update rule with reset" ( see Remarks $6, 7, 8$ ) in the **Secant-Based-NGM** satisfies inequality (36). Moreover, restarting the momentum factor $\theta_{k+1}$ with 0 as stated in [28] inhibits fast convergence since it discards the entire accumulated information from previous iterates. In the next sub-section, we establish the global convergence of the proposed **Secant-Based-NGM** for all convex functions.

### 5.3 Global Convergence of Proposed Scheme

The scheme construction of the proposed algorithm $\big($**Secant-Based-NGM**$\big)$ satisfies Nesterov's principle $\Big[$i.e. $(4), (5)\Big]$. Thus the proposed scheme satisfies the premises of Theorem 1 and therefore the **Secant-Based-NGM** is globally convergent with

$$f(x_k) - f(x^*) \leq \Big( \prod_{i=0}^{k-1} (1-\beta_i) \Big) \times \Big[ f(x_0) - f(x^*) + \frac{\gamma_0}{2} \|x - x_0\|^2 \Big] \,, \qquad \forall k \geq 1 \,, \tag{47}$$

where $\beta_i \in (0,\, 1)$ and $\gamma_0 > 0$ . If the gradient's Lipschitz constant L is known, then (47) reduces to

$$f(x_k) - f(x^*) \leq \Big( \prod_{i=0}^{k-1} (1-\beta_i) \Big) \times \Big[ \frac{\gamma_0 + \mathrm{L}}{2} \|x - x_0\|^2 \Big] \,, \qquad \forall k \geq 1 \,. \tag{48}$$

## 6 Numerical Results

Consider test examples of the form

$$\mathcal{UCOP} \;: \qquad \min_{x \in \mathbb{R}^n} \, f(x), \tag{49}$$

where $x \in \mathbb{R}^n$ is the unknown variable. The numerical tests investigate the effects of increasing the dimension and condition number respectively on the performance of the proposed algorithm.

Computational Setup: All numerical tests were coded in 64-bit MATLAB on a Dell-Optilex-780 PC with Intel dual-core CPU of 2.93 GHz, RAM of 16GB and a 160GB-free Hard-Disk. All Matlab sessions were single-threaded, *feature('accel','off')* and process-Priority set to "High". All Matlab sessions were executed on the PC running Windows 7 in "'Safe Mode"'. The stopping criterion was $\|\nabla f(x)\| \le 10^{-9}$ and $\|\nabla f(x)\| \le 10^{-6}$ for examples $1, 3$ and $2, 4$ respectively. All matrices are random square matrices such that the Hessian has eigenvalues in the interval $[\,1{+}\mu\,,\mathrm{L}{+}\mu\,]$ . This was achieved using singular value decomposition to allocate the desired eigenvalues. All matrices and vectors were randomly generated in MATLAB with seed *rng(1234,'twister').* Except for Ex. 4 , the computations were also repeated with seed *rng(5678,'twister').* Except for Ex. 2 , the first run-time effects were accounted for by ignoring the run-times of the first run while averaging over the subsequent 3 runs of each algorithm.

Test Functions and Solvers: We consider four sets of problems with varied number of instances. The test function in Ex. 1 is is a well-known convex quadratic function used for benchmarking convex solvers while the test functions in Ex. 2 and Ex. 3 are convex non-quadratic functions usually encountered in machine-learning literature [8, 14, 26, 34, 37]. The test problems in Ex. 4 is the well-known Maros and Meszaros's collection used for benchmarking convex quadratic program solvers [16, 20]. A first-order oracle [24] is used in all numerical computations. The set of solvers $\mathcal{S}$ considered is:

- Classical Nesterov gradient method(NGM)[1],
- Adaptive restart [28],
- Fixed restart[2] after $k = \sqrt{\dfrac{8\mathrm{L}}{\mu}}$ ,
- Proposed algorithm : **(Simplified) Secant-Based-NGM**.[3]

In all cases, $\beta_0 = \sqrt{\mu/\mathrm{L}}$ if $\mu \neq 0$ and $\beta_0 = \dfrac{\sqrt{5}-1}{2}$ if otherwise. These chosen $\beta_0$ correspond to $\gamma_0 = \mu$ and $\gamma_0 = \mathrm{L}$ respectively. The performance profile in the sense of Dolan and More [9] is adopted to analyze the performance data of the above set of solvers $\mathcal{S}$ on each problem set $\mathcal{P}$ . The percentage of the test problems for which a method is the fastest is given on the left axis of the profile plot. The right side of the profile plot gives the percentage of the test problems that were successfully solved by each of the methods. In other words, the right side of the profile plot is a measure of an algorithm's robustness.

**Ex. 1**: Ridge regression problem [21]

This is a linear least squares problem with Tikhonov regularization. Given $A \in \mathbb{R}^{m\times n}$ , $b \in \mathbb{R}^n$ and

[1] The simplified Nesterov gradient method **Algorithm 1b** is used for computations.

[2] In the case of $\mu = 0$ (Ex. 2 and Ex. 4), fixed restart is done after $k = \max\{\mathrm{N}, \sqrt{\mathrm{L}}\}$ iterations.

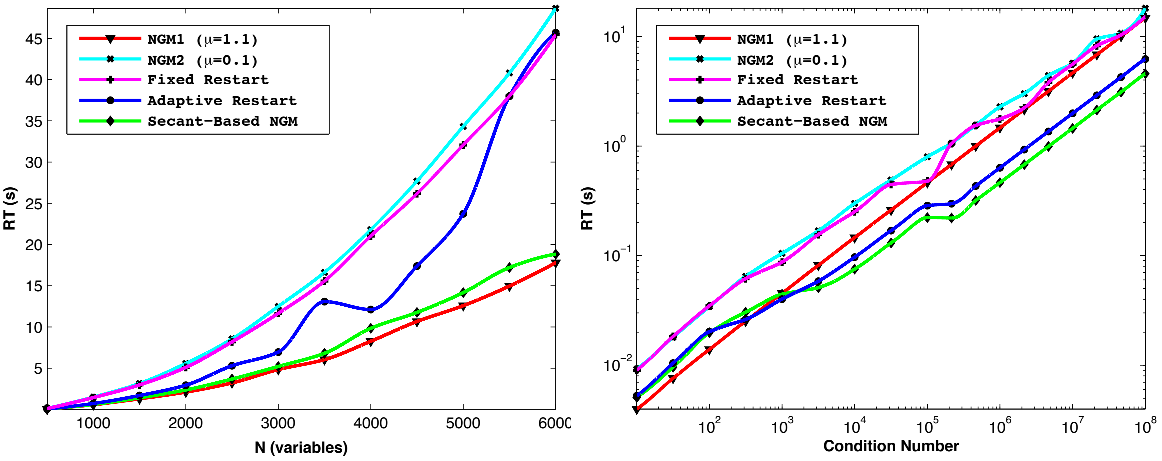
[3] In the case of $\mu = 0$ (Ex. 2 and Ex. 4), the algorithm resets with $\gamma_{k+1} = \min\left(\dfrac{\gamma^F_{k+1}}{\mathrm{L}^2},\, 10^{-6}\gamma^F_{k+1}\right)$ , see Remark 8.

the convexity parameter $\mu = 0.1$ .

$$f(x) = \frac{\mu}{2}\|x\|_2^2 + \frac{1}{2}\|Ax - b\|_2^2 \,. \tag{50}$$

The objective function $f(x) \in \mathcal{S}^{1,1}_{\mu,\mathrm{L}}$ is a positive-definite quadratic convex function with Lipschitz gradient of $\mathrm{L} = \lambda_{max}(A^T A) + \mu$ and trivial convexity parameter of $\mu = 0.1$ . All algorithms use the trivial convexity parameter $\mu = 0.1$ except algorithm NGM1 which used a nontrivial convexity parameter of $\mu = 1.1$ .

The plots in Fig. 5(a) and Fig. 5(b) show the effect of increasing the problem size (N) and condition number (L/$\mu$) respectively. As expected the NGM with trivial convexity parameter $\mu = 0.1$ (NGM2) is slowest with the largest run-time (RT) as observed in Fig. 5(a). Using a fixed restart shows slightly improved performance with increasing dimension. However, the adaptive restart and **Secant-Based-NGM** perform significantly better than NGM2 while the **Secant-Based-NGM** performs comparable with NGM1 as the dimension number increases. It can be noted in Fig. 5(b) that the adaptive restart and **Secant-Based-NGM** perform better than even the NGM1. Moreover, the **Secant-Based-NGM** outperforms the adaptive restart as the condition number becomes high.

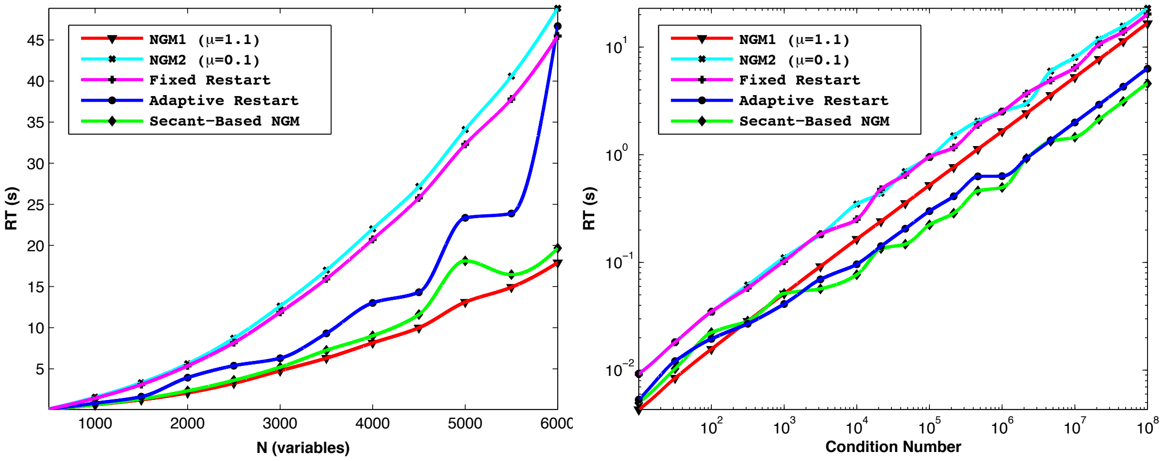


(a) L=100.

(b) N=100.

**Fig. 5** Effect of problem size (N) and condition number (L/$\mu$) on the run-time.
NGM1 - NGM with nontrivial $\mu$ ; NGM2 - NGM with trivial $\mu$.

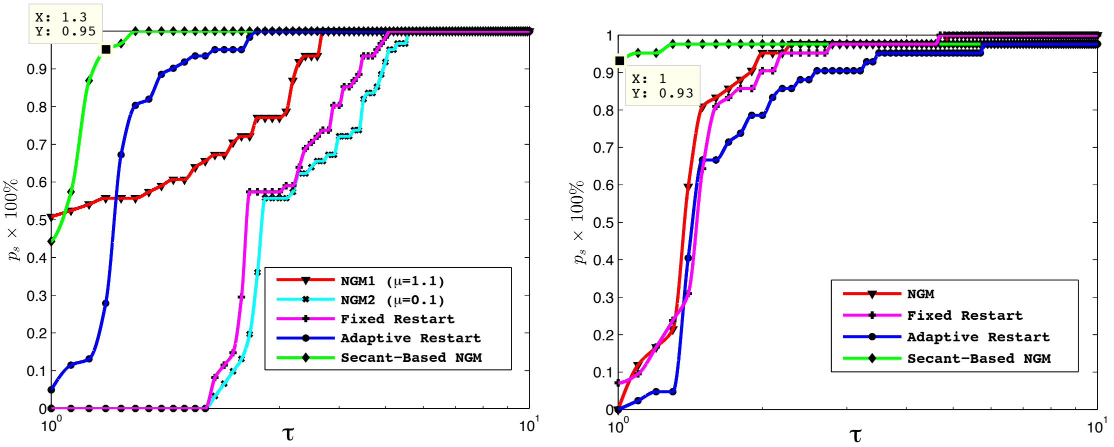
The computations were repeated for randomly-generated matrices and vectors with seed *rng(5678,'twister')*. The plots in Fig. 6(a) and Fig. 6(b) show the effect of increasing the problem size (N) and the condition



(a) L=100. (b) N=100.

**Fig. 6** Effect of problem size (N) and condition number (L/$\mu$) on the run-time. NGM1 - NGM with nontrivial $\mu$ ; NGM2 - NGM with trivial $\mu$.

number (L/$\mu$) respectively. Similar conclusions can be drawn from the plots in Fig. 6(a) and Fig. 6(b). The performance profiles [9] for all problem instances $\Big($ i.e. with seeds *rng(1234,'twister')* and *rng(5678,'twister')* $\Big)$ is shown in Fig. 7(a). As expected, It is clear from Fig. 7(a) that the NGM with nontrivial convexity has the most wins (i.e. the highest probability of being the optimal solver). However, it can also be observed



(a) Ex. 1 (b) Ex. 2

**Fig. 7** The performance profiles of the set of solvers $\mathcal{S}$ for Ex. 1 and Ex. 2.

from Fig. 7(a) that the performance of NGM with a trivial convexity parameter is improved when a fixed restart or adaptive restart [28] is used. Moreover, **Secant-Based-NGM** performs better than when restarts are used with NGM. In general, the proposed **Secant-Based-NGM** has the highest probability $\left(p_s(\tau){=}0.95\right)$ of being the fastest solver within a factor $\tau = 1.3$ of the best solver.

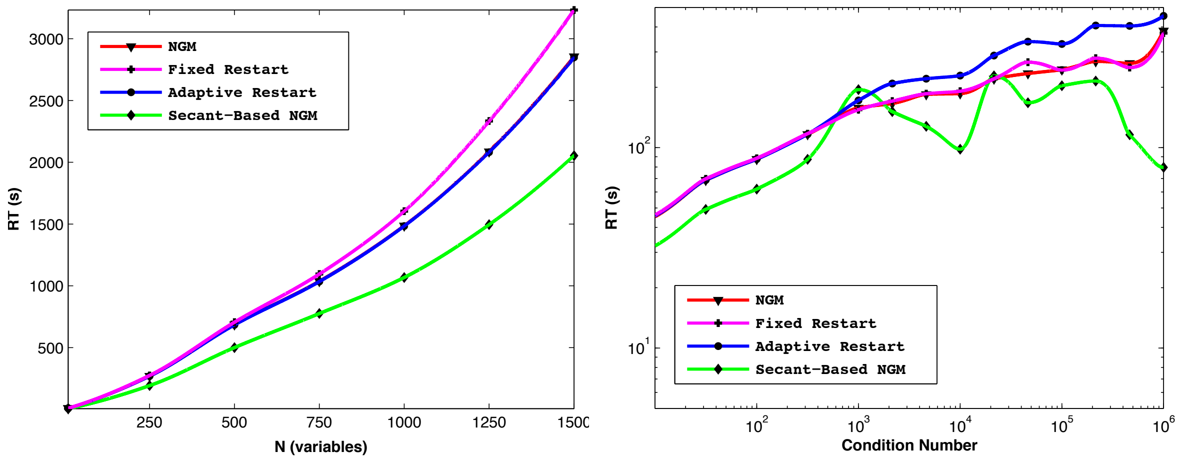
**Ex. 2**: Binary classification problem [8, 26]

This a logistic regression problem with $l_2$-regularization. Given $z_i \in \mathbb{R}^n$ , $y_i \in \{-1,\, 1\}$ , $\mu = 0$ and $\mathrm{N} \geq 1$ .

$$f(x) = \frac{\mu}{2}\|x\|_2^2 + \sum_{i=1}^{\mathrm{N}} \log(1 + e^{-y_i z_i^T x})\,. \tag{51}$$

The objective function $f(x) \in \mathcal{F}_{\mathrm{L}}^{1,1}$ is a non-quadratic convex function with Lipschitz gradient of $\mathrm{L} = 0.25\lambda_{max}(F^TF)$ and convexity parameter of $\mu = 0$ , where the design matrix $F = [z_1 \cdots z_N]^T \in \mathbb{R}^{\mathrm{N}\times n}$. The explained variable $y = \mathrm{sign}(w^T F^T)$ was generated as described in [8,30] except that $w = [1;1;1\cdots 1]$ . The choice of $w$ means that each feature has an equal effect on the explained variable $y$.

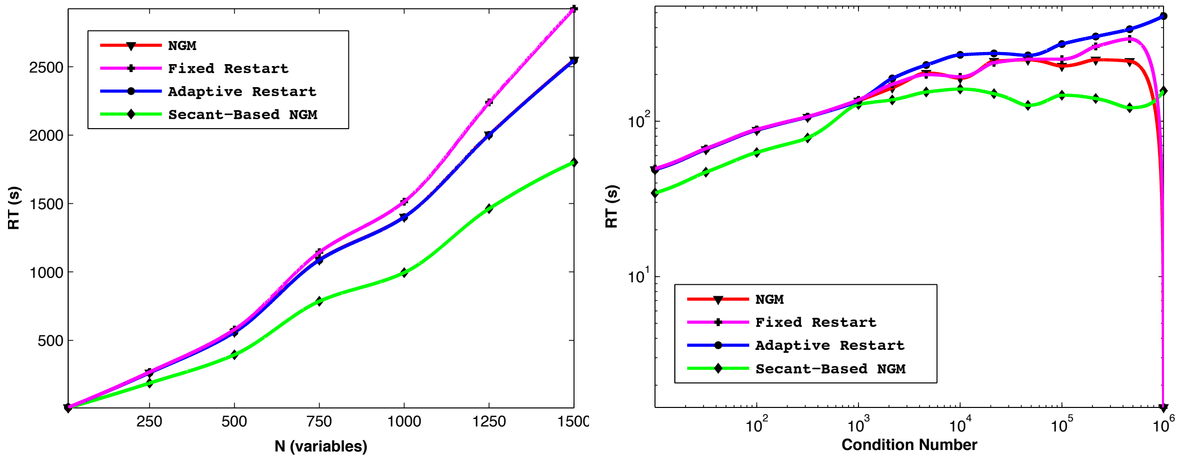
The plots in Fig. 8(a) and Fig. 8(b) show the effect of increasing the problem size (N) and condition number (L/$\mu$) respectively. The computations were repeated for randomly-generated matrices and vectors with seed *rng(5678,'twister')*. The plots in Fig. 9(a) and Fig. 9(b) show the effect of increasing the problem size (N) and condition number (L/$\mu$) respectively. Similar conclusions can be drawn from the plots in both Fig. 8 and Fig. 9. It can be observed in these figures that all the algorithms have similar performance except for **Secant-Based-NGM** that clearly outperforms the rest as the dimension number increases. In Fig. 9(b), we observe a rather strange behaviour of the Fixed Restart and NGM at the end of the plot. Nevertheless, the proposed **Secant-Based-NGM** significantly outperforms all other



(a) L=100.

(b) N=100.

**Fig. 8** Effect of problem size (N) and condition number (L/$\mu$) on the run-time.



(a) L=100. (b) N=100.

**Fig. 9** Effect of problem size (N) and condition number (L/$\mu$) on the run-time.

algorithms as the condition number increases while the adaptive restart performs worse than the NGM.

The performance profiles [9] for all problem instances is shown in Fig. 7(b). It is clear from Fig. 7(b) that **Secant-Based-NGM** is the most efficient of the considered solvers. It solved 93% of the problems significantly faster than the other solvers. It can also be observed from Fig. 7(b) that the performance of adaptive restart [28] is worse than the fixed restart or the classical NGM. In general, the proposed **Secant-Based-NGM** significantly improves over and above the classical NGM and the adaptive restart suggested in [28].

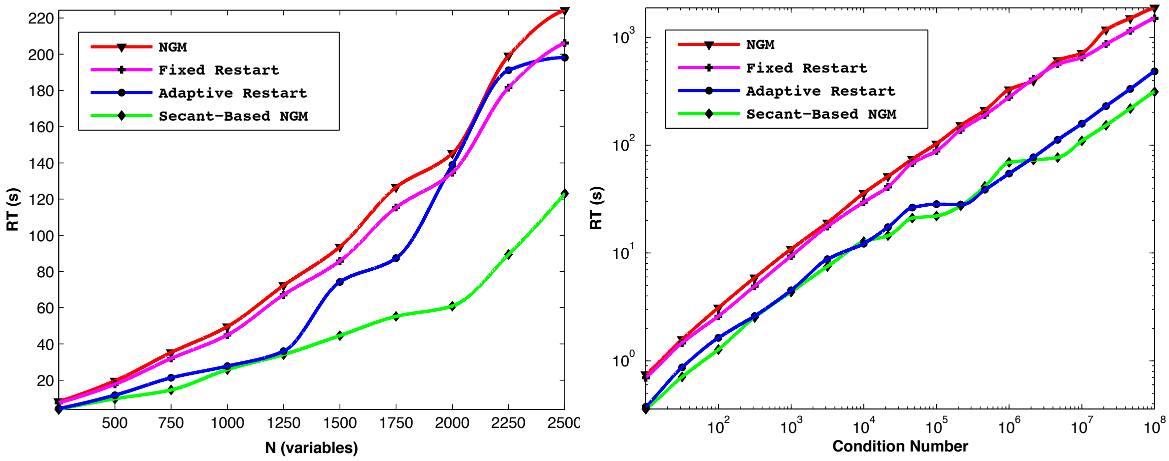
**Ex. 3**: Approximate Huber loss [14, 34, 37]

Given $z_i \in \mathbb{R}^n$ , $y_i \in \{-1, 1\}$ , $\mu = 0.1$ and $\mathrm{N} \geq 1$ .

$$f(x) = \frac{\mu}{2}\|x\|_2^2 + \sum_{i=1}^{\mathrm{N}} \log\Big(\cosh(y_i - z_i^T x)\Big) . \tag{52}$$

The objective function $f(x) \in \mathcal{S}_{\mu,\mathrm{L}}^{1,1}$ is a non-quadratic convex function with Lipschitz gradient of $\mathrm{L} = \lambda_{max}(F^T F)$ and convexity parameter of $\mu = 0.1$ , where the design matrix $F = [z_1 \cdots z_N]^T \in \mathbb{R}^{\mathrm{N}\times n}$. The explained variable $y = \mathrm{sign}(w^T F^T)$ was generated as described in Ex. 2.

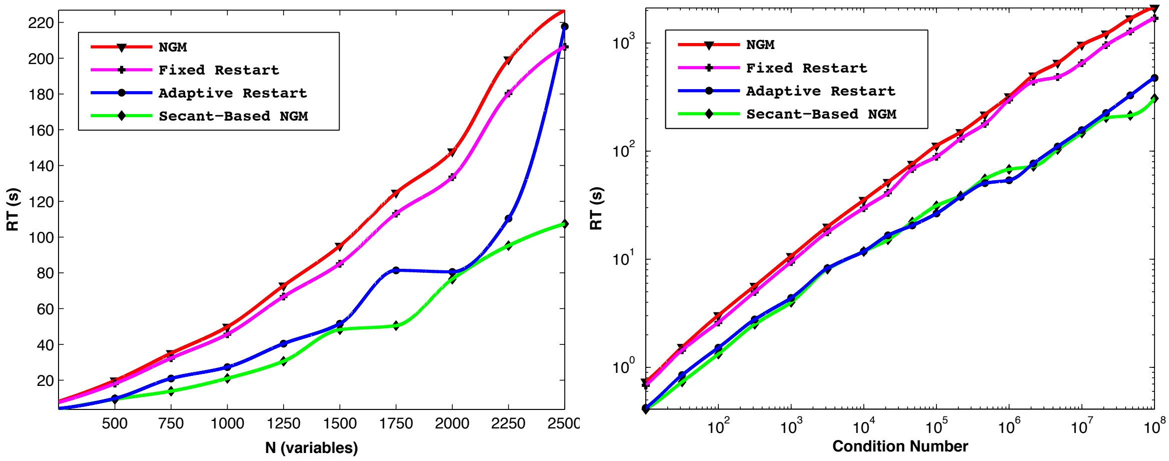
The plots in Fig. 10(a) and Fig. 10(b) show the effect of increasing the problem size (N) and condition number (L/$\mu$) respectively. The computations were repeated for randomly-generated matrices and vectors with seed *rng(5678,'twister')*. The plots in Fig. 11(a) and Fig. 11(b) show the effect of increasing the problem size (N) and condition number (L/$\mu$) respectively. Similar conclusions can be drawn from the



(a) L=100. (b) N=100.

**Fig. 10** Effect of problem size (N) and condition number (L/$\mu$) on the run-time.

plots in both Fig. 10 and Fig. 11. It can be observed in these figures that all the algorithms have similar performance except for **Secant-Based-NGM** that clearly outperforms the rest as the dimension number increases. Also in Fig. 10(b), the proposed **Secant-Based-NGM** significantly outperforms other algorithms as the condition number increases while the NGM has the worst performance.



(a) L=100. (b) N=100.

**Fig. 11** Effect of problem size (N) and condition number (L/$\mu$) on the run-time.

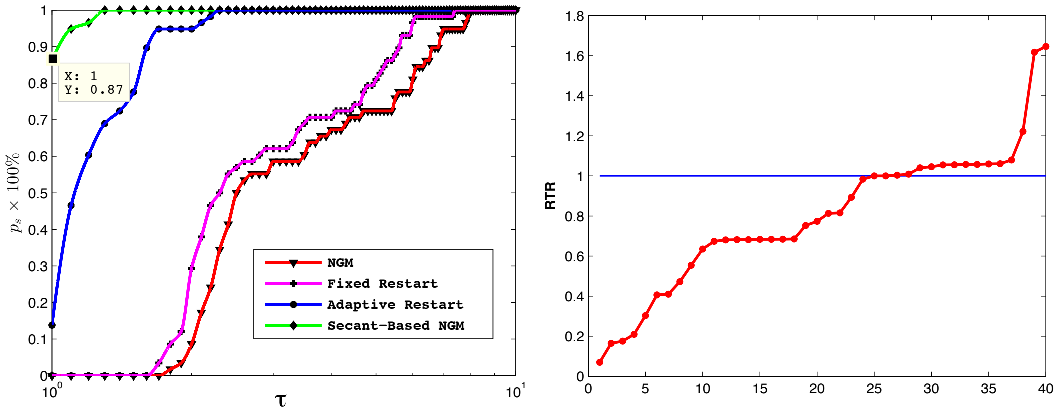
The performance profiles [9] for all problem instances is shown in Fig. 12(a). It is clear from Fig. 12(a) that **Secant-Based-NGM** is the most efficient of the considered solvers. It solved about 87% of the problems significantly faster than the other solvers. It can also be observed from Fig. 12(a) that the adaptive restart [28] has better run-time performance than the classical NGM. In general, the proposed **Secant-Based-NGM** significantly improves over and above the classical NGM and the adaptive restart suggested in [28].

**Ex. 4**: Maros and Meszaros Test Problems [16, 20]

This is a test suite of 138 convex quadratic programs by Maros and Meszaros. This test suite contains selected problems from the BRUNEL and CUTE collections. The objective functions were regularized with $10^{-9}\|x\|_2^2$ to ensure that the Hessians are positive-definite. All considered algorithms use the trivial convexity parameter $\mu = 0$. The initial vectors $x_0$ were randomly-generated with seed *rng(1234,'twister')*.

In Table 1 (see Appendix), we report the running time (RT) for each of the solvers. We report only on those test problems in which the run-time (RT) of the classical NGM is within the interval $[1s, 500s]$. We also ensured that the objective functions of the considered test problems are not in repetition.

As expected the classical NGM had the slowest run-times. The run-time is slightly improved using a fixed-restart and further improved using the adaptive restart. However, as shown in Fig. 12(b), our proposed algorithm **Secant-Based-NGM** performs significantly better than the adaptive restart in about 65% of the test problems and no worse than 1.65 times slower than the adaptive restart.



(a) Performance profiles for Ex. 3

(b) Ratio profile for Ex. 4

**Fig. 12** The performance profiles and ratio profile of the set of solvers $\mathcal{S}$ for Ex. 3 and Ex. 4 respectively. RTR is the ratio of the Secant-Based NGM's run-time to the Adaptive Restart's run-time.

## 7 Conclusion

The Nesterov gradient method (NGM) needs to be restarted (e.g. [18, 28]) when a nontrivial convexity parameter is not available. This paper introduces a new secant-based Nesterov gradient method (**Secant-Based-NGM**) and also establishes that it is globally convergent for all convex functions. The algorithm only requires a trivial lower bound of the convexity parameter $\mu \geq 0$ and the gradient's Lipschitz constant. The efficiency of the proposed algorithm derives from updating the estimate-sequence parameter $\gamma_k$ by imposing a secant condition on the choice of search point $y_k$. Furthermore, the proposed **Secant-Based-NGM** embodies an "update rule with reset" that parallels the restart rule suggested in [28]. The effectiveness of the proposed algorithm is confirmed in numerical computations involving large dataset of test problems with varying dimension and condition number. The proposed **Secant-Based-NGM** significantly improves over and above the classical NGM and when restarts (e.g. [18, 28]) are used with the Nesterov gradient method.

## References

1. Amini, K., Ahookhosh, M., Nosratipour, H.: An inexact line search approach using modified nonmonotone strategy for unconstrained optimization. Numerical Algorithms **66(1)**, 49–78 (2013)
2. Barzilai, J., Borwein, J.: Two-point step size gradient methods. IMA Journal of Numerical Analysis **8**, 141–148 (1988)
3. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal of Imaging Sciences **2(1)**, 183–202 (2009)
4. Becker, S.R., Candes, E.J., Grant, M.C.: Templates for convex cone problems with applications to sparse signal recovery. Mathematical Programming Computation **3(3)**, 165–218 (2011)
5. Bhaya, A., Kaszkurewicz, E.: Steepest descent with momentum for quadratic functions is a version of the conjugate gradient method. Neural Networks **17(1)**, 65–71 (2004)
6. Birgin, E.G., Martinez, J.M., Raydan, M.: Nonmonotone spectral projected gradient methods on convex sets. SIAM Journal on Optimization **10(4)**, 1196–1211 (2000)
7. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2009)
8. Collins, M., Schapire, R.E., Singer, Y.: Logistic regression, Adaboost and Bregman distances. Machine Learning **48**, 253–285 (2002)
9. Dolan, E.D., More, J.J.: Benchmarking optimization software with performance profiles. Mathematical Programming **91(2)**, 201–213 (2002)
10. Fletcher, R.: On the Barzilai-Borwein method. In: L. Qi, K. Teo, X. Yang (eds.) Optimization and Control with Applications, *Applied Optimization*, vol. 96, pp. 235–256. Springer US (2005)
11. Goldstein, T., O'Donoghue, B., Setzer, S.: Fast alternating direction optimization methods. Technical Report, UCLA (May 2012 (Revised January 2014))
12. Gonzaga, C.C., Karas, E.W.: Fine tuning Nesterov's steepest descent algorithm for differentiable convex programming. Mathematical Programming, Series A **138**, 141–166 (2013)
13. Gu, M., Lim, L.H., Wu, C.: ParNes: A rapidly convergent algorithm for accurate recovery of sparse and approximately sparse signals. Numerical Algorithms **64(2)**, 321–347 (2013)
14. He, R., Tan, T., Wang, L.: Robust recovery of corrupted low-rank matrix by implicit regularizers. IEEE Transactions on Pattern Analysis and Machine Intelligence **36(4)**, 770–783 (2014)
15. Hu, S.L., Huang, Z.H., Lu, N.: A nonmonotone line search algorithm for unconstrained optimization. Journal of Scientific Computing **42**, 38–53 (2010)
16. Kozma, A., Conte, C., Diehl, M.: Benchmarking large-scale distributed convex quadratic programming algorithms. Optimization Methods and Software **30(1)**, 191–214 (2015)

17. Kozma, A., Frasch, J.V., Diehl, M.: A distributed method for convex quadratic programming problems arising in optimal control of distributed systems. In: Proceedings of the 52nd IEEE Conference on Decision and Control, Florence, Italy (December 2013)
18. Lan, G., Monteiro, R.D.: Iteration-complexity of first-order penalty methods for convex programming. Mathematical Programming **138(1-2)**, 115–139 (2013)
19. Lin, Q., Xiao, L.: An adaptive accelerated proximal gradient method and its homotopy continuation for sparse optimization. In: Proceedings of The 31st International Conference on Machine Learning, Beijing, China (2014)
20. Maros, I., Meszaros, C.: A repository of convex quadratic programming problems. Optimization Methods and Software **11(1-4)**, 671–681 (1999)
21. Meng, X., Chen, H.: Accelerating Nesterov's method for strongly convex functions with Lipschitz gradient. Mathematics -Optimization and Control, 90C25. arXiv:1109.6058v1 pp. 1–13 (2011)
22. Nemirovski, A.S.: Efficient methods in convex programming, Lecture Notes, Technion - Israel Institute of Technology (1994)
23. Nesterov, Y.: A method of solving a convex programming problem with convergence rate of $(1/k^2)$. Soviet Mathematics Doklady **27(2)**, 372–376 (1983)
24. Nesterov, Y.: Introductory Lectures on Convex Programming: A Basic Course. Kluwer Academic Publishers, Dordrecht (2004)
25. Nesterov, Y.: Gradient methods for minimizing composite objective function. CORE discussion paper (2007)
26. Nicolas, L.R., Mark, S., Francis, B.: A stochastic gradient method with an exponential convergence rate for finite training sets. Mathematics - Optimization and Control, arXiv:1202.6258v4 pp. 1–34 (2013)
27. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York (2006)
28. O'Donoghue, B., Candes, E.: Adaptive restart for accelerated gradient schemes. The Journal of the Society for Foundations of Computational Mathematics **1**, 1–18 (2013)
29. Patrinos, P., Bemporad, A.: An accelerated dual gradient-projection algorithm for linear model predictive control. In: Proceedings of the 51st IEEE Conference on Decision and Control. Maui, US (December 2012.)
30. Pedregosa, F.: Numerical optimizers for logistic regression (2013). URL http://fa.bianp.net/blog/2013/numerical-optimizers-for-logistic-regression/#fn:2
31. Polyak, B.: Some methods of speeding up the convergence of iteration methods. USSR Computational Mathematics and Mathematical Physics **4(5)**, 1–17 (1964)
32. Richter, S., Jones, C.N., Morari, M.: Real-time input-constrained MPC using fast gradient methods. In: Proceedings of the 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference, Shanghai, China (December 2009)
33. Shah, B., Buehler, R., Kempthorne, O.: Some algorithms for minimizing a function of several variables. Journal of the Society for Industrial and Applied Mathematics **12(1)**, 74–92 (1964)
34. Telgarsky, M.: Steepest descent analysis for unregularized linear prediction with strictly convex penalties. In: Proceedings of the 4th International Workshop on Optimization for Machine Learning (OPT), held as a part of the NIPS workshops series (December 2011)
35. Torii, M., Hagan, M.T.: Stability of steepest descent with momentum for quadratic functions. IEEE Transactions on Neural Networks **13(3)**, 752–756 (2002)
36. Vogl, T.P., Mangis, J., Rigler, A., Zink, W., Alkon, D.: Accelerating the convergence of the back-propagation method. Biological cybernetics **59**(4-5), 257–263 (1988)
37. Worthington, P.L., Hancock, E.R.: Surface topography using shape-from-shading. Pattern Recognition **34(4)**, 823–840 (2001)

## 8 Appendix

See Table 1.

**Table 1** Performance of solvers $\mathcal{S}$ on Ex. 4. All algorithms were stopped after 500$s$.

| S/N | Test Problem | Secant-Based-NGM<br>RT(s) | Adaptive Restart<br>RT(s) | Fixed Restart<br>RT(s) | Classical NGM<br>RT(s) |
|---|---|---|---|---|---|
| 1 | 009 ’BOYD1’ | 4.50 | 5.04 | 66.81 | 67.03 |
| 2 | 017 ’CVXQP1_L’ | 234.41 | 142.38 | 169.84 | 415.49 |
| 3 | 018 ’CVXQP1_M’ | 5.16 | 5.14 | 13.87 | 18.89 |
| 4 | 028 ’DUAL1’ | 0.17 | 0.10 | 0.89 | 1.11 |
| 5 | 032 ’DUALC1’ | 0.61 | 0.79 | 26.58 | 25.95 |
| 6 | 035 ’DUALC8’ | 0.17 | 0.23 | 1.81 | 2.48 |
| 7 | 039 ’GOULDQP3’ | 0.22 | 0.23 | 0.28 | 0.99 |
| 8 | 042 ’HS268’ | 0.53 | 0.65 | 5.95 | 5.86 |
| 9 | 069 ’PRIMAL1’ | 10.09 | 14.76 | 16.78 | 16.53 |
| 10 | 070 ’PRIMAL2’ | 10.96 | 16.09 | 18.26 | 17.96 |
| 11 | 071 ’PRIMAL3’ | 11.28 | 16.50 | 18.81 | 18.50 |
| 12 | 072 ’PRIMAL4’ | 13.44 | 19.61 | 22.38 | 22.14 |
| 13 | 073 ’PRIMALC1’ | 9.78 | 14.51 | 16.52 | 16.26 |
| 14 | 074 ’PRIMALC2’ | 9.73 | 14.27 | 16.19 | 15.95 |
| 15 | 075 ’PRIMALC5’ | 10.01 | 14.63 | 16.60 | 16.40 |
| 16 | 076 ’PRIMALC8’ | 10.62 | 15.58 | 17.71 | 17.41 |
| 17 | 079 ’QAFIRO’ | 32.65 | 58.93 | 54.56 | 53.63 |
| 18 | 080 ’QBANDM’ | 47.61 | 44.87 | 43.45 | 42.75 |
| 19 | 081 ’QBEACONF’ | 500.00 | 500.00 | 227.99 | 224.73 |
| 20 | 083 ’QBRANDY’ | 58.08 | 55.81 | 53.99 | 53.35 |
| 21 | 084 ’QCAPRI’ | 78.65 | 74.56 | 72.25 | 71.33 |
| 22 | 085 ’QE226’ | 82.19 | 500.00 | 388.40 | 383.64 |
| 23 | 087 ’QFFFFF80’ | 333.21 | 318.67 | 355.52 | 345.00 |
| 24 | 088 ’QFORPLAN’ | 160.76 | 152.01 | 147.27 | 145.41 |
| 25 | 090 ’QGROW15’ | 203.03 | 500.00 | 340.63 | 335.75 |
| 26 | 091 ’QGROW22’ | 264.73 | 250.82 | 242.93 | 240.55 |
| 27 | 092 ’QGROW7’ | 121.38 | 114.55 | 110.63 | 109.59 |
| 28 | 098 ’QPILOTNO’ | 253.61 | 251.34 | 254.42 | 225.82 |
| 29 | 100 ’QRECIPE’ | 34.76 | 500.00 | 58.01 | 57.20 |
| 30 | 104 ’QSCFXM1’ | 151.47 | 500.00 | 255.69 | 252.71 |
| 31 | 105 ’QSCFXM2’ | 204.93 | 500.00 | 344.88 | 341.66 |
| 32 | 106 ’QSCFXM3’ | 236.10 | 500.00 | 397.89 | 394.56 |
| 33 | 109 ’QSCSD1’ | 87.64 | 500.00 | 147.92 | 145.46 |
| 34 | 110 ’QSCSD6’ | 104.74 | 500.00 | 176.15 | 174.71 |
| 35 | 111 ’QSCSD8’ | 189.45 | 179.14 | 173.31 | 171.69 |
| 36 | 114 ’QSCTAP3’ | 500.00 | 500.00 | 500.00 | 408.50 |
| 37 | 117 ’QSHARE2B’ | 88.83 | 82.29 | 79.56 | 78.61 |
| 38 | 128 ’S268’ | 0.53 | 0.65 | 5.94 | 5.84 |
| 39 | 132 ’STCQP1’ | 0.17 | 0.14 | 1.20 | 1.46 |
| 40 | 138 ’ZECEVIC2’ | 18.17 | 28.61 | 30.20 | 29.82 |