Stochastic Line Search Method for Minimax Problems

Alok Kumar* & Nilay Beniwal*
Supervisor: Prof. Ketan Rajawat

Indian Institute of Technology Kanpur
*Bachelors of Science in Mathematics and Scientific Computing

Abstract

The success of adaptive stochastic optimization algorithms for solving problems arising in ML and SP are now widely recognized. Minimax optimization problems occur frequently in a wide range of signal and data processing applications such as fair beamforming, training generative adversarial networks (GANs), and robust machine learning (ML).Through this project we want to extend stochastic line search method to a class of minimax problems. Our aim is to study the convergence of proposed method to minimax optimization problems, i.e., $\min_{x\in X}\max_{y\in Y} H(x,y)$, where $H(.,y)$ can be convex for each y and $H(x,.)$ is concave for each x.
Here we adapt a classical backtracking Armijo line search[3] to the stochastic optimization setting. In our method we assume that the computation of gradients are available upto some dynamically adjusted accuracy with large and fixed probabilities, in contrast to traditional line search which relies on exact computations of the gradient and values of the objective function.

1 Introduction:

In this paper we consider minimax optimization problem of the form $\min_{x\in X}\max_{y\in Y} H(x,y)$, where $H(.,y)$ can be convex for each y and $H(x,.)$ is concave for each x. Convex-concave methods are well understood with many efficient algorithms but theoretical guarantees for minimax problems are lacking . We adapt a classical backtracking Armijo line-search to the stochastic optimization setting to solve the minimax optimization problem. The advantage of using this over traditional line search is that we don't have to know the exact value of the gradients, function value and almost no hyperparameter tuning.

1.1 Related work:

[3] proposes an adaptive backtracking line-search method, where the sample sizes for gradient and function estimates are chosen adaptively using knowable quantities along with the step-size. They show that this method converges to the optimal solution with probability one and derive strong convergence rates that match those of the deterministic gradient descent methods in the non-convex $O(\varepsilon^{-2})$, convex $O(\varepsilon^{-1})$, and strongly convex $O(\log(\varepsilon^{-1}))$ cases. This paper offers the first stochastic line search method with convergence rates analysis, and is the first to provide convergence rates analysis for adaptive sample size selection based on knowable quantities.
[2] provides a new algorithm combining Mirror-Prox and Nesterov's AGD for solving minimax problem , and show that it can find optimum in different cases as follows-

(i). $1/k^2$ convergence rate for smooth, strongly-convex - concave problems, improving upon the previous best known rate of $1/k$

(ii). $1/k^{1/3}$ convergence rate for smooth, nonconvex - concave problems, improving upon the previous best known rate of $1/k^{1/5}$

1.2 Our contribution:

In this work we extended Armijo Line Search method to the minimax problem in two ways ,

(i) Using Armijo Line Search for taking multiple steps with respect to x till convergence, then go to y and apply Armijo Line Search till convergence, and then go back to x, and so on.

(ii). Via a coupled algorithm where we take one step of Armijo line search in each iteration for each iterate.

We also devised a well defined algorithm to choose sample size for a given iteration of the algorithm.

2 Problem Formulation:

The minimax optimization problem is defined as,

$$\mathcal{P}_E : \min_{x\in\mathcal{X}} \max_{y\in Y} H(x,y), \quad H : \mathcal{X}\times\mathcal{Y} \to R$$
$$H(x,y) := E[h_\xi(x,y)]$$

where $\xi \in \mathcal{X}\times\mathcal{Y}$ is random, and $h_\xi(x,y)$ are loss functions on the data points.

3 Line Search Method:

Consider the classical stochastic optimization problem,

$$min_{x\in R^n}\left\{ f(x) = E[\tilde{f}(x;\xi)] \right\}$$

where ξ is a random variable obeying some distribution. In the case of empirical risk minimization with a finite training set, ξ_i is a random variable that is defined by a single random sample drawn uniformly from the training set. More generally may represents a sample or a set of samples drawn from the data distribution.

Assumption 3.1. *We assume that all iterates x_k of Algorithm 1 satisfy $x_k \in \Omega$ where ,Ω is a set in R^n . Moreover, the gradient of f is L-Lipschitz continuous for all $x \in \Omega$ and that*

$$f_{min} \leq f(x), \forall x \in \Omega.$$

Assumption 3.2. *Suppose in addition to Assumption 3.1, f is convex. Let x^* denote the global minimizer of f and $f^* = f(x^*)$. We assume there exists a constant D such that*

$$\| x - x^* \| \leq D \quad \textit{for all } x \in \Omega,$$

where Ω is the set that contains all iteration realizations as stated in Assumption 3.1. Moreover, we assume there exist a $L_f > 0$ such that $\| \nabla f(x) \| \leq L_f$ for all $x \in \Omega$.

Armijo Line search condition for an objective function is given by,

$$f(x_k + \alpha d_k) \leq f_k - \theta\alpha \| \nabla f(x_k) \|^2$$

where θ is constant($\theta \in (0,1)$) and $\mathrm{d}_k = -\nabla f(x_k)$.

Algorithm 1 ArmijoLS(f, x_0) //Armijo Line Search Method

Input: x_0, f
Output: x_T
Initialization: Choose constants $\gamma > 1, \theta \in (0,1)$, δ_0 and α_{max}. Define $\alpha_0 = \gamma^{j_0}\alpha_{max}$ for some $j_0 \leq 0$.
for $k = 1, 2, \ldots, T$ **do**
  Compute Gradient Estimate g_k
  Calculate Step Length $s_k = -\alpha_k g_k$
  Compute Function Estimates f_k^0 and f_k^s of $f(x_k)$ and $f(x_k + s_k)$ resp.
  if $f_k^s \leq f_k^0 - \alpha_k\theta \| g_k \|^2$ ***then***
    $x_{k+1} = x_k - \alpha_k g_k$
    $\alpha_{k+1} = min\{\alpha_{max}, \gamma\alpha_k\}$
    if $\alpha_k \| g_k \|^2 \geq \delta_k^2$ ***then*** *// (Reliable Step)*
      $\delta_{k+1}^2 = \gamma\delta_k^2$
    else *// (Unreliable Step)*
      $\delta_{k+1}^2 = \gamma^{-1}\delta_k^2$
    end if
  else
    $x_{k+1} = x_k$
    $\alpha_{k+1} = \gamma^{-1}\alpha_k$
    $\delta_{k+1}^2 = \gamma^{-1}\delta_k^2$
  end if
end for
return x_T

In each iteration, Algorithm 1 requires to compute a random direction g_k which can be obtained by a mini-batch stochastic gradient estimate or sampling the function f(x) itself and using finite differences. But, we resort to former method for computation of random direction g_k. Then, we compute stochastic function estimates at the current iterate and prospective new iterate, respectively f_k^0 and f_k^s. We check the Armijo condition using the stochastic estimates

$$f_k^s \leq f_k^0 - \alpha_k\theta \| g_k \|^2$$

If above holds, the next iterate becomes $x_{k+1} = x_k - \alpha_k g_k$ and stepsize α_k increases; otherwise $x_{k+1} = x_k$ and α_k decreases. Unlike classical back-tracking line search, in Algorithm 1 there is an additional control, δ_k, which acts as a guess of the true function decrease and controls the accuracy of the function estimates.

Notation and definitions - Algorithm 1 generates a random process $\{X_k, G_k, F_k^0, F_k^s, \mathcal{A}_k, \Delta_k, S_k\}$, from now on we will denote all random quantities by capital letters and their realization by small letters. Hence random gradient estimate is denoted by G_k and its realizations by $g_k = G_k(\omega)$. Similarly, let the random quantities $x_k = X_k(\omega)$ (iterates), $\alpha_k = \mathcal{A}_k(\omega)$ (stepsize), $\delta_k = \Delta_k(\omega)$(control size) and $s_k = S_k(\omega)$ (step) denote their respective realizations. Similarly, we let $\{F_k^0, F_k^s\}$ denote estimates of $f(X_k)$ and $f(X_k + S_k)$, with their realizations denoted by $f_k^0 = F_k^0(\omega)$ and $f_k^s = F_k^s(\omega)$. Our goal is to show that under some assumptions on G_k and $\{F_k^0, F_k^s\}$ the resulting stochastic process converges with probability one and at an appropriate rate. In particular, we assume that the estimates G_k and F_k^0 and F_k^s are sufficiently accurate with sufficiently high probability, conditioned on the past. To formalize past conditioning,
let $\mathcal{F}_{k-1}^{G,F}$ denote σ-algebra generated by the random variables G_0, G_1, ..., G_{k-1} and F_0^0, F_0^s, F_1^0, F_1^s, ..., F_{k-1}^0, F_{k-1}^s and $\mathcal{F}_{k-1/2}^{G,F}$ denote σ-algebra generated by the random variables G_0, G_1, ..., G_{k-1}, G_k and F_0^0, F_0^s, F_1^0, F_1^s, ... F_{k-1}^0, F_{k-1}^s (for completeness we set $\mathcal{F}_{-1}^{G,F} = \sigma(x_0)$). By the construction of random variables X_k and $\mathcal{A}_k$ in Algorithm 1 , it can be seen that $\mathbf{E}[X_k|\mathcal{F}_{k-1}^{G,F}] = X_k$ and $\mathbf{E}[\mathcal{A}_k|\mathcal{F}_{k-1}^{G,F}] = \mathcal{A}_k$ for all $k \geq 0$.

To measure the accuracy of the gradient and function estimates we have to use following definitions -

Definition 3.1. *A sequence of random directions $\{G_k\}$ is p_g-probabilistically k_g-sufficiently accurate for Algorithm 1 for the corresponding sequence of $\{\mathcal{A}_k, X_k\}$,if there exist a positive constant k_g , such that event*

$$I_k = \{\| G_k - \nabla f(X_k) \| \leq k_g \mathcal{A}_k \| G_k \|\}$$

satisfy the condition

$$\boldsymbol{Pr}(I_k|\mathcal{F}_{k-1}^{G.K}) = \boldsymbol{E}[1_{I_k}|\mathcal{F}_{k-1}^{G.K}] \geq p_g$$

Definition 3.2. *A sequence of random estimates $\{F_k^0, F_k^s\}$ is said to be p_f- probabilistically ϵ_f accurate if with respect to the corresponding sequence $\{X_k, \mathcal{A}_k, S_k\}$ if the event*

$$J_k = \{|F_k^0 - f(x_k)| \leq \epsilon_f \mathcal{A}_k^2 \| G_k \|^2 and \ |F_k^s - f(x_k + s_k)| \leq \epsilon_f \mathcal{A}_k^2 \| G_k \|^2\}$$

satisfy the condition

$$\boldsymbol{Pr}(J_k|\mathcal{F}_{k-1/2}^{G.K}) = \boldsymbol{E}[1_{J_k}|\mathcal{F}_{k-1/2}^{G.K}] \geq p_f$$

Assumption 3.3. *The following hold for the quantities in the Algorithm 1 :*

(i) The sequence of random gradients G_k generated by Algorithm 1 is p_g-probabilistically k_g sufficiently accurate for some sufficiently large $p_g \in$ (0, 1].

(ii) The sequence of estimates $\{F_k^0, F_k^s\}$ generated by Algorithm 1 is p_f -probabilistically ϵ_f - accurate estimates for some $\epsilon_f \leq \frac{\theta}{4\alpha_{max}}$ and sufficiently large $p_f \in$ (0, 1].

(iii) The sequence of estimates $\{F_k^0, F_k^s\}$ generated by Algorithm 1 satisfies a k_f -variance condition for all $k \geq 0$,

$$\boldsymbol{E}[|F_k^s - f(X_k + S_k)|^2|\mathcal{F}_{k-1/2}^{G.F}] \leq max\{k_f^2 \mathcal{A}_k^2 \| \nabla f(X_k) \|^4, \theta^2 \Delta_k^4\}$$
$$and \ \boldsymbol{E}[|F_k^0 - f(X_k)|^2|\mathcal{F}_{k-1/2}^{G.F}] \leq max\{k_f^2 \mathcal{A}_k^2 \| \nabla f(X_k) \|^4, \theta^2 \Delta_k^4\}$$

Assumption 3.4. Suppose there exists a constant $\delta_{\max}$ such that the random variable $\Delta_k \leq \delta_{\max}$. Also, the dynamics of the algorithm suggest Δ_k eventually decreases until it is smaller than any $\varepsilon > 0$.

3.1 Computing G_k, F_0^k, and F_s^k

Assuming that the variance of random function and gradient realizations is bounded as

$$\mathbf{E}(\| \nabla \tilde{f}(x;\xi) - \nabla f(x) \|^2) \leq V_g \text{ and } \mathbf{E}(|\tilde{f}(x;\xi) - f(x)|^2) \leq V_f$$

Assumptions 3.3 can be made to hold if G_k, F_k^0 and F_k^s are computed using a sufficient sample-size. In particular, let S_k be a sample of realizations $\nabla f(x, \xi_i), i \in S_k$ and $G_k := \frac{1}{|S_k|} \sum_{i \in S_k} \nabla \tilde{f}(X_k, \xi_i)$, then for

$$|S_k| \geq \tilde{O}(\frac{V_g}{k_g^2 \mathcal{A}_k^2 \| G_k \|^2}) \tag{1}$$

(where $\tilde{O}$ hides the log factor of $\frac{1}{(1-p_g)}$), Assumption 3.3(i) is satisfied. While G_k is not known when $|S_k|$ is chosen, one can design a simple loop by guessing the value of $\| G_k \|$ and increasing

Algorithm 2 Sample Size Selection

Input: V_g, k_g, p_g, A_k
Output: $|S_k|$
Initialization: Guess the value of $\| G_k \|$
$|S_k| = \log\left(\frac{1}{(1-p_g)}\right)\left(\frac{V_g}{k_g^2 \mathcal{A}_k^2 \|G_k\|^2}\right)$
while $|S_k| < \log\left(\frac{1}{(1-p_g)}\right)\left(\frac{V_g}{k_g^2 \mathcal{A}_k^2 \|G_k\|^2}\right)$ **do:**
    $|S_k| = log\left(\frac{1}{(1-p_g)}\right)\left(\frac{V_g}{k_g^2 \mathcal{A}_k^2 \|G_k\|^2}\right)$
    Recalculate $\| G_k \|$ using $|S_k|$ as sample size
end while
return $|S_k|$

the number of samples until (1) is satisfied, the well-defined procedure is given by Algorithm 2.
Similarly, to satisfy Assumption 3.3(ii), it is sufficient to compute,

$$F_k^0 = \frac{1}{|S_k^0|} \sum_{i \in S_k^0} \tilde{f}(X_k, \xi_i) \text{ with } |S_k^0| \geq \tilde{O}(\frac{V_f}{k_f^2 \mathcal{A}_k^2 \parallel G_k \parallel^4})$$

(where $\tilde{O}$ hides the log factor of $\frac{1}{(1-p_f)}$) and to obtain F_k^s analogously. Finally, it is easy to see that Assumption 3.3(iii) is simply satisfied if $|S_k^0| \geq \frac{V_f}{\theta^2 \Delta_k^4}$ by standard properties of variance.

4 Convex - Concave setting:

Definition 4.1. *: A function H(x,y) is said to be L-smooth if:*

$$max\{\| \nabla_x H(x,y) - \nabla_x H(x',y') \|, \| \nabla_y H(x,y) - \nabla_y H(x',y') \|\} \leq L(\| x - x' \| + \| y - y' \|) \quad (2)$$

The following holds for convex $H(.,y) \forall y \in \mathcal{Y}$,

$$min_{x \in \mathcal{X}} \ H(x, \hat{y}) \leq H(\hat{x}, \hat{y}) \leq max_{y \in \mathcal{Y}} H(\hat{x}, y)$$

Which then implies that, $max_{y \in \mathcal{Y}} min_{x \in \mathcal{X}} H(x,y) \leq min_{x \in \mathcal{X}} max_{y \in \mathcal{Y}} H(x,y)$.
The equality holds according to the minimax theorem for the convex-concave setting when $\mathcal{Y}$ is a compact set, i.e., $max_{y \in \mathcal{Y}} min_{x \in \mathcal{X}} H(x,y) = min_{x \in \mathcal{X}} max_{y \in \mathcal{Y}} H(x,y)$. Furthermore, any point (x^*, y^*) is an optimal solution to minimax problem defined in section 2 if and only if :

$$min_{x \in \mathcal{X}} H(x, y^*) = H(x^*, y^*) = max_{y \in \mathcal{Y}} H(x^*, y)$$

Algorithm 3 defined below uses Armijo Line Search method for taking multiple steps with respect to x till convergence, then go to y and apply Armijo Line Search till convergence, and then go back to x, and so on.

Algorithm 4 defined below is a slight modification of Algorithm 1 . It takes only one step of the method described in Algorithm 1. It is a helper Algorithm for the coupled algorithm described in Algorithm 5 for solving the minimax problem.

Algorithm 5 defined below is a Coupled Algorithm which take a single step in x and uses the updated x to get new y and then single step using the updated y to get the next x and so on. Here steps are taken using Algorithm 4.

Algorithm 3 Stochastic line search Minimax optimization

Input: H
Output: x_{K+1}, y_{K+1}
Initialization: x_0, y_0
for $t = 0, 2, \ldots, T$ **do**
    $f_y \leftarrow -H(x_t, .)$
    $y_{t+1} \leftarrow ArmijoLS(f_y, y_t)$
    $f_x \leftarrow H(., y_{t+1})$
    $x_{t+1} \leftarrow ArmijoLS(f_x, x_t)$
end for
return x_{K+1}, y_{K+1}

Algorithm 4 UniStepArmijoLS(f, $curr$, α_{curr}, α_{max}, δ_{curr}, γ, θ)

Input: f, $curr$, α_{curr}, δ_{curr}
Output: $next$, α_{next}, δ_{next}
Compute Gradient Estimate g_{curr}
Calculate Step Length $s_{curr} = \alpha_{curr} g_{curr}$
Compute Function Estimates f^0_{curr} and f^s_{curr} of $f(curr)$ and $f(curr + s_{curr})$ resp.
if $f^s_{curr} \leq f^0_{curr} - \alpha_{curr}\theta \| g_{curr} \|^2$ ***then***
    $next = curr - \alpha_{curr} g_{curr}$
    $\alpha_{next} = min\{\alpha_{max}, \gamma\alpha_{curr}\}$
    if $\alpha_{curr} \| g_{curr} \|^2 \geq \delta^2_{curr}$ ***then***
        $\delta^2_{next} = \gamma \delta^2_{curr}$
    else
        $\delta^2_{next} = \gamma^{-1} \delta^2_{curr}$
    end if
else
    $next = curr$
    $\alpha_{next} = \gamma^{-1} \alpha_{curr}$
    $\delta^2_{next} = \gamma^{-1} \delta^2_{curr}$
end if
return $next$, α_{next}, δ_{next}

Algorithm 5 Coupled Stochastic Line Search Minimax optimization

Input: H
Output: x_{T+1}, y_{T+1}
Initialization: x_0, y_0, α^x_0, α^y_0, α^x_{max}, α^y_{min}, δ^x_0, δ^y_0, $\gamma > 1$, θ
for $t = 0, 1, \ldots, T$ **do**
    $f_y \leftarrow -H(x_t, .)$
    $y_{t+1}, \alpha^y_{t+1}, \delta^y_{t+1} \leftarrow UniStepArmijoLS(f_y, y_t, \alpha^y_t, \alpha^y_{max}, \delta^y_t, \gamma, \theta)$
    $f_x \leftarrow H(., y_{t+1})$
    $x_{t+1}, \alpha^x_{t+1}, \delta^x_{t+1} \leftarrow UniStepArmijoLS(f_x, x_t, \alpha^x_t, \alpha^x_{max}, \delta^x_t, \gamma, \theta)$
end for
return x_{T+1}, y_{T+1}

5 Convergence Analysis:

In Algorithm 3 ,for an iteration t, the difference can be broken down to

$$\begin{aligned} H(x^*, y_t) - H(x_{t-1}, y^*) &= H(x^*, y_t) + H(x_{t-1}, y_t) - H(x_{t-1}, y_t) - H(x_{t-1}, y^*) \\ &= H(x^*, y_t) - H(x_{t-1}, y_t) + H(x_{t-1}, y_t) - H(x_{t-1}, y^*) \end{aligned}$$

Where $H(x^*, y_t)$ is the optima of the function with respect to one variable x, given y_t is fixed, i.e,

$$H(x^*, y_t) = \underset{x \in \mathcal{X}}{min} H(x, y_t)$$

Similarly,

$$H(x_t, y^*) = \underset{y \in \mathcal{Y}}{max} H(x_t, y)$$

Convergence of an algorithm in stochastic condition will require convergence of

$$\underset{t=1,2...T}{min} \mathbf{E}[H(x^*, y_t) - H(x_t, y^*)] \tag{3}$$

This expectation (3) can be divided into two separate expectations as follows-

$$\mathbf{E}[H(x^*, y_t) - H(x_t, y^*)] = \mathbf{E}[H(x^*, y_t) - H(x_{t-1}, y_t)] + \mathbf{E}[H(x_{t-1}, y_t) - H(x_t, y^*)] \tag{4}$$

We base our proof of convergence on properties of the random function

$$\Phi_k = \nu(f(X_k) - f_{\min}) + (1-\nu)\frac{1}{L^2}\mathcal{A}_k \parallel \nabla f(X_k) \parallel^2 + (1-\nu)\theta {\Delta_k}^2. \tag{5}$$

From Theorem 9.1 we can write,

$$\mathbf{E}[\Phi_{k+1} - \Phi_k | \mathcal{F}_{k-1}^{G.F}] \leq -\frac{p_g p_f (1-\nu)(1-\gamma^{-1})}{4} \left(\frac{\mathcal{A}_k}{L^2} \parallel \nabla f(X_k) \parallel^2 + \theta {\Delta_k}^2 \right)$$

Telescopic sum of above will lead to convergence of required expectation,

$$\begin{aligned} \sum_{k=1}^{T} \mathbf{E}[\Phi_{k-1} - \Phi_k | \mathcal{F}_T^{G.F}] &\leq \sum_{k=1}^{T} -\frac{p_g p_f (1-\nu)(1-\gamma^{-1})}{4} \left(\frac{\mathcal{A}_k}{L^2} \parallel \nabla f(X_k) \parallel^2 + \theta {\Delta_k}^2 \right) \\ \implies \mathbf{E}[\Phi^* - \Phi_1 | \mathcal{F}_T^{G.F}] &\leq -\frac{p_g p_f (1-\nu)(1-\gamma^{-1})}{4} \left(\sum_{k=1}^{T} \frac{\mathcal{A}_k}{L^2} \parallel \nabla f(X_k) \parallel^2 + \sum_{k=1}^{T} \theta {\Delta_k}^2 \right) \end{aligned}$$

As everything on the RHS is bounded (using 9.1), we have,

$$\mathbf{E}[\Phi_1 - \Phi^* | \mathcal{F}_T^{G.F}] < \infty$$

$$\implies \mathbf{E}[f(X_1) - f_{\min}] < \infty \tag{6}$$

As inequality (6) holds for both the Lines 6 & 8 in the Algorithm 3. Therefore, expectations $\mathbf{E}[H(x^*, y_t) - H(x_{t-1}, y_t)]$ and $\mathbf{E}[H(x_{t-1}, y_t) - H(x_t, y^*)]$ are bounded. Hence, using equation (4), $\mathbf{E}[H(x^*, y_t) - H(x_t, y^*)]$ is also bounded.
Now, in Algorithm 3, each iteration at worst will take $\mathcal{O}(\frac{1}{\epsilon})$ time and at max there are $\mathcal{O}(\frac{1}{\epsilon})$ iterations, so the worst case complexity bound is $\mathcal{O}(\frac{1}{\epsilon^2})$.

Paper [3] proves the convergence of Algorithm 1 by showing a lim-inf convergence result, i.e., $\liminf_{k\to\infty} \| \nabla f(X_k) \| = 0$. The typical convergence results for stochastic algorithms prove either high probability results or that the expected gradient at an averaged point converges. Showing the convergence of a sub-sequence of the $\| \nabla f(X_k) \|$ which is done in [3] is slightly stronger than previous results. With this convergence result, stopping times based on either $\| \nabla f(x) \| < \epsilon$ and/or $f(x) - f_{min} < \epsilon$ are finite almost surely.

Let's define, $f_x := H(., y)$

$f_y := H(x, .)$

Suppose $H(x, y)$ can be decomposed as, $H(x, y) = H_x + H_{xy} + H_y$
where H_x is a function in only x , H_y is a function in only y and H_{xy} is a function of both x and y
If $H_{xy} = 0$, then

$$H(x, y) = H_x + H_y \tag{7}$$

$\implies \nabla_x H \& \nabla_y H$ can be treated independently for iterations of Algorithm 5

Thus, no coupling effect come into the picture of analysis of two stochastic processes, $\{X_k, G_k^x, F_k^{x,0}, F_k^{x,s}, \mathcal{A}_k^x, \Delta_k^x\}$ & $\{Y_k, G_k^y, F_k^{y,0}, F_k^{y,s}, \mathcal{A}_k^y, \Delta_k^y\}$ generated by Algorithm 5.

Theorem 5.1. ***:*** *Let the assumptions 3.1 & 3.3 hold. Then the sequence of random iterates generated by Algorithm 5, X_k, Y_k almost surely satisfy,*

$$\liminf_{k\to\infty} \| \nabla f_x(X_k) \| = 0 = \liminf_{k\to\infty} \| \nabla f_y(Y_k) \|$$

Proof. Let us assume that there exists constants $\mathcal{E}_1(\omega) > 0$, $\mathcal{E}_2(\omega) > 0$ and $K_0^1(\omega)$, $K_0^2(\omega)$ such that,

$$\| \nabla f_x(X_{k_1}) \| > \mathcal{E}_1, \qquad \forall k_1 \geq K_0^1$$

$$\| \nabla f_y(Y_{k_2}) \| > \mathcal{E}_2, \qquad \forall k_2 \geq K_0^2$$

holds with some positive probability.
Because of Corollary 9.1, we have that,

$$\mathcal{A}_{k_1} \| \nabla f_x(X_{k_1}) \|^2 \to 0 \quad \& \quad [\mathcal{A}_{k_2} \| \nabla f_y(Y_{k_2}) \|^2 \to 0$$

Hence, we have

$$\mathbf{Pr}\left(\left\{\omega : \| \nabla f_x(X_{k_1}) \| > \mathcal{E}_1 \text{ for all } k_1 \geq K_0^1 \text{ and } \lim_{k_1\to\infty} \mathcal{A}_{k_1} \| \nabla f_x(X_{k_1}) \|^2 = 0\right\}\right) > 0$$
$$\& \ \mathbf{Pr}\left(\left\{\omega : \| \nabla f_y(Y_{k_2}) \| > \mathcal{E}_2 \text{ for all } k_2 \geq K_0^2 \text{ and } \lim_{k_2\to\infty} \mathcal{A}_{k_2} \| \nabla f_y(Y_{k_2}) \|^2 = 0\right\}\right) > 0$$

Let $\{x_{k_1}\}$, $\{\alpha_{k_1}\}$, ϵ_1, and k_0^1 be the realization of $\{X_{k_1}\}, \{\mathcal{A}_{k_1}\}, \mathcal{E}_1$, and K_0^1, respectively , for which,

$$\| \nabla f_x(x_{k_1}) \| > \epsilon_1 \text{ for all } k_1 \geq K_0^1 \text{ and } \lim_{k_1\to\infty} \alpha_{k_1} \| \nabla f_x(x_{k_1}) \|^2 = 0$$

Let $\{y_{k_2}\}$, $\{\alpha_{k_2}\}$, ϵ_2, and k_0^2 be the realization of $\{Y_{k_2}\}, \{\mathcal{A}_{k_2}\}, \mathcal{E}_2$, and K_0^2, respectively , for which,

$$\| \nabla f_y(y_{k_1}) \| > \epsilon_2 \text{ for all } k_2 \geq K_0^2 \text{ and } \lim_{k_2\to\infty} \alpha_{k_2} \| \nabla f_y(y_{k_2}) \|^2 = 0$$

An immediate consequence is that $\alpha_1 \to 0$ & $\alpha_2 \to 0$. Consequently , we deduce that

$$0 < \mathbf{Pr}\left(\left\{\omega : \| \nabla f_x(X_{k_1}) \| > \mathcal{E}_1 \text{ for all } k_1 \geq K_0^1 \text{ and } \lim_{k_1 \to \infty} \mathcal{A}_{k_1} \| \nabla f_x(X_{k_1}) \|^2 = 0\right\}\right) \leq \mathbf{Pr}\left(\left\{\omega : \lim_{k_1 \to \infty} \mathcal{A}_{k_1} = 0\right\}\right)$$

$$0 < \mathbf{Pr}\left(\left\{\omega : \| \nabla f_y(Y_{k_2}) \| > \mathcal{E}_2 \text{ for all } k_2 \geq K_0^2 \text{ and } \lim_{k_2 \to \infty} \mathcal{A}_{k_2} \| \nabla f_y(Y_{k_2}) \|^2 = 0\right\}\right) \leq \mathbf{Pr}\left(\left\{\omega : \lim_{k_2 \to \infty} \mathcal{A}_{k_2} = 0\right\}\right)$$

Now, define two new random variables corresponding to random process $\{X_{k_1}, G_{k_1}^x, F_{k_1}^{x,0}, F_{k_1}^{x,s}, \mathcal{A}_{k_1}^x, \Delta_k^x\}$, by $R_{k_1}^1 := \log(\mathcal{A}_{k_1})$ and $Z_{k_1}^1$ defined by the recursion,

$$Z_{k_1+1}^1 := min\Big\{ \log(\bar{\mathcal{A}}_1), 1_{I_{k_1}} 1_{J_{k_1}} (\log(\gamma) + Z_k^1) + (Z_{k_1}^1 - \log(\gamma))(1 - 1_{I_{k_1} J_{k_1}}) \Big\} \text{ and } Z_0^1 = R_0^1 = \log(\alpha_0^1).$$

Also, define two new random variables corresponding to random process $\{Y_{k_2}, G_{k_2}^y, F_{k_2}^{y,0}, F_{k_2}^{y,s}, \mathcal{A}_{k_2}^y, \Delta_{k}^y 2\}$, by $R_{k_2}^2 := \log(\mathcal{A}_{k_2})$ and $Z_{k_2}^2$ defined by the recursion,

$$Z_{k_2+1}^2 := min\Big\{ \log(\bar{\mathcal{A}}_2), 1_{I_{k_2}} 1_{J_{k_2}} (\log(\gamma) + Z_{k_2}^2) + (Z_{k_2}^2 - \log(\gamma))(1 - 1_{I_{k_2} J_{k_2}}) \Big\} \text{ and } Z_0^2 = R_0^2 = \log(\alpha_0^2).$$

By definition, $Z_{k_1}^1$ & $Z_{k_2}^2$ are bounded from below and $R_{k_1}^1$ & $R_{k_2}^2$, by our assumption has a positive probability of diverging to $-\infty$. We establish a contradiction by proving that $R_{k_1}^1 \geq Z_{k_1}^1$ & $R_{k_2}^2 \geq Z_{k_2}^2$, which is what we do below.
Now, consider the random variables $R_{k_1}^1 \& Z_{k_1}^1$.The sequence of random variables increase by $\log(\gamma)$, unless it hits the maximum, with probability $p_f^x p_g^x$ and otherwise decreases by $\log(\gamma)$. Our main argument is to show that $\{\omega : \lim_{k_1 \to \infty} \mathcal{A}_{k_1} = 0\} = \{\omega : \lim_{k_1 \to \infty} R_{k_1}^1 = -\infty\}$ are null set.
By construction, the random variables $R_{k_1}^1$ and $Z_{k_1}^1$ are measurable with respect to the same σ-algebra namely $\mathcal{F}_{k_1-1}^{M.F}$ for $k \geq 0$.We next show that $R_{k_1}^1 \geq Z_{k_1}^1$.
Now we will show only $R_{k_1}^1 \geq Z_{k_1}^1$ by induction (base case is true, as $Z_0^1 = R_0^1$ given in definition) and $R_{k_2}^2 \geq Z_{k_2}^2$ follows by similar arguments.
Without loss of generality assume there exists a j $\in Z$ such that $\gamma^j \alpha_0^1 = \bar{\mathcal{A}}_1$. Assume the induction hypothesis, namely, $R_{k_1}^1 \geq Z_{k_1}^1$.
If $R_{k_1}^1 > \log(\bar{\mathcal{A}}_1)$, then

$$R_{k_1+1}^1 \geq log(\gamma^{-1} \mathcal{A}_{k_1}) = R_{k_1}^1 - log(\gamma) \geq log(\bar{\mathcal{A}}_1) \geq Z_{k_1}^1,$$

where the rightmost inequality follows because the assumption $R_{k_1+1}^1$ strictly larger than $\log(\bar{\mathcal{A}}_1)$ implies that,

$$R_{k_1+1}^1 \geq \log(\bar{\mathcal{A}}_1) + \log(\gamma)$$

For $R_{k_1+1}^1 \leq \log(\bar{\mathcal{A}}_1)$ we consider some cases,

- If $1_{I_{k_1}} 1_{J_{k_1}} = 1$ then by Lemma 9.1 we know,

  $$R_{k_1+1}^1 = \log(\mathcal{A}_{k_1+1}) = \min\{\log(\alpha_{\max}), R_{k_1}^1 + \log(\gamma)\}$$

  Suppose $R_{k_1+1}^1 = \log(\alpha_{\max})$. Then by definition of $\bar{\mathcal{A}}_1$ and $Z_{k_1+1}^1$,

  $$R_{k_1+1}^1 \geq \log(\bar{\mathcal{A}}_1) \geq Z_{K_1+1}^1$$

  On the other hand, suppose $R_{k_1+1}^1 = R_{k_1}^1 + \log(\gamma)$. Then by the induction hypothesis, we have,

  $$R_{k_1+1}^1 \geq Z_{k_1}^1 + \log(\gamma) \geq \min\{\bar{\mathcal{A}}_1, Z_{k_1}^1 + \log(\gamma)\} = Z_{k_1+1}^1$$

- Next, suppose $1_{I_{k_1}} 1_{J_{k_1}} = 0$. It follows that,

$$Z^1_{k_1+1} = Z^1_{k_1} - \log(\gamma) \leq R^1_{k_1} - \log(\gamma) = \log(\mathcal{A}_{k_1}\gamma^{-1}) \leq R^1_{k_1+1}$$

Therefore, we showed that $R^1_{k_1} \geq Z^1_{k_1}$ for all $k_1 \geq 0$. Moreover, we see that $\{Z^1_{k_1}\}$ is a random walk with a maximum and a drift upward. Therefore,

$$1 = \mathbf{Pr}(\limsup_{k_1} Z^1_{k_1} \geq \log(\bar{\mathcal{A}}_1)) = \mathbf{Pr}(\limsup_{k_1} R^1_{k_1} \geq \log(\bar{\mathcal{A}}_1)).$$

However, this contradicts the fact that $\mathbf{Pr}(\omega : (\limsup_{k_1} R^1_{k_1} = -\infty) > 0$.

Thus,

$$\liminf_{k\to\infty} \parallel \nabla f_x(X_k) \parallel = 0 = \liminf_{k\to\infty} \parallel \nabla f_y(Y_k) \parallel$$

□

Now, suppose $\{X_k, G^x_k, F^{x,0}_k, F^{x,s}_k, \mathcal{A}^x_k, \Delta^x_k\}$ & $\{Y_k, G^y_k, F^{y,0}_k, F^{y,s}_k, \mathcal{A}^y_k, \Delta^y_k\}$ are the random processes generated by Algorithm 5 then there exists probabilities p^x_g,p^x_f,p^y_g and p^y_f and constants $\nu_x, \nu^y \in (0,1)$ such that inequality(15) holds for these random processes. Let's define, $p_x := p^x_g \cdot p^x_f$ & $p_y := p^y_g \cdot p^y_f$

Theorem 5.2. *Let the assumptions of Theorem 9.2 hold for the processes $\{X_k, G^x_k, F^{x,0}_k, F^{x,s}_k, \mathcal{A}^x_k, \Delta^x_k\}$ & $\{Y_k, G^y_k, F^{y,0}_k, F^{y,s}_k, \mathcal{A}^y_k, \Delta^y_k\}$ with constants ν_x&ν_y and probabilities $p^x_f p^x_g$&$p^y_f p^y_g$ as in Theorem 9.2. Then the expected number of iterations for problems of form (7) that the Algorithm 5 takes until $f_x(X_k) - f^*_x < \varepsilon_1$ & $f_y(Y_k) - f^*_y < \varepsilon_2$ hold together is bounded as follows*

$$\boldsymbol{E}[T] \leq \mathcal{O}(1) \cdot max\left(\frac{p_x}{2p_x - 1}\left(\frac{L^3_x k^3_{g_x}\left(D^2_1 + L^2_{f_x} + \delta^2_{\max}\right)}{\varepsilon_1}\right), \frac{p_y}{2p_y - 1}\left(\frac{L^3_y k^3_{g_y}\left(D^2_2 + L^2_{f_y} + \delta^2_{\max}\right)}{\varepsilon_2}\right)\right).$$

Proof. Consider the random process $\{X_k, G^x_k, F^{x,0}_k, F^{x,s}_k, \mathcal{A}^x_k, \Delta^x_k\}$, generated by Algorithm 5 with constants ν_x and probabilities $p^x_f p^x_g$. Then by using Theorem 9.3, expected number of iterations, T_{ε_1} to attain $f_x(X_k) - f^*_x < \varepsilon_1$ can be bounded by,

$$\mathbf{E}[T_{\varepsilon_1}] \leq \mathcal{O}(1) \cdot \left(\frac{p_x}{2p_x - 1}\left(\frac{L^3_x k^3_{g_x}\left(D^2_1 + L^2_{f_x} + \delta^2_{\max}\right)}{\varepsilon_1}\right)\right). \tag{8}$$

Again, consider the random process $\{Y_k, G^y_k, F^{y,0}_k, F^{y,s}_k, \mathcal{A}^y_k, \Delta^y_k\}$, generated by Algorithm 5 with constants ν_y and probabilities $p^y_f p^y_g$. Then by using Theorem 9.3,expected number of iterations, T_{ε_2} to attain $f_y(Y_k) - f^*_y < \varepsilon_2$ can be bounded by,

$$\mathbf{E}[T_{\varepsilon_2}] \leq \mathcal{O}(1) \cdot \left(\frac{p_y}{2p_y - 1}\left(\frac{L^3_y k^3_{g_y}\left(D^2_2 + L^2_{f_y} + \delta^2_{\max}\right)}{\varepsilon_2}\right)\right). \tag{9}$$

Since in each iteration of Algorithm 5 random processes $\{X_k, G^x_k, F^{x,0}_k, F^{x,s}_k, \mathcal{A}^x_k, \Delta^x_k\}$ & $\{Y_k, G^y_k, F^{y,0}_k, F^{y,s}_k, \mathcal{A}^y_k, \Delta^y_k\}$ updates one by one. Therefore, the Algorithm 5 terminates when both bounds 8 & 9 hold together. Thus, the expected number of iterations that Algorithm 5 takes can be bounded by,

$$\mathbf{E}[T] \leq \mathcal{O}(1) \cdot max\left(\frac{p_x}{2p_x - 1}\left(\frac{L^3_x k^3_{g_x}\left(D^2_1 + L^2_{f_x} + \delta^2_{\max}\right)}{\varepsilon_1}\right), \frac{p_y}{2p_y - 1}\left(\frac{L^3_y k^3_{g_y}\left(D^2_2 + L^2_{f_y} + \delta^2_{\max}\right)}{\varepsilon_2}\right)\right).$$

□

6 Experiments:

We have tested our proposed method to solve Minimax problem and in this section we briefly describes the performance of the same. Our implemented codes can be found **here**.

6.1 Experiment with Algorithm 1-

Let's first check Algorithm 1 for a simple stochastic minimization problem -
Consider,

$$q(x) = 100 + a.x \text{ , for given positive constant a}$$

let,

$$q_i := q(x_i) + N(0, \sigma^2), \text{ for small } \sigma$$

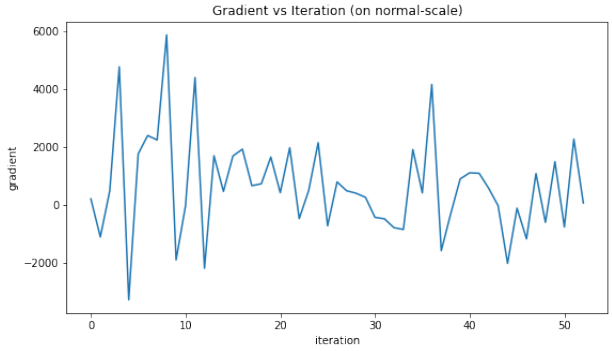
Now fix a = 5 and generate N = 1000 data points, i.e, (x_i, y_i, H_i) , where $1 \leq i \leq N$
define,

$$Err := \frac{\sum_{i=1}^{N} l(x_i, q_i)}{N}$$

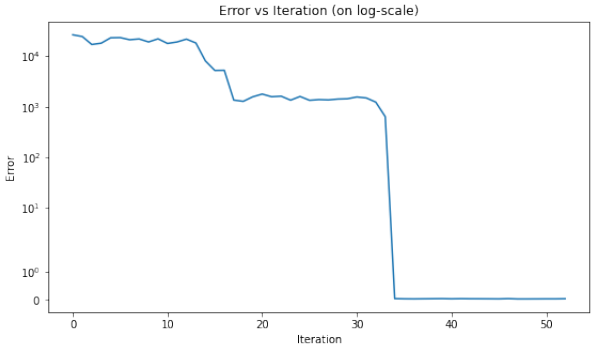
where l is the squared loss function defined as,

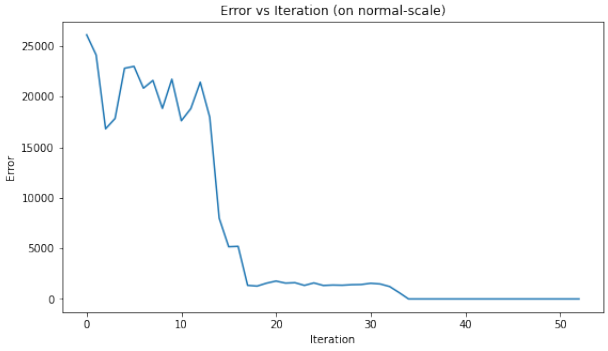
$$l(x_i, q_i) = (q(x_i) - q_i)^2$$

Now we can estimate a by minimising the error, Err.
To minimise the above error function we used the Algorithm 1. And estimated the value of a to be, **4.993793634037582**. The variates associated with the algorithm can be summarised by below plots. The below graph represents the variation of estimated gradients during run of algorithm.

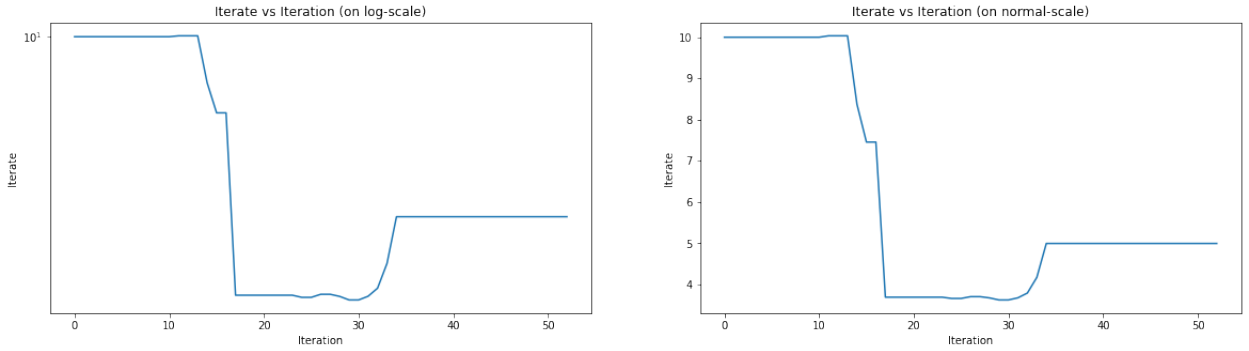


Variation of Error through the course of algorithm,





From above plot it is clear that error tends to zero as algorithm progresses.



Plot of iterate while optimizing, reached the optimum value of 5 (algorithm terminated at $\tilde{\mathbf{4.99}}$).

6.2 Experiments with Algorithm 3-

Consider,

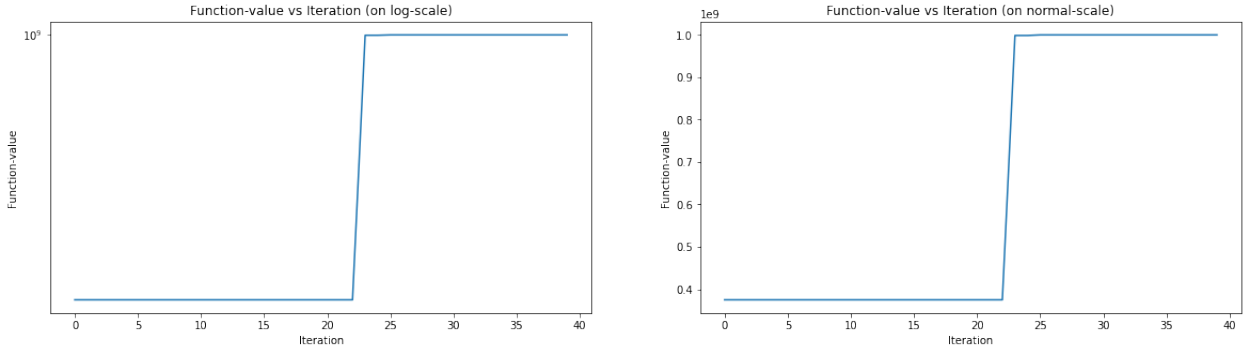
$$Q(x, y) = (1 + a^2x^2) * (m - b^2y^2) \text{ , for given positive constants a, m \& b}$$

let, a = 1, b = 1 ,m=1000 Then,

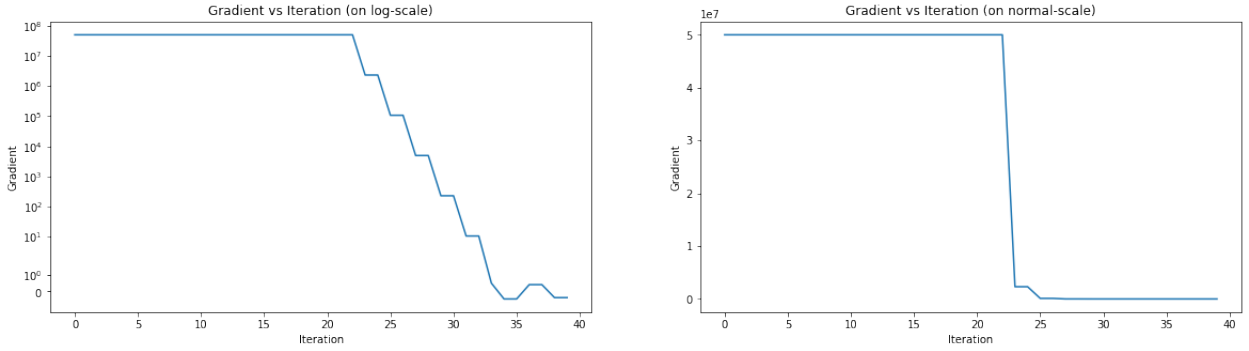
$$H(x, y) = \min_{x} \max_{y} (1 + x^2) * (1000 - y^2)$$

We solved the above problem by Algorithm 3 and obtained the results summarised below.

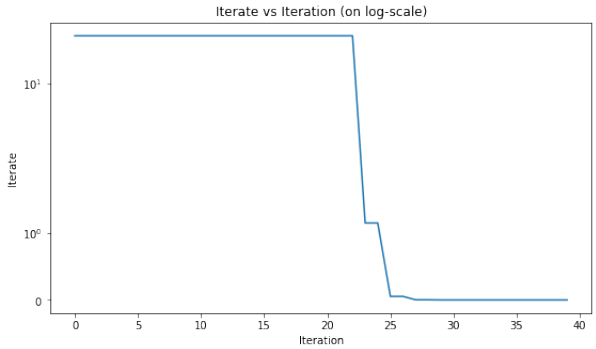
$$x* = 3.884892407768348e^{-09}$$
$$y* = 1.674625858607091e^{-07}$$
$$H(x*, y*) \approx 1000$$

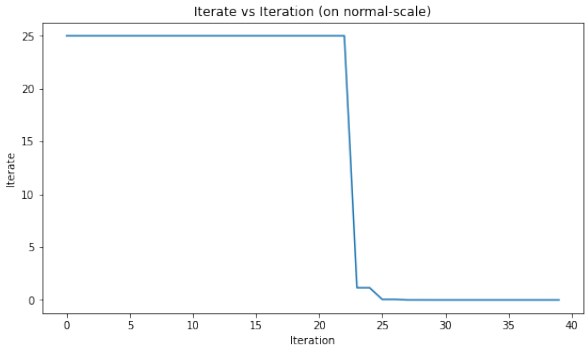


Plot of function value at each iteration reaches the value at optimum in about 25 iterations .

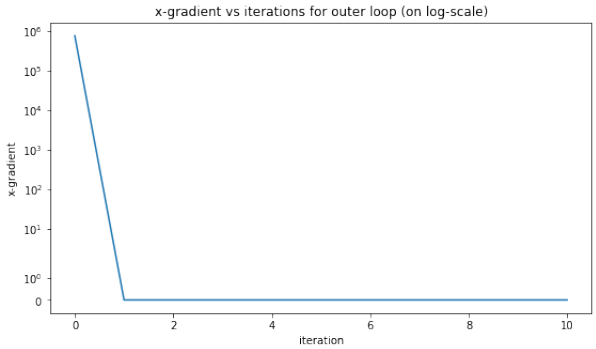


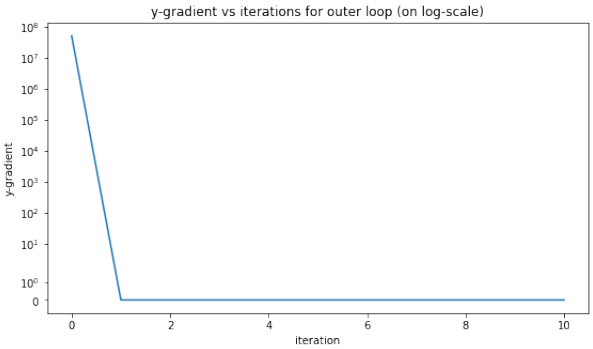
Plot of gradient value at each iteration tending to zero as optima is reached .



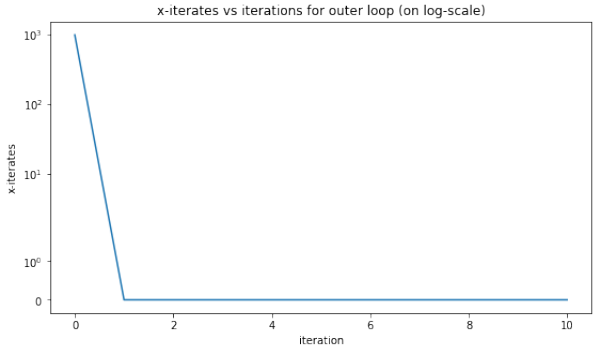


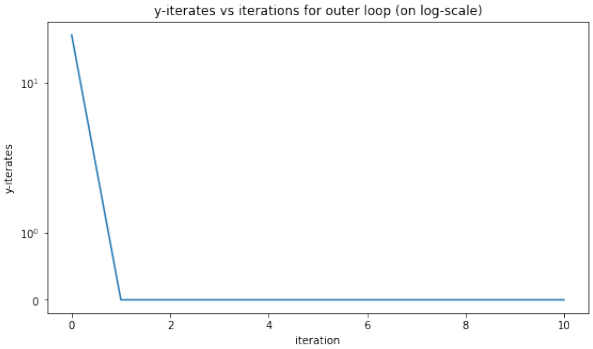
Plot of iterate value x and y in each iteration reaching to the saddle point((0,0)) .



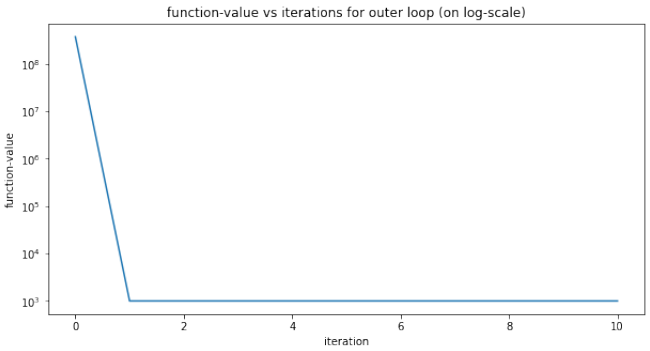


Plot of partial gradient w.r.t x and y in each outer loop iteration, remained almost constant after two iterations.

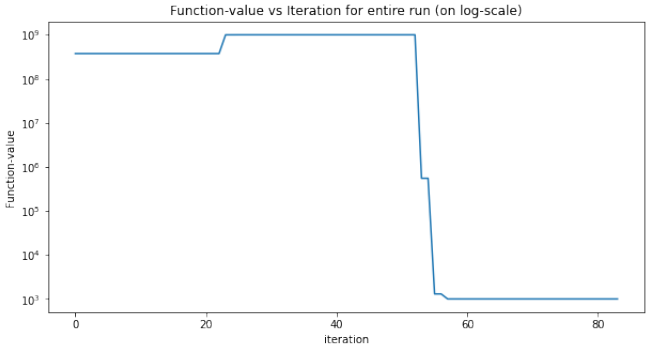




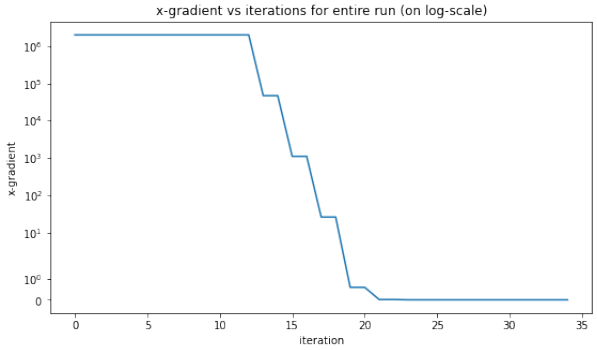
Plot of values of iterate x and y at each outer loop iterations,remained almost constant after two iterations.

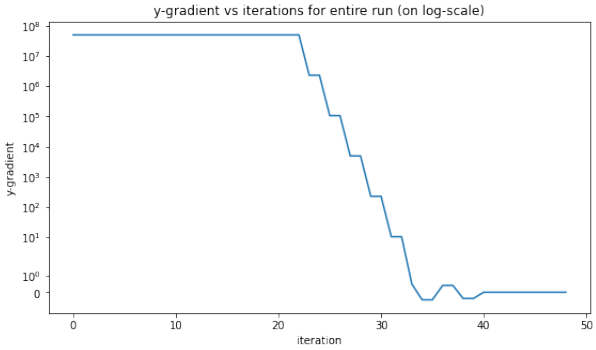


Plot of function value at each iteration of outer loop,remained almost constant after two iterations.

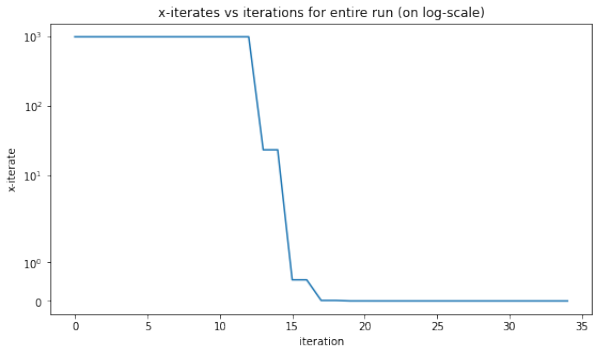


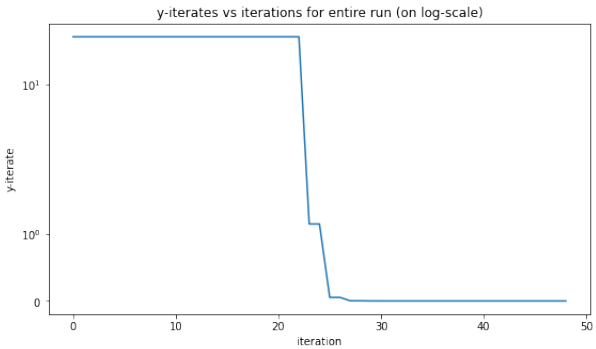
Plot of function value at each iteration of entire run, reached to optimum value of 1000 in about 60 iterations .



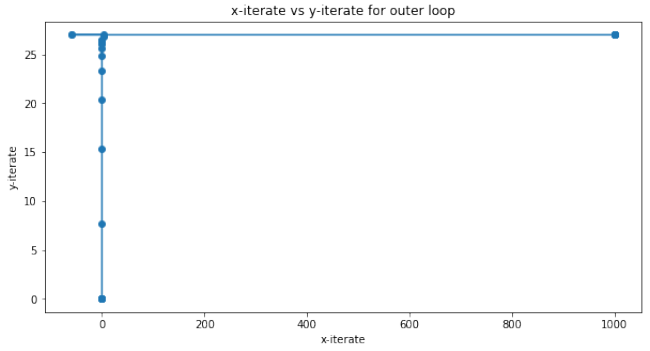


plot of value of Partial gradient w.r.t x and y for the entire run, slowly tending to zero as optima is achieved.





Plot of iterate value x and y for the entire run, reached the saddle point ((0,0)).



Scatter plot of x and y iterate for outer loop.

6.3 Experiments with Algorithm 5-

Consider,

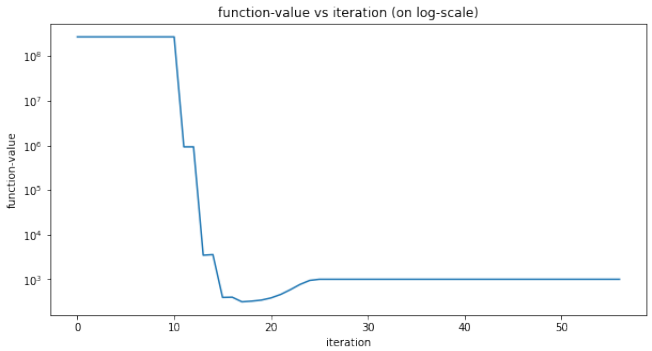
$$Q(x,y) = (1 + a^2x^2) * (m - b^2y^2) \text{ , for given positive constants a, m \& b}$$

let, a = 1, b = 1 ,m=1000

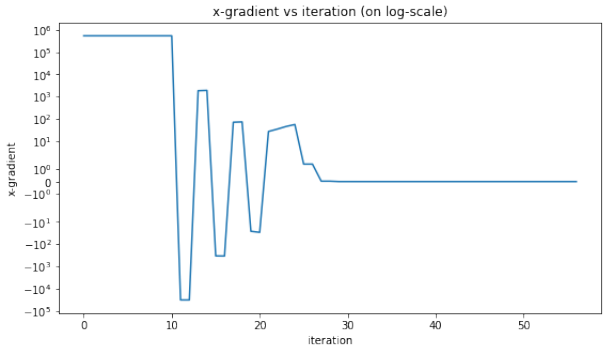
$$H(x,y) = \min_x \max_y (1 + x^2) * (1000 - y^2)$$

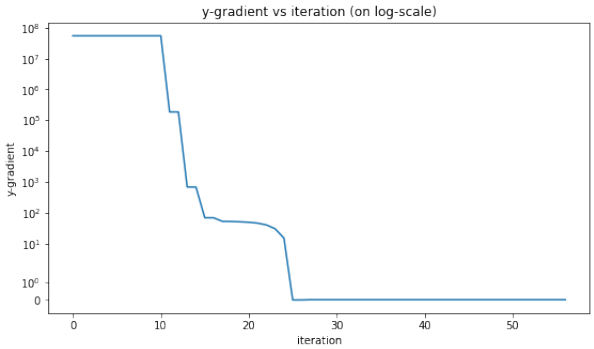
We solved the above problem by Algorithm 5 and obtained the results summarised below.

$$x* = 5.110225008526857e^{-09}$$
$$y* = 2.178559559039861e^{-07}$$
$$H(x*, y*) \approx 1000$$

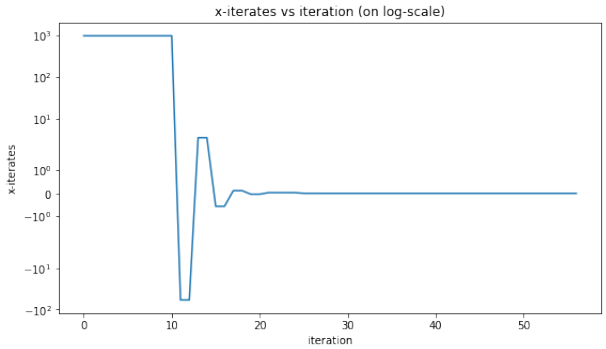


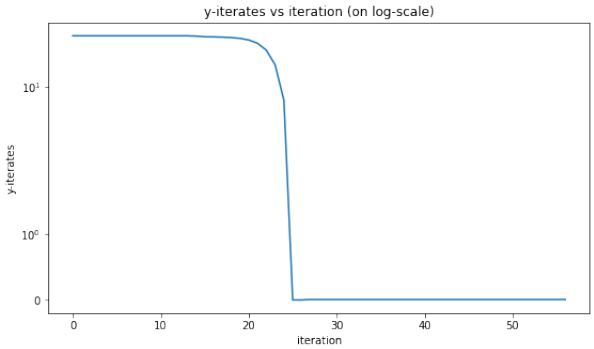
Plot of function value at each iteration, reached the optimum value of 1000 in about 30 iterations.





Plot of value of partial gradient w.r.t x and y, slowly tending to zero as optima is achieved.





Plot of value of x and y iterate in each iteration, reached the saddle point ((0,0)).

7 Conclusions:

- Algorithm 5 works better than Algorithm 3 in terms of number of iterations and oracle calls in the considered problem and both the algorithms converge practically.
- Algorithm 3 works in complexity $\mathcal{O}(\frac{1}{\epsilon^2})$
- Algorithm 3 surely works but there are a lot of unwanted armijoLS steps when we are close to the optima .
- If the mini-max problem is separable in x and y as in 7, then the complexity will be $\mathcal{O}(\frac{1}{\epsilon})$ using Algorithm 5.

8 References:

1. *F. E. Curtis and K. Scheinberg, "Adaptive Stochastic Optimization: A Framework for Analyzing Stochastic Optimization Algorithms," in IEEE Signal Processing Magazine, vol. 37, no. 5, pp. 32-42, Sept. 2020, doi: 10.1109/MSP.2020.3003539.*

2. *Kiran Koshy Thekumparampil, Prateek Jain, Praneeth Netrapalli, Sewoong Oh. Efficient Algorithms for Smooth Minimax Optimization. In arXiv:1907.01543v1 (2019)*

3. *Courtney Paquette,Katya Scheinberg,"A Stochastic Line Search Method with Convergence Rate Analysis". arXiv:1807.07994*

4. *C. Cartis and K. Scheinberg. Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. Mathematical Programming, Apr 2017.*

5. *J. Blanchet, C. Cartis, M. Menickelly, and K. Scheinberg. Convergence Rate Analysis of a Stochastic Trust Region Method for Nonconvex Optimization. arXiv:1609.07428, 2017.*

9 Appendix:

The following theorems,lemmas and corollaries are present in paper[3] which are used very frequently in our proof.

Theorem 9.1. *Let Assumptions 3.1 and 3.3 hold. Suppose* $\{X_k, G_k, F_k^0, F_k^s, \mathcal{A}_k, \Delta_k\}$ *is the random process generated by Algorithm 1. Then there exist probabilities* $p_g, p_f > 1/2$ *and a constant* $\nu \in (0,1)$ *such that the expected decrease in* Φ_k *is*

$$\boldsymbol{E}[\Phi_{k+1} - \Phi_k | \mathcal{F}_{k-1}^{G.F}] \leq -\frac{p_g p_f (1-\nu)(1-\gamma^{-1})}{4} \left(\frac{\mathcal{A}_k}{L^2} \parallel \nabla f(X_k) \parallel^2 + \theta \Delta_k 2 \right). \tag{10}$$

In particular, the constant ν *and probabilities* $p_g, p_f > 1/2$ *satisfy*

$$\frac{\nu}{1-\nu} \geq \max\left\{ \frac{32\gamma\alpha_{\max}2}{\theta}, 16(\gamma-1), \frac{16\gamma(k_g\alpha_{\max}+1)^2}{\theta} \right\}, \tag{11}$$

$$p_g \geq \frac{2\gamma}{1/2(1-\gamma^{-1}) + 2\gamma} \tag{12}$$

$$\textit{and} \qquad \frac{p_g p_f}{\sqrt{1-p_f}} \geq \max\left\{ \frac{8L^2\nu + 16\gamma(1-\nu)}{(1-\nu)(1-\gamma^{-1})}, \frac{8\nu}{(1-\nu)(1-\gamma^{-1})} \right\}. \tag{13}$$

Corollary 9.1. *Let the same assumptions as Theorem 9.1 hold. Suppose* $\{X_k, G_k, F_k^0, F_k^s, \mathcal{A}_k, \Delta_k\}$ *is the random process generated by Algorithm 1. Then we have*

$$\sum_{k=0}^{\infty} \mathbf{E}[\mathcal{A}_k \parallel \nabla f(X_k) \parallel^2] < \infty.$$

Proof. By taking expectations of (10) and summing up, we deduce

$$\tilde{C} \sum_{k=0}^{\infty} \mathbf{E}[\mathcal{A}_k \parallel \nabla f(X_k) \parallel^2] \leq \sum_{k=0}^{\infty} \mathbf{E}[\Phi_k] - \mathbf{E}[\Phi_{k+1}] \leq \Phi_0 < \infty,$$

□

$$\Psi_k = \frac{1}{\nu\varepsilon} - \frac{1}{\Phi_{k \wedge T_\varepsilon}}^1. \tag{14}$$

Theorem 9.2. *Let Assumptions 3.1, 3.2, 3.3 and 3.4 hold. Suppose* $\{X_k, G_k, F_k^0, F_k^s, \mathcal{A}_k, \Delta_k\}$ *is the random process generated by Algorithm 1. Then there exists probabilities* p_g *and* p_f *and a constant* $\nu \in (0,1)$ *such that*

$$1_{\{T_\varepsilon > k\}} \boldsymbol{E}[\Psi_{k+1} - \Psi_k | \mathcal{F}_{k-1}^{G.K}] \leq -\frac{p_g p_f (1-\nu)(1-\gamma^{-1})}{8(\nu DL + \frac{(1-\nu)\alpha_{\max} L_f}{L} + (1-\nu)\sqrt{\theta}\delta_{\max})^2} \mathcal{A}_k 1_{\{T_\varepsilon > k\}} \tag{15}$$

where Ψ_k *is defined above In particular, the probabilities* p_g *and* p_f *and constant* ν *satisfy (11), (12), and (13) from Theorem 9.1.*

Proof. First, by convexity, we have that

$$\begin{aligned}
\Phi_k &= \nu(f(X_k) - f^*) + (1-\nu)|\mathcal{A}_k \tfrac{\|\nabla f(X_k)\|^2}{L^2} + (1-\nu)\theta\Delta_k \\
&\leq \nu \nabla f(X_k), X_k - x^* + (1-\nu)\alpha_{\max} \tfrac{\|\nabla f(X_k)\|^2}{L^2} + (1-\nu)\theta\delta_{\max}\Delta_k \\
&\leq \left(\nu DL + \tfrac{(1-\nu)\alpha_{\max} L_f}{L} + (1-\nu)\sqrt{\theta}\delta_{\max}\right)\left(\tfrac{\|\nabla f(X_k)\|}{L} + \sqrt{\theta}\Delta_k\right),
\end{aligned}$$

where we used $\| \nabla f(X_k) \| < L_f$. Without loss of generality, we assume $\alpha_{\max} \leq 1$; one may prove the same result with any stepsize, but for the sake simplicity we will defer to the standard case when $\alpha_{\max} \leq 1$. By squaring both sides, we conclude

$$\frac{\mathcal{A}_k \Phi_k^2}{\tilde{C}} := \frac{\mathcal{A}_k \Phi_k^2}{2(\nu DL + \frac{(1-\nu)\alpha_{\max} L_f}{L} + (1-\nu)\sqrt{\theta}\delta_{\max})^2} \leq \mathcal{A}_k \frac{\| \nabla f(X_k) \|^2}{L^2} + \theta\Delta_k^2,$$

where we used the inequality $(a+b)^2 \leq 2(a^2+b^2)$. From the above inequality combined with (10) we have

$$\mathbf{E}[1_{\{T_\varepsilon > k\}}(\Phi_{k+1} - \Phi_k) | \mathcal{F}_{k-1}^{G.F}] \leq \frac{-p_g p_f (1-\nu)(1-\gamma^{-1})\mathcal{A}_k \Phi_k^2}{4\tilde{C}} \cdot 1_{\{T_\varepsilon > k\}}.$$

Now using the simple fact that $1_{\{T_\varepsilon > k\}}(\Phi_{k+1} - \Phi_k) = \Phi_{(k+1)\wedge T_\varepsilon} - \Phi_{k \wedge T_\varepsilon}$ we can write

$$\mathbf{E}[\Phi_{(k+1)\wedge T_\varepsilon} - \Phi_{k \wedge T_\varepsilon} | \mathcal{F}_{k-1}^{G.F}] \leq \frac{-p_g p_f (1-\nu)(1-\gamma^{-1})\mathcal{A}_k \Phi_k^2}{4\tilde{C}} \cdot 1_{\{T_\varepsilon > k\}}.$$

[1] We use $a \wedge b = \min\{a, b\}$.

We can then use Jensen's inequality to derive

$$\begin{aligned}\mathbf{E}\left[\frac{1}{\Phi_{k\wedge T_\varepsilon}}-\frac{1}{\Phi_{(k+1)\wedge T_\varepsilon}}\big|\mathcal{F}_{k-1}^{G.F}\right] &\leq \frac{1}{\Phi_{k\wedge T_\varepsilon}}-\frac{1}{\mathbf{E}[\Phi_{(k+1)\wedge T_\varepsilon}|\mathcal{F}_{k-1}^{G.F}]}=\left(\frac{\mathbf{E}[\Phi_{(k+1)\wedge T_\varepsilon}-\Phi_{k\wedge T_\varepsilon}|\mathcal{F}_{k-1}^{G.F}]}{\Phi_{k\wedge T_\varepsilon}\mathbf{E}[\Phi_{(k+1)\wedge T_\varepsilon}|\mathcal{F}_{k-1}^{G.F}]}\right)\\ &\leq \frac{-p_g p_f(1-\nu)(1-\gamma^{-1})\mathcal{A}_k}{4\tilde{C}}\cdot\frac{\Phi_k^2}{\Phi_{k\wedge T_\varepsilon}\mathbf{E}[\Phi_{(k+1)\wedge T_\varepsilon}|\mathcal{F}_{k-1}^{G.F}]}\cdot 1_{\{T_\varepsilon>k\}}\\ &\leq \frac{-p_g p_f(1-\nu)(1-\gamma^{-1})\mathcal{A}_k}{4\tilde{C}}\cdot 1_{\{T_\varepsilon>k\}}\end{aligned}$$

where the last inequality follows from $\mathbf{E}[\Phi_{(k+1)\wedge T_\varepsilon}|\mathcal{F}_{k-1}^{G.F}]\leq\Phi_{k\wedge T_\varepsilon}$. The result follows after noting that $1_{\{T_\varepsilon>k\}}(\Psi_{k+1}-\Psi_k)=\Phi_{k\wedge T_\varepsilon}^{-1}-\Phi_{(k+1)\wedge T_\varepsilon}^{-1}$.

□

Theorem 9.3. *Let the assumptions of Theorem 9.2 hold with constant ν and probabilities $p_f p_g$ as in Theorem 9.2. Then the expected number of iterations that Algorithm 1 takes until $f(X_k)-f^*<\varepsilon$ is bounded as follows*

$$\boldsymbol{E}[T_\varepsilon]\leq\mathcal{O}(1)\cdot\frac{p_g p_f}{2p_g p_f-1}\cdot\frac{(k_g\alpha_{\max}+1)^2(k_g+L+\varepsilon_f)(\nu DL+\frac{(1-\nu)\alpha_{\max}L_f}{L}+(1-\nu)\sqrt{\theta}\delta_{\max})^2}{\varepsilon}+1.$$

The above bound can be further simplified as follows

$$\boldsymbol{E}[T_\varepsilon]\leq\mathcal{O}(1)\cdot\frac{p_g p_f}{2p_g p_f-1}\left(\frac{L^3k_g^3(D^2+L_f^2+\delta_{\max}^2)}{\varepsilon}\right).$$

Lemma 9.1 (Accurate gradients and estimates ⇒ successful iteration)**.** *Suppose g_k is k_g-sufficiently accurate and $\{f_k^0,f_k^s\}$ are ε_f-accurate estimates. If*

$$\alpha_k\leq\frac{1-\theta}{k_g+\frac{L}{2}+2\varepsilon_f}$$

then the trial step x_k+s_k is successful. In particular, this means $f_k^s\leq f_k^0-\theta\alpha_k\parallel g_k\parallel^2$.

Proof. The L-smoothness of f and the k_g-sufficiently accurate gradient immediately yield

$$\begin{aligned}f(x_k+s_k)&\leq f(x_k)-\alpha_k(\nabla f(x_k)-g_k)^Tg_k-\alpha_k\parallel g_k\parallel^2+\tfrac{L\alpha_k 2}{2}\parallel g_k\parallel^2\\ &\leq f(x_k)+k_g\alpha_k 2\parallel g_k\parallel^2-\alpha_k\parallel g_k\parallel^2+\tfrac{L\alpha_k 2}{2}\parallel g_k\parallel^2.\end{aligned}$$

Since the estimates are ε_f-accurate, we obtain

$$\begin{aligned}f_k^s-\varepsilon_f\alpha_k 2\parallel g_k\parallel^2&\leq f(x_k+s_k)-f_k^s+f_k^s\\ &\leq f(x_k)-f_k^0+f_k^0+k_g\alpha_k 2\parallel g_k\parallel^2-\alpha_k\parallel g_k\parallel^2+\tfrac{L\alpha_k 2}{2}\parallel g_k\parallel^2\\ &\leq f_k^0+\varepsilon_f\alpha_k 2\parallel g_k\parallel^2+k_g\alpha_k 2\parallel g_k\parallel^2-\alpha_k\parallel g_k\parallel^2+\tfrac{L\alpha_k 2}{2}\parallel g_k\parallel^2.\end{aligned}$$

The result follows by noting $f_k^s\leq f_k^0-\alpha_k\parallel g_k\parallel^2\left(1-\alpha_k\left(k_g+\frac{L}{2}+2\varepsilon_f\right)\right)$. □