Selection of the Step Size in Gradient Descent and Momentum based Variants

George Michailidis

gmichail@ucla.edu

STAT 102B

Key take home message from Lecture 1.2

Gradient Descent algorithm is a simple yet effective iterative algorithm to solve numerically the gradient equation $\nabla f(x) = 0$

Its behavior and performance (number of iterations required to stop) is regulated by the step size (learning rate)

Selection Strategies for the Step Size η_k

1. Exact Line Search method
2. Backtracking line search method
3. Constant step size
4. Diminishing step size along a fixed sequence

Exact Line Search method for η_k - I

Select η_k that minimizes the function $h : \mathbb{R} \to \mathbb{R}$ defined as:

$$h(\eta) = f\left(x_k - \eta \nabla f(x_k)\right)$$

Hence,

$$\eta_k = \text{argmin}_{\eta > 0} f\left(x_k - \eta \nabla f(x_k)\right) \tag{1}$$

Exact Line Search method for η_k - II

For most functions $f : \mathbb{R}^n \to \mathbb{R}$, solving (1) is harder than solving the original optimization problem $\min_{x \in \mathbb{R}^n} f(x)$

A notable exception is the quadratic function $f : \mathbb{R}^n \to \mathbb{R}$

$$f(x) = \frac{1}{2} x^\top Q x + x^\top b + c \tag{2}$$

with $Q \succ 0$ (positive definite)

Remark:
The SSE(β) function is a quadratic function, with (i) $x \leftarrow \beta$, $Q \leftarrow \frac{1}{n}(X^\top X)$, $b = \frac{1}{2n}(Xy^\top)$ and $c \leftarrow \frac{1}{2n}(y^\top y)$

Exact Line Search method for η_k - III

For the quadratic function (2), some algebra shows that the solution to the optimization problem (1) is given by

$$\eta_k = \frac{[\nabla f(x_k)]^\top [\nabla f(x_k)]}{[\nabla f(x_k)]^\top Q [\nabla f(x_k)]} \tag{3}$$

Backtracking Line Search - I

Consider the function $h : \mathbb{R} \to \mathbb{R}$

$$h(\eta_k) = f\left(x_k - \eta_k \nabla f(x_k)\right)$$

Underlying this method is the following Armijo condition:

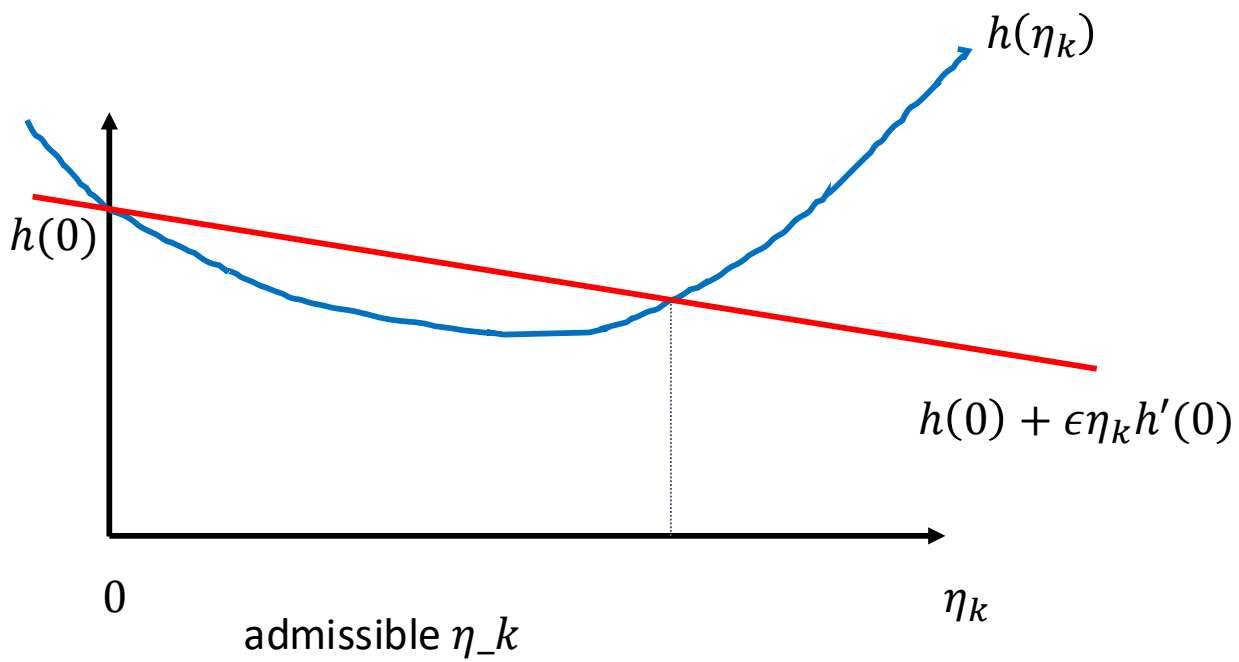
$$h(\eta_k) \leq h(0) + \epsilon \eta_k h'(0), \quad \epsilon \in (0, 1) \tag{4}$$

where $h'(0)$ denotes the derivative of the function $h(\eta_k)$ evaluated at $\eta_k = 0$

It aims to prevent η_k becoming "too large"

Backtracking Line Search - II

Pictorial illustration of the condition



Backtracking Line Search Algorithm

- Select initial value $\eta^0 > 0$ (usually large), $\epsilon \in (0, 1)$ and $\tau \in (0, 1)$ (e.g., $\epsilon = \tau = 0.5$)
- Set $\eta_1 = \eta^0$
- At iteration k, set $\eta_k \leftarrow \eta_{k-1}$:
  1. Check whether the Armijo condition holds for η_k:

$$h(\eta_k) \leq h(0) + \epsilon \eta_k h'(0)$$

  2.
     - If yes, then terminate and keep η_k and proceed with a GD update; i.e., $x_{k+1} = x_k - \eta_k \nabla f(x_k)$
     - Otherwise, set $\eta_k = \tau \eta_k$ and go to step 1

What is the value of $h'(0)$?

Note that in the backtracking line search algorithm, we need to calculate $h'(0)$

Recall that

$$h(\eta_k) = f\left(x_k - \eta_k \nabla f(x_k)\right)$$

An application of the chain rule yields that the derivative of $h(\cdot)$ with respect to η_k is given by:

$$h'(\eta_k) = -[\nabla f(x_k - \eta_k \nabla f(x_k))]^\top \nabla f(x_k)$$

and thus

$$h'(0) = -[\nabla f(x_k)]^\top \nabla f(x_k)$$

Gradient Descent Algorithm with Backtracking Line Search for η_k

1. Select $x_0 \in \mathbb{R}^n$
2. While stopping criterion$>$tolerance do:
   - Use backtracking line search algorithm to select η_k
   - $x_{k+1} = x_k - \eta_k \nabla f(x_k)$
   - Calculate the value of the stopping criterion

Revisit the Example of a Function in $\mathbb{R}^2$

Recall the function $f : \mathbb{R}^2 \to \mathbb{R}$, with

$$f(x) \equiv f(x_1, x_2) = 3x_1^2 + 0.5x_2^2 + 2x_1x_2$$

We have that

$$\nabla f(x) = \begin{bmatrix} 6x_1 + 2x_2 \\ x_2 + 2x_1 \end{bmatrix}$$

Some algebra gives that the global minimum is $x_{\min} = (0, 0)$, since

$$\nabla^2 f(x) = \begin{bmatrix} 6 & 2 \\ 2 & 1 \end{bmatrix}$$

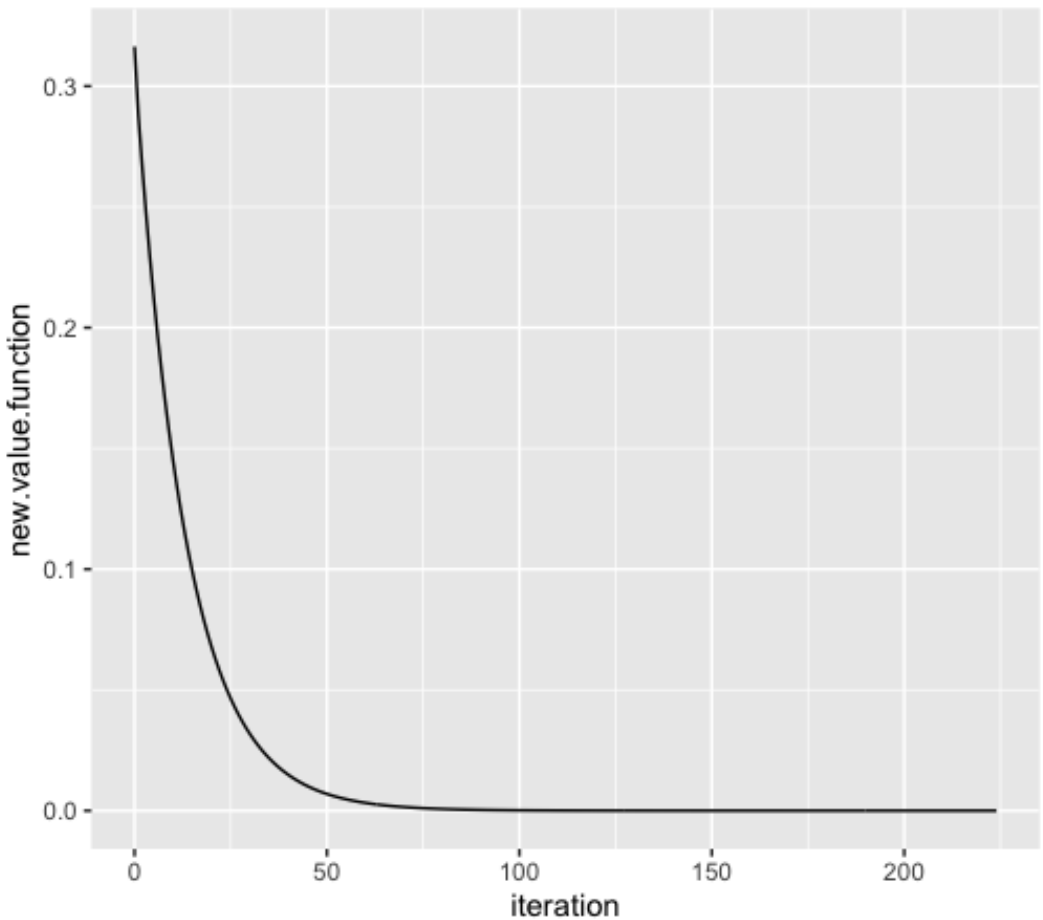
a positive definite matrix (its determinant is positive)

Results for the Example Function with Backtracking Line Search - I

$\eta^0 = 1$, tolerance $= 0.000000001$, $\epsilon = 0.5, \tau = 0.5$

$\hat{x}_{\text{min}} = (9.558577e-05, 2.724941e-04)$, so practically the theoretical solution

After the first iteration the step size becomes 0.125 and does not change in subsequent iterations

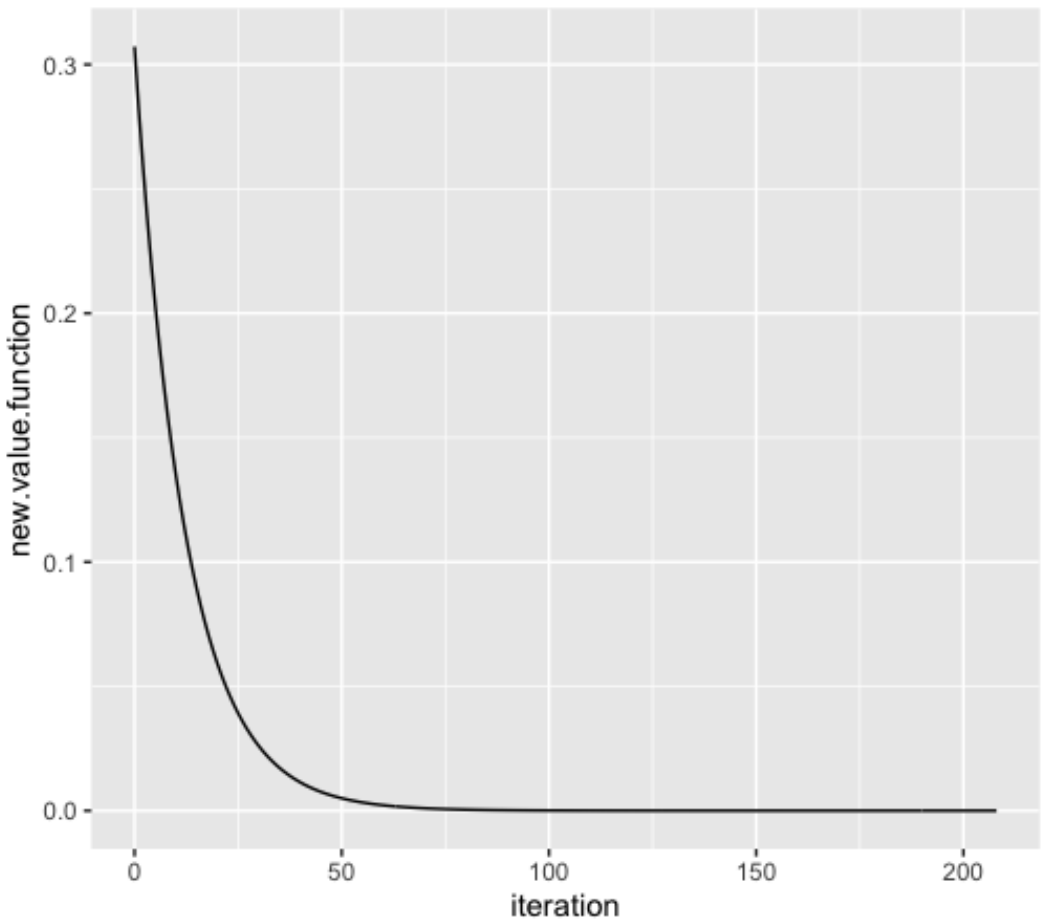


Results for the Example Function with Backtracking Line Search - II

$\eta^0 = 1$, tolerance $= 0.000000001$, $\epsilon = 0.5, \tau = 0.9$

$\hat{x}_{\text{min}} = (-9.098907e-05, 2.593899e-04)$, so practically the theoretical solution

After the first iteration the step size becomes 0.135 and does not change in subsequent iterations



Constant Step Size η

Theoretical analysis of the convergence properties of gradient descent for strictly convex functions[1] (i.e., functions that have a unique global minimum) establishes that

- if

$$\eta \in (0, \frac{1}{L})$$

  then the gradient descent algorithm will converge to the unique global minimum

[1] a brief overview of convex functions will be given later on, when we discuss the order of the number of iterations gradient descent and its variants require to meet the required tolerance criterion

Technical Digression - What is L?

L is known as the Lipschitz constant that quantifies how "smooth" the function's gradient $\nabla f(x)$ is

Technically, we have that

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2, \quad \forall\, x, y \in \mathbb{R}^n$$

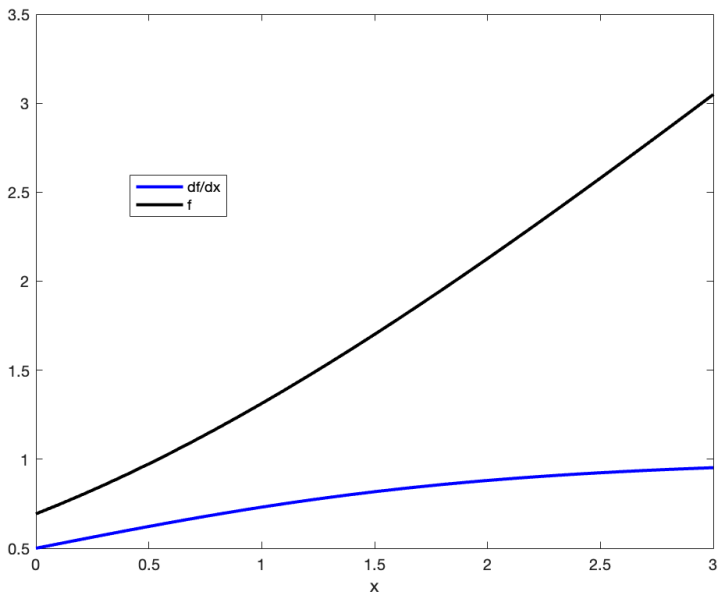
Illustrative Example

Let $f : \mathbb{R} \to \mathbb{R}$ with

$$f(x) = \log(1 + \exp(x))$$

and

$$\frac{df}{dx} = \frac{\exp(x)}{1 + \exp(x)}$$



Then, $L = 1/4$ (not unique though; for example, you can use $L = 1$, but this choice would compromise the range of admissible constant step sizes)

The special case of the quadratic function - I

Let $f : \mathbb{R}^n \to \mathbb{R}$, with

$$f(x) = \frac{1}{2} x^\top Q x + x^\top b + c,$$

with Q being a $n \times n$ matrix and positive definite

Then, its gradient is $\nabla f(x) = Qx + b$ and its Hessian $H = Q$

Since H is positive definite by assumption on the Q, the function $f(\cdot)$ possesses a unique global minimum, since the gradient equation

$$\nabla f(x) = 0 \Longrightarrow Qx + b = 0 \tag{5}$$

corresponds to a system of linear equations where the coefficient matrix Q is full rank and hence has a unique solution

Example of a quadratic function

Let $f : \mathbb{R}^2 \to \mathbb{R}$, with

$$f(x) \equiv f(x_1, x_2) = \frac{3}{2}x_1^2 + x_2^2 + x_1 x_2$$

Let $x = [x_1 \ x_2]^\top$

$$Q = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$$

$b = [0 \ \ 0]^\top$ and $c = 0$;
it can be seen that f can be written in the general form of the quadratic function

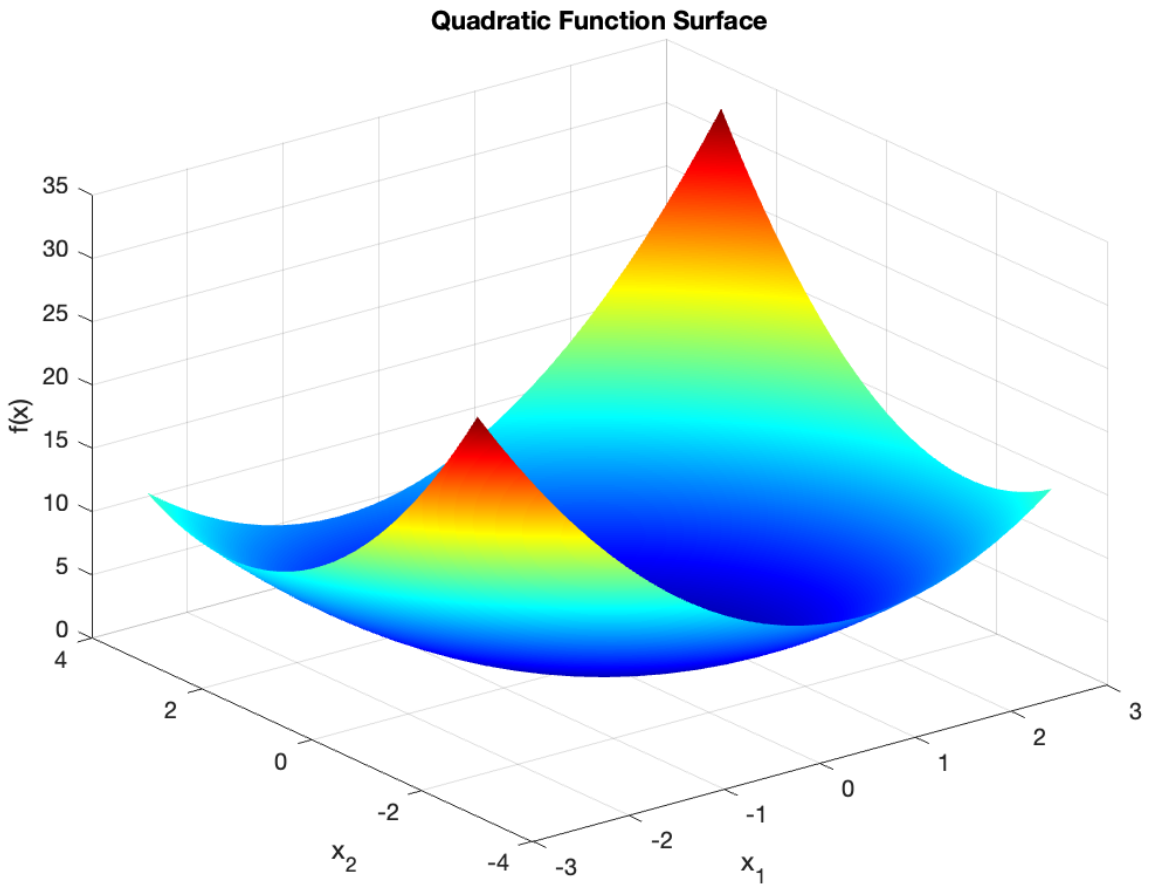


Figure 1: Plot of the function $f(x)$

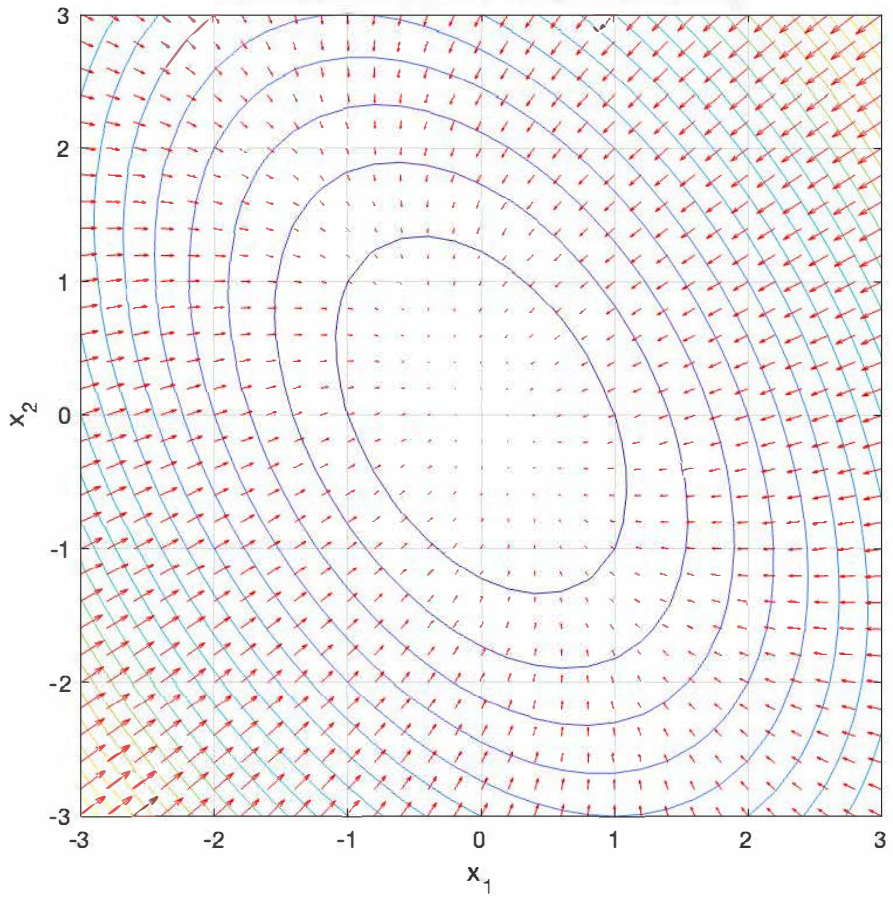


Figure 2: Plot of the gradient directions $-\nabla f(x)$

The special case of the quadratic function - II

It can be shown that $L = \lambda_{\max}(Q)$ for the quadratic function, where $\lambda_{\max}(\cdot)$ denotes the maximum eigenvalue of Q
A brief overview of

However, given the special property of the quadratic function, namely that its Hessian $H = Q$ is constant (and does not depend on x), the range of admissible step sizes is

$$\eta \in (0, \frac{2}{\lambda_{\max}(Q)}) = \frac{2}{L},$$

instead of $1/L$ which is the admissible range for (strictly) convex functions

Back to our example quadratic function - I

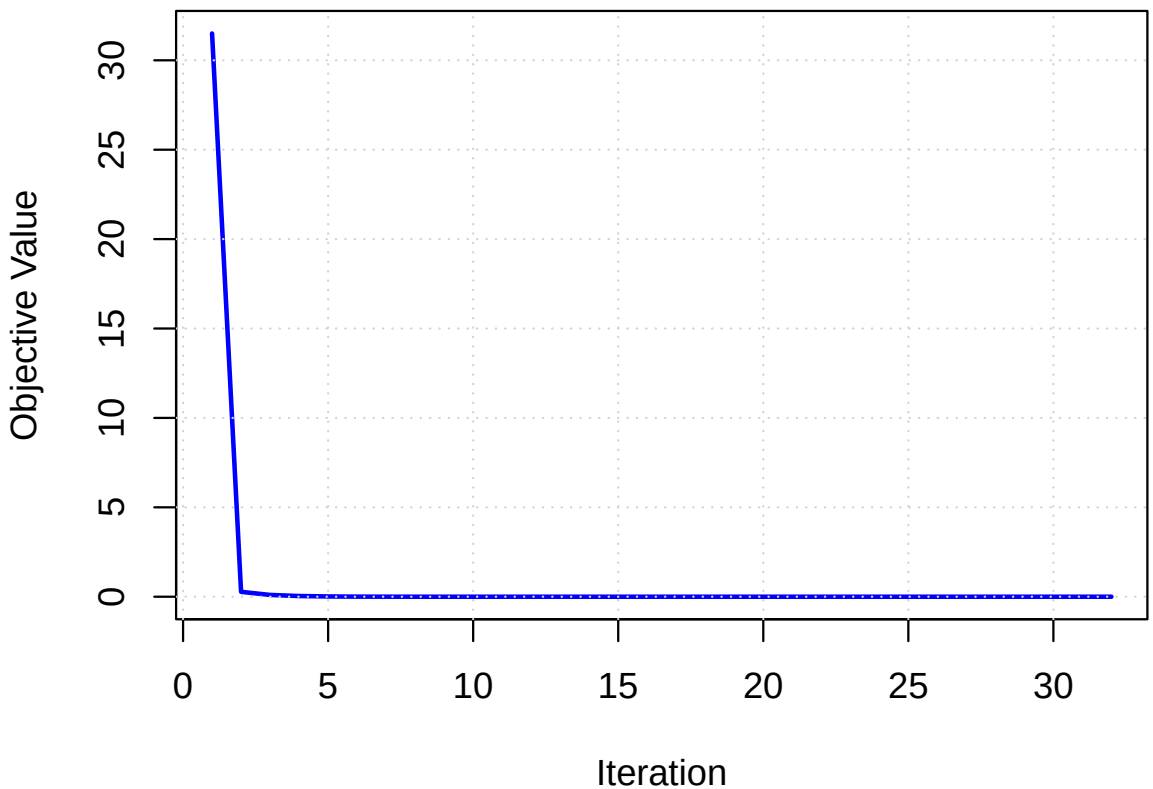
$f(x) = \frac{1}{2} x^\top Q x$, with

$$Q = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$$

Hence, $\lambda_{\max}(Q) \approx 3.618$ and thus $\eta \in (0, 0.5528)$

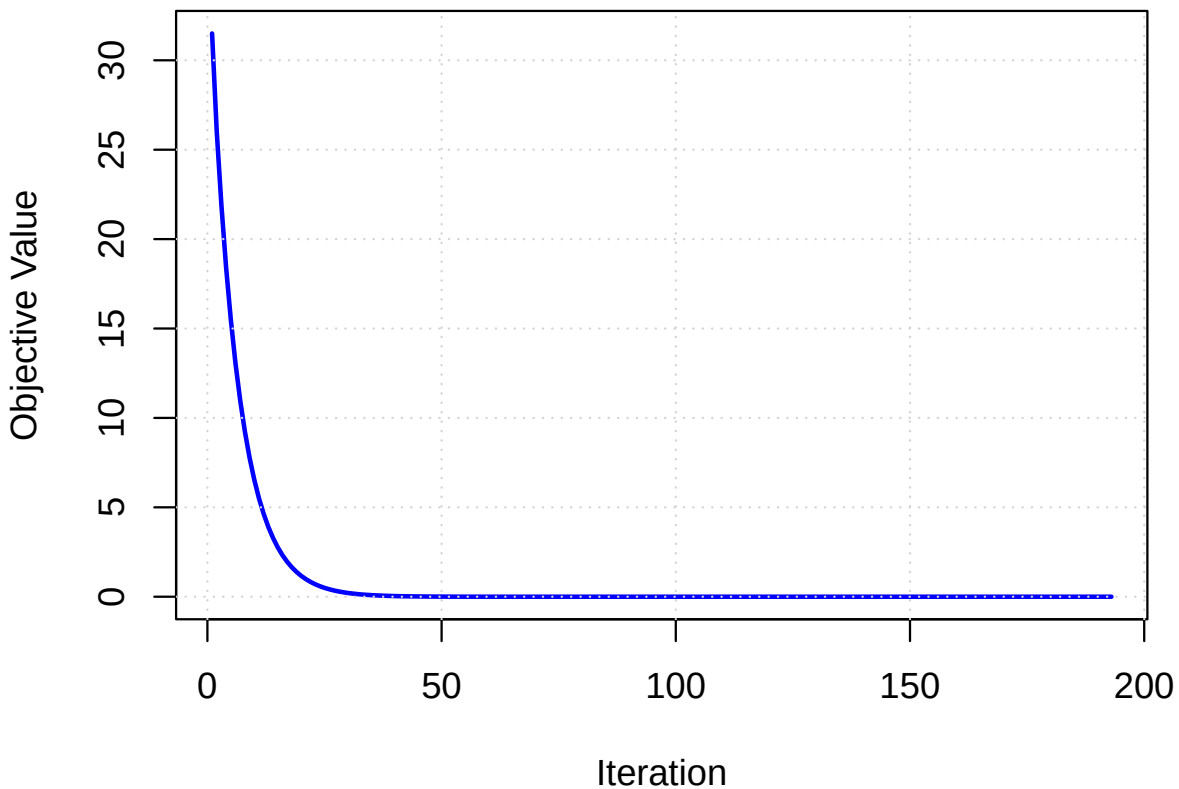
Back to our example function - II

$\eta = \frac{1}{L} = \frac{1}{3.618}$, tolerance $= 10^{-6}$,
$\hat{x}_{\text{min}} = (-2.380805e-06, 3.852224e-06)$, so practically the theoretical solution



Back to our example function - III

$\eta = 0.53$ (close to $\frac{2}{L}$), tolerance $= 10^{-6}$,
$\hat{x}_{\text{min}} = (-2.380805e-06, 3.852224e-06)$, so practically the theoretical solution



Back to our example function - IV

The theoretical result provides an upper bound on the value that a constant step size η can take so that gradient descent is convergent for quadratic functions

It does not necessarily imply that a value very close to the upper bound is better (in terms of the number of iterations required for the same tolerance level for the stopping criterion) than a value quite further away

Diminishing Step Size Along a Fixed Sequence

Key Requirement:

- $\sum_{k=1}^{\infty} \eta_k = \infty$ (ensures progress)

Remark:

- Typical sequences used in practice:
  1. $\eta_k = \frac{\eta^0}{k}$
  2. $\eta_k = \frac{\eta^0}{\sqrt{k}}$
- Descent is not guaranteed at each iteration; only later when η_k becomes small (below $1/L$ for strictly convex functions)
- The gradient descent algorithm may require many iterations
- More useful for Stochastic Gradient Descent (will revisit this choice when we discuss this variant)

Momentum based Variants of Gradient Descent

Motivating Example I

Consider the following quadratic function $f : \mathbb{R}^2 \to \mathbb{R}$,

$$f(x) = \frac{1}{2} x^\top Q x$$

with

$$Q = \begin{bmatrix} 3 & -2 \\ -2 & 3 \end{bmatrix}$$

so that $\lambda_{\max}(Q) = 5$

$\eta = 0.35$, $x_0 = [5\ 10]^\top$, tolerance=10^{-6}

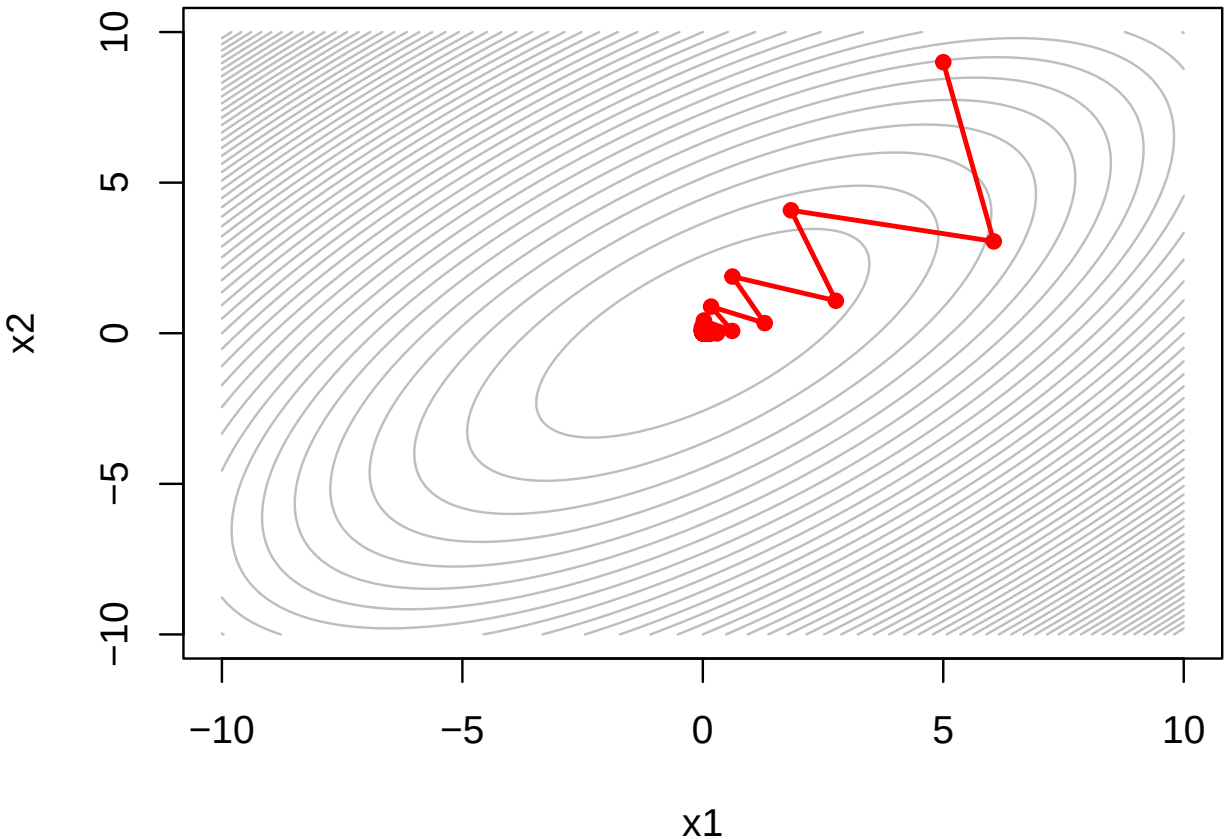


Figure 3: $\hat{x}_{\text{min}}$ =(-1.132936e-07, 1.134909e-07), # iterations = 58

Motivating Example II

Consider the following quadratic function $f : \mathbb{R}^2 \to \mathbb{R}$,

$$f(x) = \frac{1}{2} x^\top Q x$$

with

$$Q = \begin{bmatrix} 2.55 & -2.45 \\ -2.45 & 2.55 \end{bmatrix}$$

$\lambda_{\max}(Q) = 5$

$\eta = 0.35$, $x_0 = [5\ 10]^\top$, tolerance$=10^{-06}$

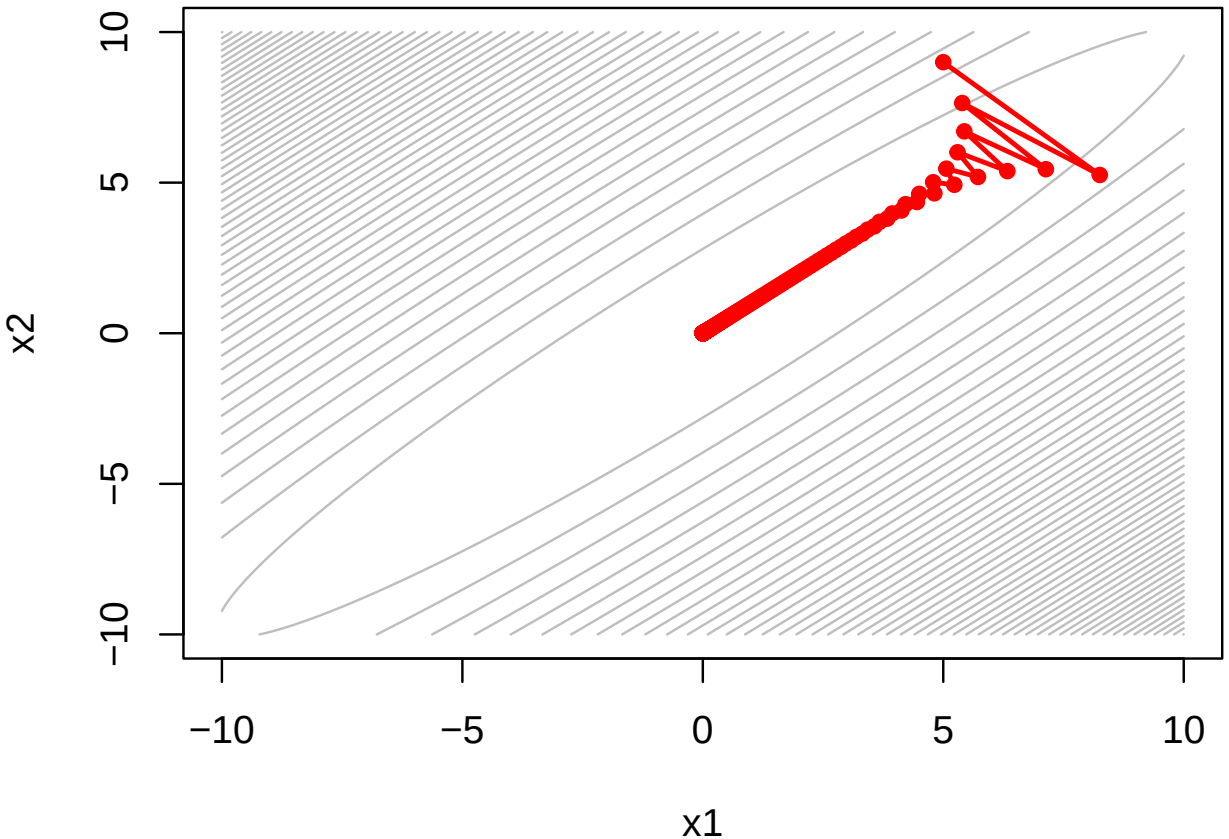


Figure 4: $\hat{x}_{\text{min}}$ =(6.945374e-06, 6.945374e-06), # iterations = 388

Motivating Example III

Consider the following quadratic function $f : \mathbb{R}^2 \to \mathbb{R}$,

$$f(x) = \frac{1}{2} x^\top Q x$$

with

$$Q = \begin{bmatrix} 2.505 & -2.495 \\ -2.495 & 2.505 \end{bmatrix}$$

$\lambda_{\max}(Q) = 5$

$\eta = 0.35$, $x_0 = [5\ 10]^\top$, tolerance$=10^{-6}$

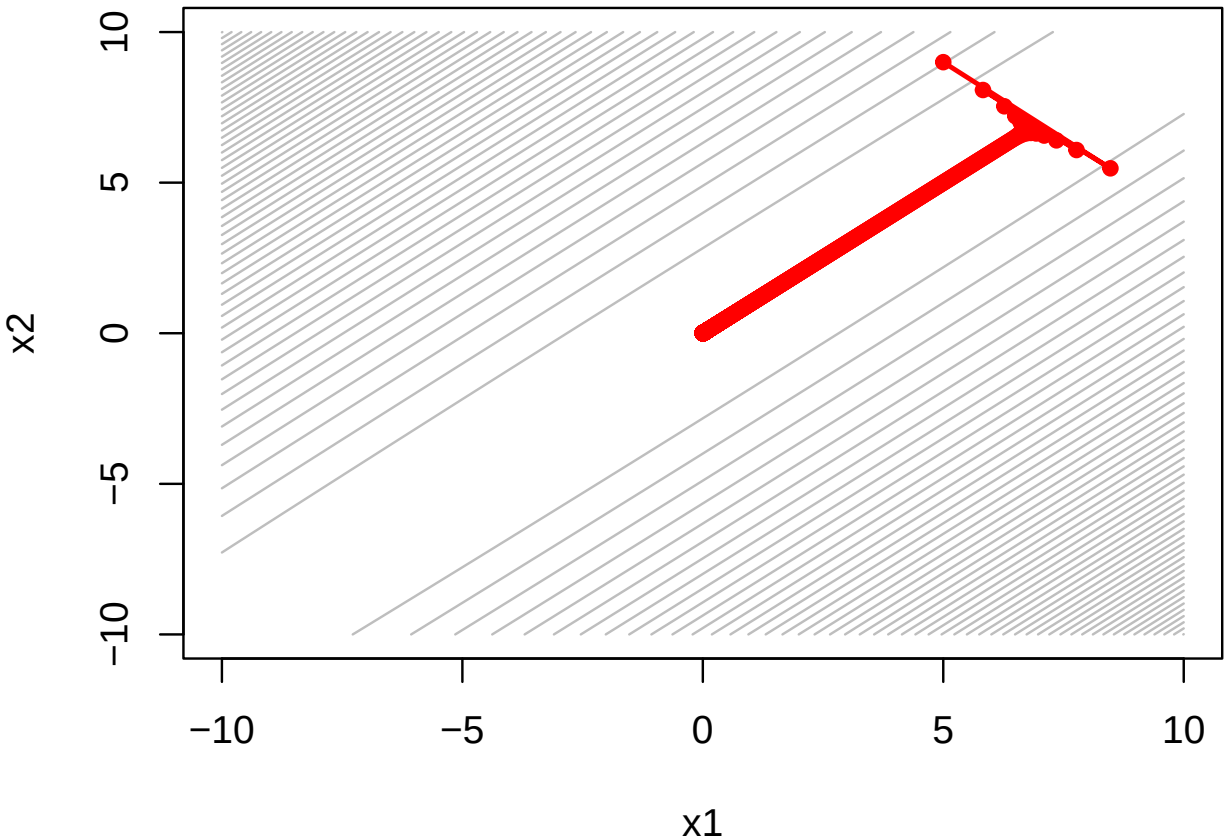


Figure 5: $\hat{x}_{\text{min}}$ =(7.065279e-05, 7.065279e-05), # iterations = 3281

Remarks

The performance of gradient descent deteriorates when the objective function is highly curved in some directions, but flat in others

This is a common phenomenon for the optimization problem for many statistical and machine learning models
e.g., the problem of multicollinearity in linear regression manifests itself with an objective function that resembles that of example III (but in more dimensions)

Momentum methods aim to implicitly "tell the difference" between the curved and flat directions using information that is already available to the algorithm

Polyak's Heavy Ball Method

Recall that the gradient descent update is given by

$$x_{k+1} = x_k - \eta_k \nabla f(x_k)$$

Polyak's heavy ball method uses the following update

$$x_{k+1} = y_k - \eta_k \nabla f(x_k), \quad \text{(gradient step)} \tag{6}$$

$$y_k = x_k + \underbrace{\xi(x_k - x_{k-1})}_{\text{momentum}}, \quad \xi \in (0,1) \quad \text{(momentum step)} \tag{7}$$

- ξ is known as the momentum parameter
- in practice, ξ is selected in the range $(0.5, 0.95)$
- a larger ξ is not necessarily helpful

Intuition behind Polyak's Heavy Ball Method

Suppose that the direction of x_k and x_{k-1} (recall that they are vectors in $\mathbb{R}^n$) does not "change" much; then, their difference will be very small and hence the update will take an "almost full" step along the gradient direction

On the other hand, if the directions of x_k and x_{k-1} are very different, then since we are adding a fraction of the difference to the current update x_k, the update x_{k+1} will be further along than the current one

Obviously, the choice of the parameter ξ matters in the implementation of the method

Revisiting Motivating Example II

$\eta = 0.35, x_0 = [5\ 10]^\top$, tolerance $= 10^{-6}$, $\xi = 0.75$

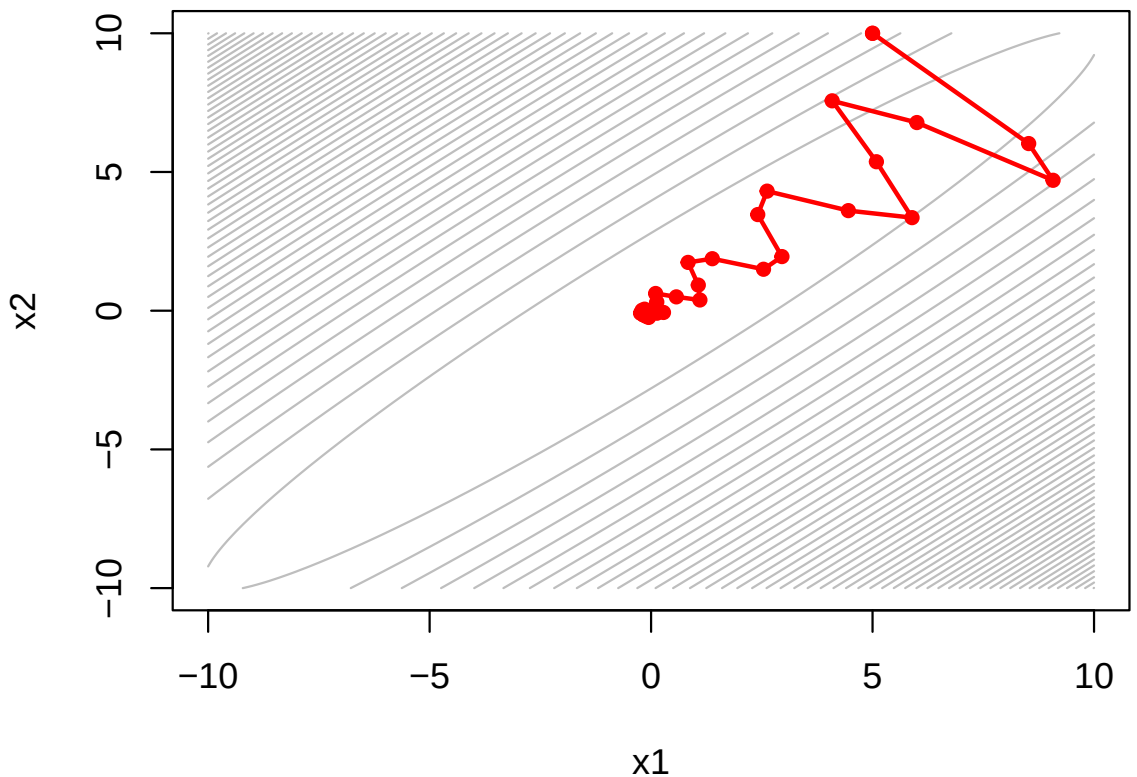


Figure 6: $\hat{x}_{\text{min}} = (1.015242e-06, 1.032939e-06)$, # iterations $= 103$

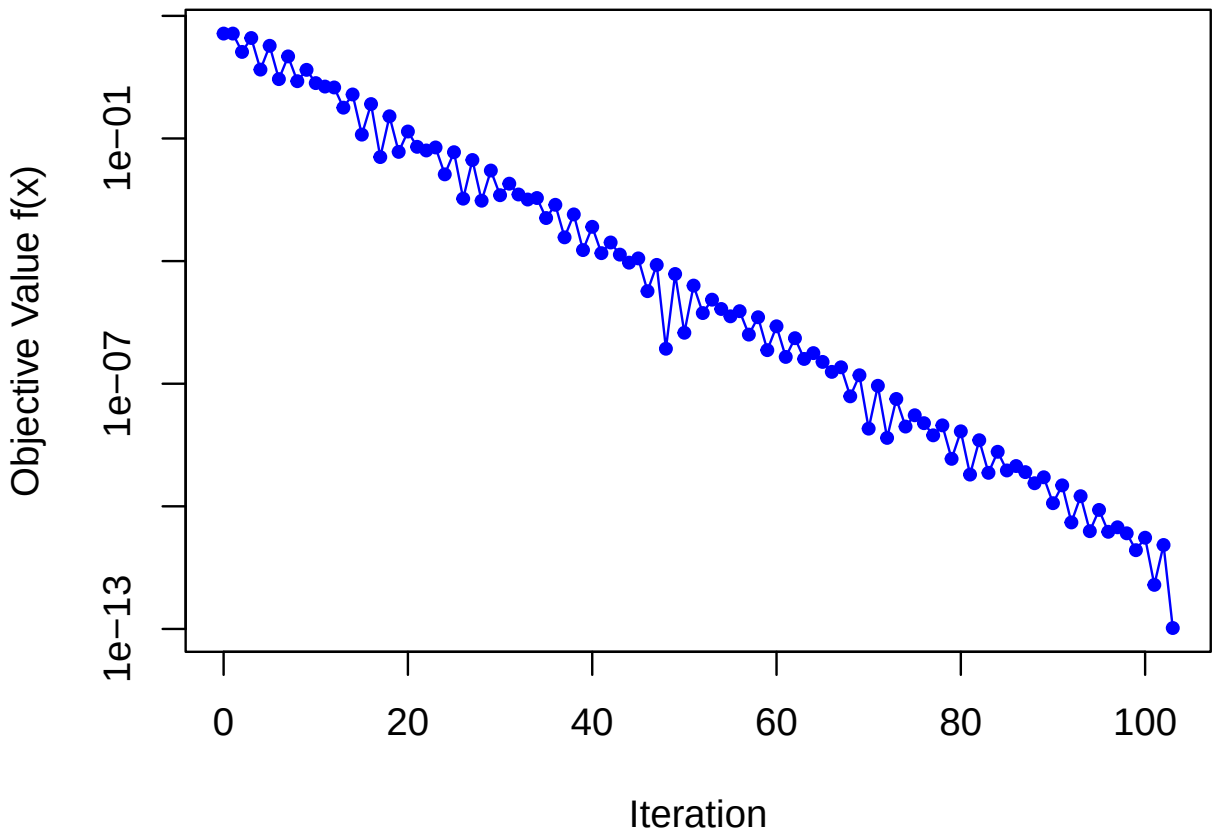


Figure 7: Polyak's method does not utilize a pure descent direction at each iteration! Nevertheless, it is guaranteed to converge for twice differentiable convex functions; best performance for ill-conditioned quadratic functions

Revisiting Motivating Example III

$\eta = 0.35, x_0 = [5\ 10]^\top$, tolerance $= 10^{-6}$, $\xi = 0.75$

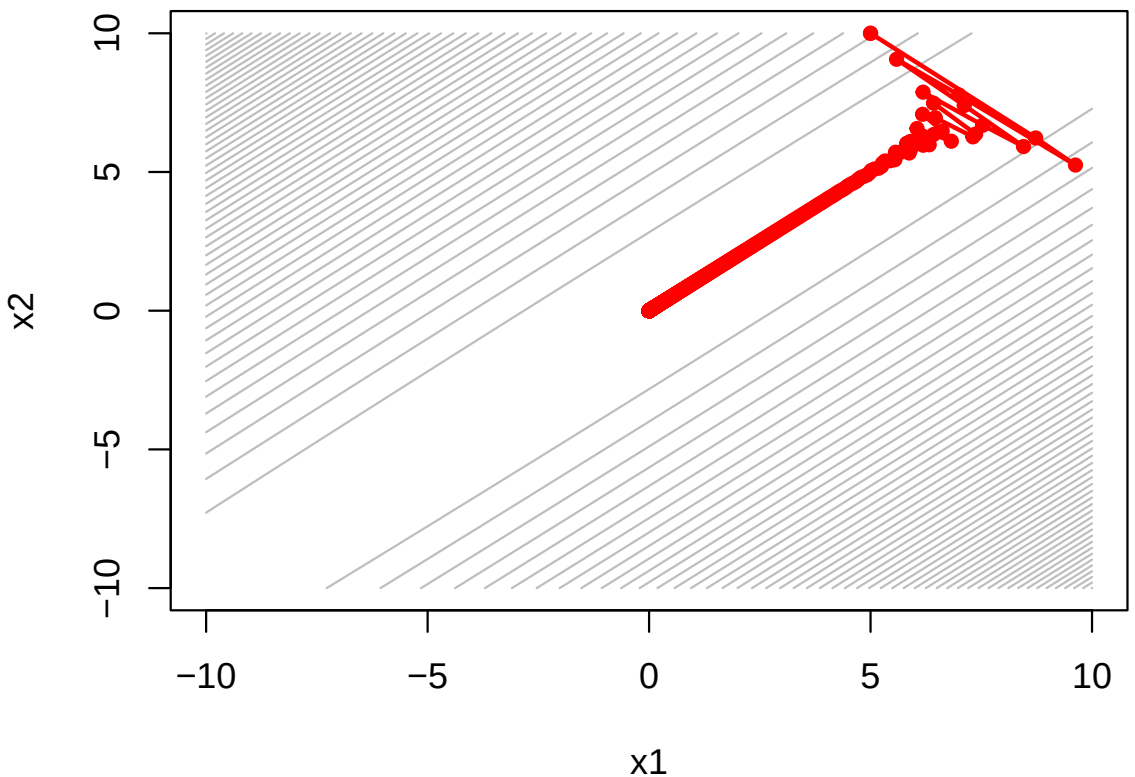


Figure 8: $\hat{x}_{\text{min}} = (7.002553e-05, 7.002553e-05)$, # iterations $= 927$

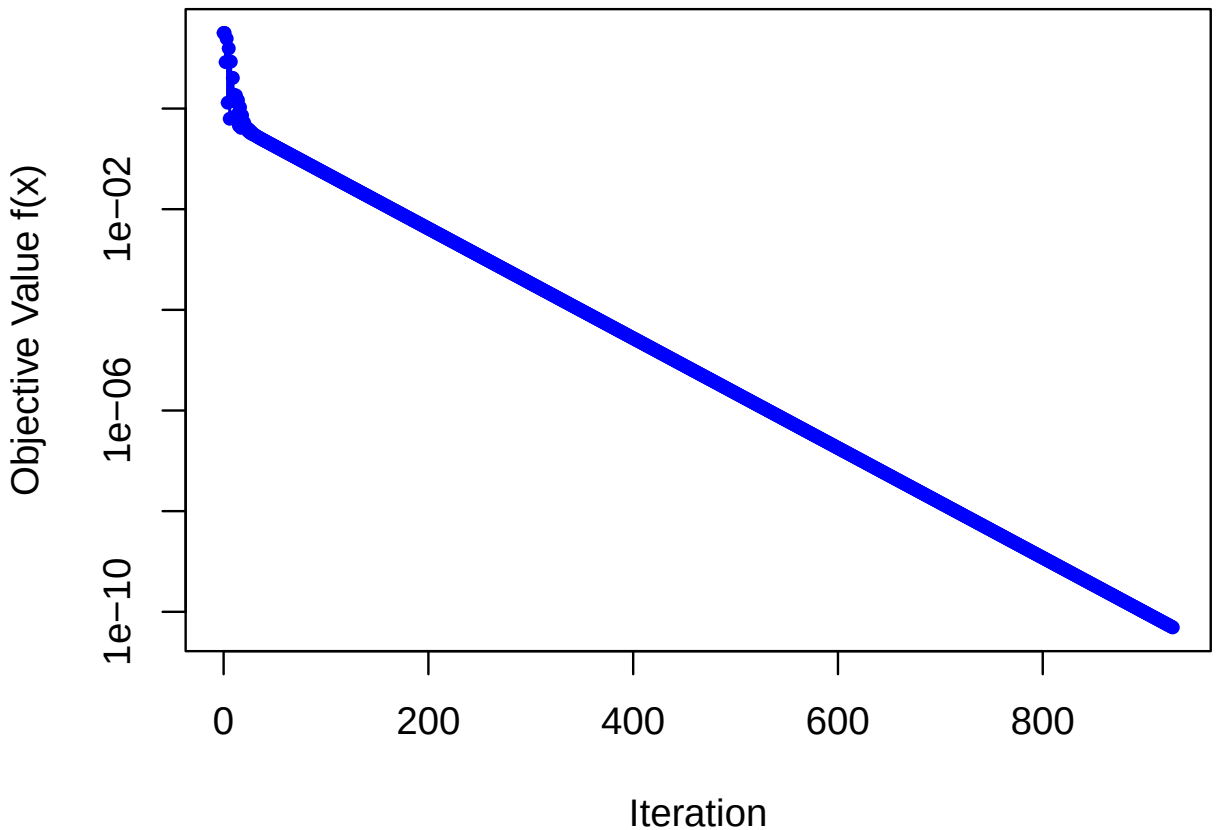


Figure 9: Polyak's method does not utilize a pure descent direction at each iteration!

Nesterov's Momentum Method

It uses the following update

$$x_{k+1} = y_k - \eta_k \nabla f(y_k), \quad (\text{"look-ahead" gradient step}) \tag{8}$$

$$y_k = x_k + \underbrace{\xi_k (x_k - x_{k-1})}_{\text{momentum}}, \quad \xi_k \in (0, 1) \quad (\text{momentum step}) \tag{9}$$

The technical analysis of this algorithm shows that for convex functions (e.g., quadratic, sum-of-square-errors), the optimal choice for $\xi_k = \frac{k-1}{k+2}$, where k is the iteration index

Revisiting Motivating Example II

$\eta = 0.2, x_0 = [5\ 10]^\top$, tolerance $= 10^{-6}$, $\xi = 0.75$

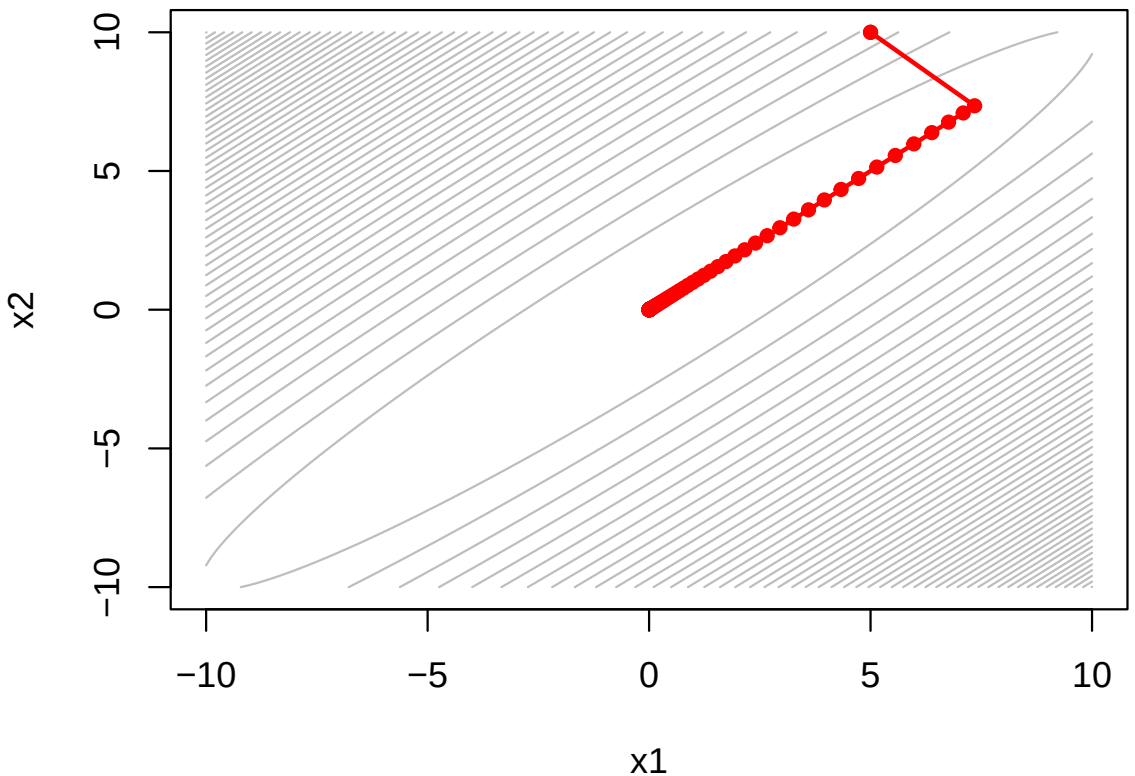


Figure 10: $\hat{x}_{\text{min}} = (6.383492e-06, 6.383492e-06)$, # iterations $= 116$

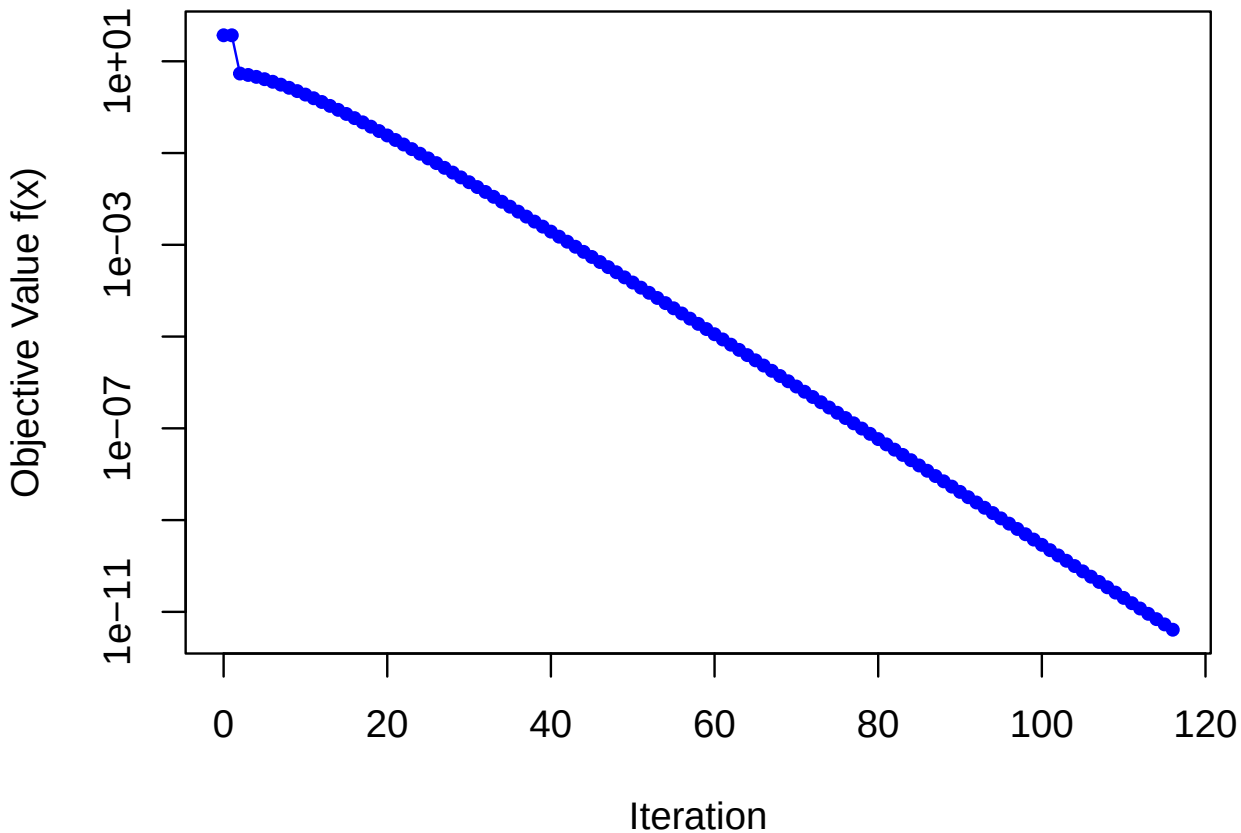


Figure 11: Nesterov's method does not utilize a pure descent direction at each iteration! Nevertheless, it is guaranteed to converge for convex functions

Revisiting Motivating Example III

$\eta = 0.2, x_0 = [5\ 10]^\top$, tolerance $= 10^{-6}$, $\xi = 0.90$

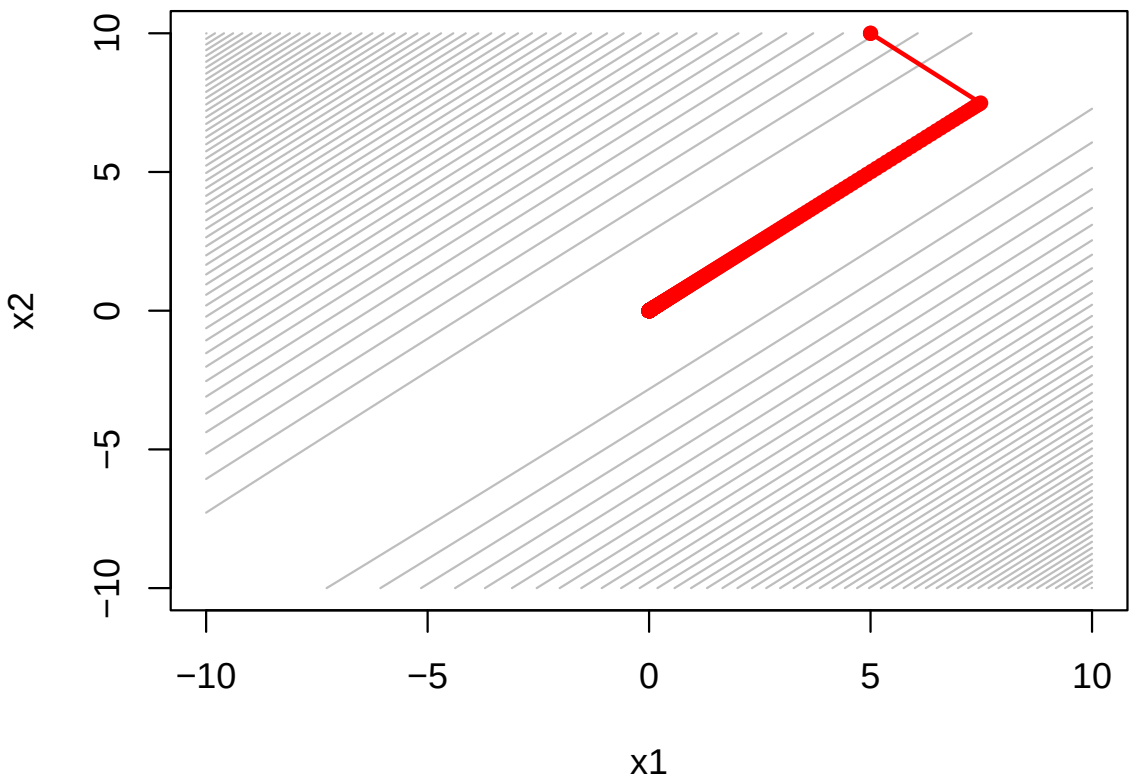


Figure 12: $\hat{x}_{\text{min}} = (6.914624e-05, 6.914624e-05)$, # iterations $= 463$

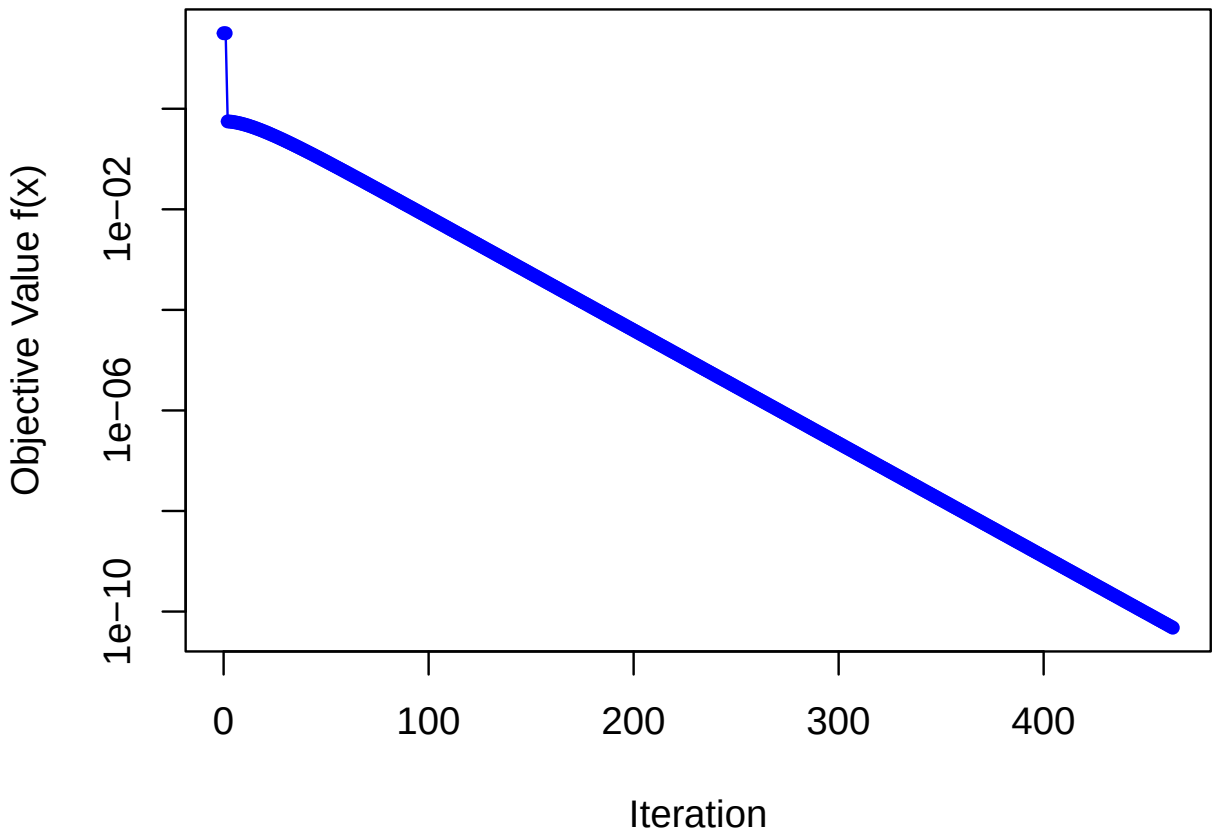


Figure 13: Nesterov's method does not utilize a pure descent direction at each iteration!