MATH3027: Optimization 2022

Week 4: The Gradient Descent Method (I)

Prof. Richard Wilkinson
Please send any comments or mistakes to r.d.wilkinson@nottingham.ac.uk

So far we have focussed on problems where the optima can be explicitly found, i.e., where there is a closed form mathematical expression giving the solution. In practice, few problems can be solved explicitly, and we have to resort to numerical methods.

We begin this week with the **gradient descent** method, which is arguably the most important algorithm in *unconstrained continuous* optimization. We discuss its rationale, and how to tune it. We will also study its convergence properties, and the importance of the condition number of a matrix in understanding the convergence rate of the method.

Descent Directions Methods

Recall that our objective is to find an optimal solution of the problem

$$\min \{f(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\} .$$

We have seen that in some particular cases, such as linear least squares problems, it is possible to obtain a direct solution by solving the normal equations. Unfortunately, this is not the case for an arbitrary nonlinear objective $f(\mathbf{x})$. However, in large problems (i.e. when n is large) even when we can obtain a direct solution, it is sometimes preferable to find the solution using a numerical approach in order to reduce the computational cost.

In this chapter we will seek a solution of this optimization problem using iterative algorithms of the form

$$\mathbf{x}^{k+1} = \mathbf{x}^k + t^k \mathbf{d}^k\,,\ k = 0, 1, \ldots\,,$$

where $\mathbf{x}^k$ is our current guess at the optima, $\mathbf{d}^k \in \mathbb{R}^n$ is a **direction** and $t^k \in \mathbb{R}$ is called the **stepsize**[1]

We will see that a careful selection of $\mathbf{d}^k$ and t^k will generate a sequence $\{\mathbf{x}^k\}_{k=0}^{\infty}$ converging to a stationary point $\mathbf{x}^*$ such that $\nabla f(\mathbf{x}^*) = 0$ (a very good candidate for solving our original problem). A first important concept is the one of descent direction.

Definition (Descent Direction)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function over $\mathbb{R}^n$. A vector $\mathbf{0} \neq \mathbf{d} \in \mathbb{R}^n$ is called a descent direction of f at $\mathbf{x}$ if the directional derivative $f'(\mathbf{x}; \mathbf{d})$ is negative, meaning that*

$$f'(\mathbf{x}; \mathbf{d}) = \nabla f(\mathbf{x})^\top \mathbf{d} < 0\,.$$

The Descent Property of Descent Directions

Lemma. *Let f be a continuously differentiable function over $\mathbb{R}^n$, and let $\mathbf{x} \in \mathbb{R}^n$. Suppose that $\mathbf{d}$ is a descent direction of f at $\mathbf{x}$. Then there exists $\varepsilon > 0$ such that*

$$f(\mathbf{x} + t\mathbf{d}) < f(\mathbf{x})$$

for any $t \in (0, \varepsilon]$.

Proof. Since $f'(\mathbf{x}; \mathbf{d}) < 0$, it follows from the definition of the directional derivative that

$$\lim_{t \to 0^+} \frac{f(\mathbf{x} + t\mathbf{d}) - f(\mathbf{x})}{t} = f'(\mathbf{x}; \mathbf{d}) < 0\,.$$

Therefore, there exists $\varepsilon > 0$ such that

$$\frac{f(\mathbf{x} + t\mathbf{d}) - f(\mathbf{x})}{t} < 0$$

for any $t \in (0, \varepsilon]$, which readily implies the desired result. □

A first version of the descent direction algorithm is presented below.

Algorithm 1: Schematic Descent Direction Method

Initialization: pick $\mathbf{x}^0 \in \mathbb{R}^n$ arbitrarily.
General Step: for any $k = 0, 1, 2, \ldots$ execute the following steps:

1 Pick a descent direction $\mathbf{d}^k$.
2 Find a stepsize t^k satisfying $f\left(\mathbf{x}^k + t^k \mathbf{d}^k\right) < f\left(\mathbf{x}^k\right)$.
3 Set $\mathbf{x}^{k+1} = \mathbf{x}^k + t^k \mathbf{d}^k$.
4 If a stopping criteria is satisfied, then STOP and give $\mathbf{x}^{k+1}$ as the output.

Of course, many details are missing in the above schematic algorithm:

[1] Note that k is a superscript not a power.

- What is the starting point?
- How to choose the descent direction?
- What stepsize should be taken?
- What is the stopping criteria?

The starting point $\mathbf{x}^0$ is usually chosen arbitrarily (or even randomly), but if we have prior information about where the minimum might be, we may choose $\mathbf{x}^0$ accordingly.

A common choice for the stopping rule is to stop when the gradient is sufficiently close to 0, i.e., to pick $\varepsilon > 0$ and to stop when

$$||\nabla f(\mathbf{x}^{k+1})|| \leq \varepsilon.$$

The choice of the stepsize and descent direction requires more thought.

Stepsize Selection Rules

Let's assume for one second that a descent direction, $\mathbf{d}^k$, has been found. How do we choose the stepsize t^k? If we choose t^k to be too large, we may overshoot the minimum and cause the algorithm to be non-decreasing (so that it will never converge upon a stationary point). Conversely, if t^k is chosen to be too small, then we will need to take many steps to achieve convergence, which is inefficient.

There are several options that are available for choosing t^k:

- **Constant stepsize**: $t^k = \bar{t}$ for any k.
- **Exact stepsize**: t^k is a minimizer[2] of f along the ray[3] $\mathbf{x}^k + t\mathbf{d}^k$ $(t \geq 0)$:

$$t^k \in \operatorname*{argmin}_{t \geq 0} f(\mathbf{x}^k + t\mathbf{d}^k)$$

- **Backtracking (or Armijo rule)**: the method requires three parameters: $s > 0, \alpha \in (0,1)$, and $\beta \in (0,1)$. We start with an initial guess at the best stepsize: $t^k = s$. Then, whilst

$$f(\mathbf{x}^k) - f(\mathbf{x}^k + t^k\mathbf{d}^k) < -\alpha t^k \nabla f(\mathbf{x}^k)^\top \mathbf{d}^k$$

  set $t^k := \beta t^k$, iterating until achieving the **Sufficient Decrease Property**

$$f(\mathbf{x}^k) - f(\mathbf{x}^k + t^k\mathbf{d}^k) \geq -\alpha t^k \nabla f(\mathbf{x}^k)^\top \mathbf{d}^k\,.$$

- and many others...

[2] We call the operation of selecting the element that minimizes instead of just computing the minimum value the argmin.

[3] A line with a fixed start point but no end point.

Finding the right t^k is referred in the literature as line search, and is illustrated in Figure 1. Which approach should we choose? The constant stepsize strategy is simple to implement, but it can be unclear how to choose the constant $\bar{t}$. The exact stepsize strategy is optimal, but can be impossible to implement in complex problems and so is rarely used in practice. The backtracking strategy is a compromise between these two approaches - it aims to find a good enough stepsize (but suboptimal stepsize). The method starts with a guess at the stepsize s, and then uses $\beta^i s$ as the stepsize, where $i \in \mathbb{Z}^+$ is the smallest power such that the sufficient decrease condition holds.

But why does the condition make sense? Note that $\nabla f(\mathbf{x}^k)^\top \mathbf{d}_k < 0$ and so the condition looks for a stepsize that ensures the decrease in f is large enough. But can we always find such a step size?

Lemma. *Suppose* $\mathbf{d}$ *is a descent direction, and that* f *is continuously differentiable. Then there exists* $\varepsilon > 0$ *such that*

$$f(\mathbf{x}^k) - f(\mathbf{x}^k + t^k \mathbf{d}^k) \geq -\alpha t^k \nabla f(\mathbf{x}^k)^\top \mathbf{d}^k \quad \forall t \in [0, \varepsilon].$$

This lemma guarantees that there is always a stepsize that satisfies the sufficient decrease property. The proof is left as an exercise.

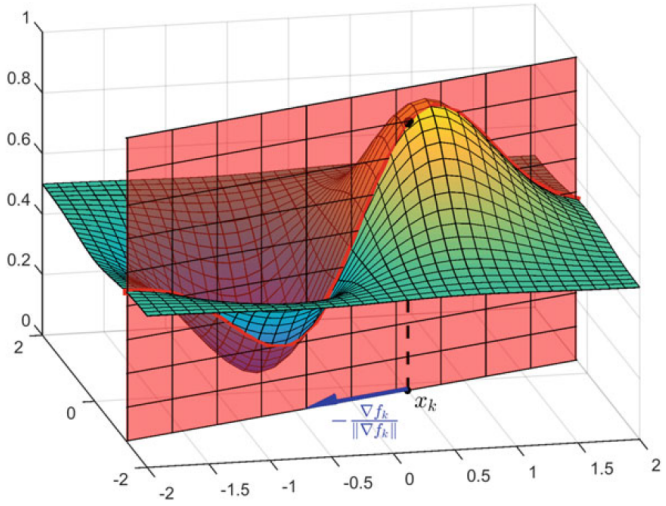

Figure 1: At a given iteration, the current point $\mathbf{x}^k$ and the descent direction $-\frac{\nabla f(\mathbf{x}^k)}{\|\nabla f(\mathbf{x}^k)\|}$ define a plane along which the next iterate $\mathbf{x}^{k+1}$ is sought. The line search procedure defines how far we move in this direction. A constant stepsize will move a fixed distance along the red line, whereas an exact stepsize will look for the minimizer of the surface constrained to the plane.

Gradient descent

How should we choose the descent direction $\mathbf{d}^k$? The most common choice, and in some sense the best choice, is to choose the negative gradient

$$\mathbf{d}^k = -\nabla f\left(\mathbf{x}^k\right).$$

This gives us the famous **gradient descent** method. Note that $\mathbf{d} = \nabla f\left(\mathbf{x}^k\right)$ is a descent direction as long as $\nabla f\left(\mathbf{x}^k\right) \neq 0$ since

$$f'(\mathbf{x}^k; -\nabla f(\mathbf{x}^k)) = -\nabla f(\mathbf{x}^k)^\top \nabla f(\mathbf{x}^k) = -\|\nabla f(\mathbf{x}^k)\|^2 < 0$$

It is an optimal choice for $\mathbf{d}$ as it is the steepest descent direction.

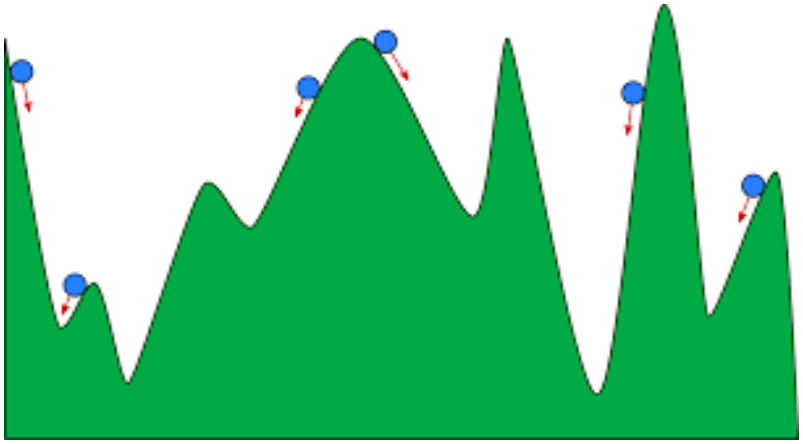

Figure 2: The landscape is $f(\mathbf{x})$ and the balls are ready to move in the direction of $-\nabla f(\mathbf{x})$. What can you say about the points where the balls are expected to end?

Lemma. *Let f be a continuously differentiable function and let $\mathbf{x} \in \mathbb{R}^n$ be a non-stationary point ($\nabla f(\mathbf{x}) \neq 0$). Then an optimal solution of*

$$\min_{\mathbf{d}} \{f'(\mathbf{x}; \mathbf{d}) : \|\mathbf{d}\| = 1\}$$

is $\mathbf{d} = -\nabla f(\mathbf{x})/\|\nabla f(\mathbf{x})\|$.

Proof. Since $f'(\mathbf{x}; \mathbf{d}) = \nabla f(\mathbf{x})^\top \mathbf{d}$, we write the problem as

$$\min_{\mathbf{d} \in \mathbb{R}^n} \left\{\nabla f(\mathbf{x})^\top \mathbf{d} : \|\mathbf{d}\| = 1\right\}$$

By the Cauchy-Schwarz inequality[4] we have

$$\nabla f(\mathbf{x})^\top \mathbf{d} \geq -\|\nabla f(\mathbf{x})\| \cdot \|\mathbf{d}\| = -\|\nabla f(\mathbf{x})\|.$$

[4] $|\mathbf{u}^\top \mathbf{v}| \leq ||\mathbf{u}||.||\mathbf{v}||$, and in particular, $\mathbf{u}^\top \mathbf{v} \geq -||\mathbf{u}||.||\mathbf{v}||$

Thus, $-\|\nabla f(\mathbf{x})\|$ is a lower bound on the optimal value of the directional derivative. On the other hand, using the direction $\mathbf{d} = -\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}$ we obtain that

$$f'\left(\mathbf{x}, -\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}\right) = -\nabla f(\mathbf{x})^\top \left(\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}\right) = -\|\nabla f(\mathbf{x})\|$$

and we thus come to the conclusion that the lower bound $-\|\nabla f(\mathbf{x})\|$ is attained at $\mathbf{d} = -\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|}$, which implies that this is an optimal solution for the descent direction. □

Algorithm 2: The Gradient Method

Initialization: A tolerance parameter $\varepsilon > 0$ and $\mathbf{x}^0 \in \mathbb{R}^n$.

General Step: for any $k = 0, 1, 2, \ldots$ execute the following steps:

1 Pick a stepsize t^k by a line search procedure on the function

$$g(t) = f(\mathbf{x}^k - t\nabla f(\mathbf{x}^k)) .$$

2 Set $\mathbf{x}^{k+1} = \mathbf{x}^k - t^k \nabla f\left(\mathbf{x}^k\right)$.

3 If $\left\|\nabla f\left(\mathbf{x}^{k+1}\right)\right\| \leq \varepsilon$, then STOP and output $\mathbf{x}^{k+1}$.

Example. Try solving

$$\min x_1^2 + 2x_2^2$$

using a gradient descent with exact line search. Use the starting guess $\mathbf{x}^0 = (2, 1)$, and a stopping tolerance of $\varepsilon = 10^{-5}$.The iteration converges in 13 steps, and the convergence history is shown in Figure 3.

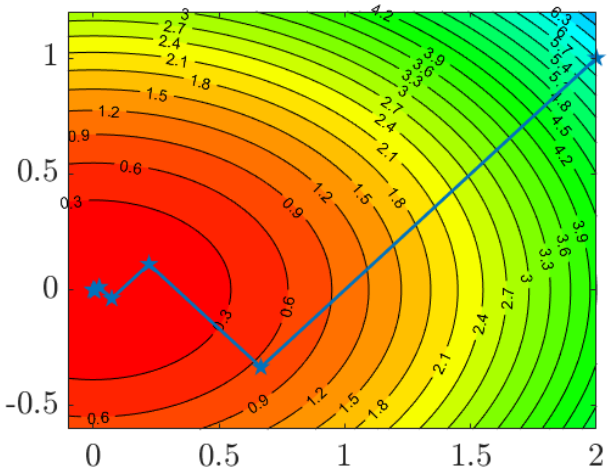

Figure 3: Gradient descent with exact line search for $f(\mathbf{x}) = x_1^2 + 2x_2^2$. After 13 iterations, the method converges to the optimum $(0, 0)$.

The Zig-Zag Effect

Gradient descent can be shown to "zig-zag", as illustrated in Figure 3, meaning that the direction found at the k-iterarion $\mathbf{x}^{k+1} - \mathbf{x}^k$ is orthogonal to the direction found at the $(k+1)$-th iteration $\mathbf{x}^{k+2} - \mathbf{x}^{k+1}$. We now establish this result.

Lemma. *Let $\{\mathbf{x}^k\}_{k>0}$ be the sequence generated by the gradient method with exact line search for solving a problem of minimizing a continuously differentiable function f. Then for any $k = 0, 1, 2, \ldots$*

$$(\mathbf{x}^{k+2} - \mathbf{x}^{k+1})^\top (\mathbf{x}^{k+1} - \mathbf{x}^k) = 0\,.$$

Proof. First, we write $\mathbf{x}^{k+1} - \mathbf{x}^k = -t^k \nabla f\left(\mathbf{x}^k\right)$, and $\mathbf{x}^{k+2} - \mathbf{x}^{k+1} = -t^{k+1} \nabla f\left(\mathbf{x}^{k+1}\right)$. Therefore, we need to prove that $\nabla f\left(\mathbf{x}^k\right)^\top \nabla f\left(\mathbf{x}^{k+1}\right) = 0$. The exact line search is by definition

$$t^k \in \underset{t \geq 0}{\operatorname{argmin}} \left\{ g(t) \equiv f(\mathbf{x}^k - t \nabla f(\mathbf{x}^k)) \right\}$$

Hence,

$$\begin{aligned} g'(t^k) &= 0\,, \\ -\nabla f(\mathbf{x}^k)^\top \nabla f(\mathbf{x}^k - t^k \nabla f(\mathbf{x}^k)) &= 0\,, \\ -\nabla f(\mathbf{x}^k)^\top \nabla f(\mathbf{x}^{k+1}) &= 0\,. \end{aligned}$$

□

Convergence of the Gradient Method

We begin this discussion with a computational example.

Example. Let's solve

$$\min x_1^2 + 2x_2^2$$

using gradient descent with constant stepsize. Set $\mathbf{x}^0 = (2, 1)$, $\varepsilon = 10^{-5}$, and $\bar{t} = 0.1$. In my implementation, the iteration sequence reads

```
[1] "-------- Iteration  1  -----------"
[1] "f(x)= 3.28   norm_grad= 4"
[1] "-------- Iteration  2  -----------"
[1] "f(x)= 1.9   norm_grad= 2.94"
...
[1] "-------- Iteration  57  -----------"
[1] "f(x)= 3.58e-11   norm_grad= 1.2e-05"
[1] "-------- Iteration  58  -----------"
[1] "f(x)= 2.29e-11   norm_grad= 9.58e-06"
```

achieving convergence after 58 iterations. If we increase the stepsize parameter to $\bar{t} = 10$, we observe

```
1] "-------- Iteration  1  -----------"
[1] "f(x)= 4490   norm_grad= 174"
[1] "-------- Iteration  2  -----------"
[1] "f(x)= 5150000   norm_grad= 6250"
...
[1] "-------- Iteration  96  -----------"
[1] "f(x)= 6.1e+305   norm_grad= 2.21e+153"
[1] "-------- Iteration  97  -----------"
[1] "f(x)= Inf   norm_grad= Inf"
```

In this case, the sequence diverges. This leads us to a very important question: how can we choose a constant stepsize so that convergence is guaranteed?

Lipschitz Continuity of the Gradient

When we study optimization methods, we always define the set of functions to which a method can be applied. For example, gradient descent can only be applied to differentiable functions. Perhaps more importantly, we also give the set of functions to which we can guarantee an optimization method will work (by which we usually mean, converge to a local minima). It is unrealistic to expect a method to work for all continuous functions, or even all continuously differentiable functions. Instead, when we study the optimization methods we often also put constraints on the magnitude of the derivatives. These constraints are presented in the form of a **Lipschitz condition**.

Definition (Functions with Lipschitz gradient $C_L^{k,p}(\mathbb{R}^n)$)**.** *Let $C_L^{k,p}(\mathbb{R}^n)$ be the class of functions such that*

- *any $f \in C_L^{k,p}(\mathbb{R}^n)$ is k times continuously differentiable on $\mathbb{R}^n$.*
- *The p^{th} derivative of $f \in C_L^{k,p}(\mathbb{R}^n)$ is Lipschitz continuous on $\mathbb{R}^n$ with constant L:*

$$\text{for all } x, y \in \mathbb{R}^n \qquad \|f^{(p)}(\mathbf{x}) - f^{(p)}(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|.$$

  *L is called the **Lipschitz constant**.*

Some remarks:

- When the constant is not relevant, we sometimes just write $C^{k,p}$.
- We must always have $p \leq k$.
- If $f \in C_L^{k,p}$ then $f \in C_{\tilde{L}}^{k,p}$ for all $\tilde{L} \geq L$, i.e., we can replace Lipschitz constant L with any larger constant $\tilde{L}$.

- If $q \geq k$ then $C_L^{q,p} \subset C_L^{k,p}$.

To understand the convergence of the gradient descent method, we will focus on the functions in $C_L^{1,1}$, that is continuously differentiable functions f such that for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|,$$

These are often called *L-smooth* functions or described as having *Lipschitz continuous gradient*

Examples of $C^{1,1}$ functions:

- **Linear functions** - Given a $\in \mathbb{R}^n$, the function $f(\mathbf{x}) = a^\top x$ is in $C_0^{1,1}$.
- **Quadratic functions** - Let $\mathbf{A}$ be a symmetric $n \times n$ matrix, $\mathbf{b} \in \mathbb{R}^n$ and $c \in \mathbb{R}$. Then the function $f(\mathbf{x}) = \mathbf{x}^\top \mathbf{A}\mathbf{x} + 2\mathbf{b}^\top \mathbf{x} + c$ is a $C^{1,1}$ function. The smallest Lipschitz constant of ∇f is $2\|\mathbf{A}\|_2$.

Directly checking whether a function is in $C^{1,1}$ can be difficult. Thankfully, there is an equivalent condition that is easier to check when the function is twice differentiable:

Theorem (Equivalence to Boundedness of the Hessian)**.** *Function f is a member of $C_L^{2,1}(\mathbb{R}^n) \subset C_L^{1,1}(\mathbb{R}^n)$ if and only if*

$$\left\|\nabla^2 f(\mathbf{x})\right\| \leq L \quad \textit{for all} \quad \mathbf{x} \in \mathbb{R}^n.$$

Proof. First we'll prove the if statement (i.e. $\left\|\nabla^2 f(\mathbf{x})\right\| \leq L \Rightarrow f \in C_L^{2,1}$). Suppose that $\left\|\nabla^2 f(\mathbf{x})\right\| \leq L$ for any $\mathbf{x} \in \mathbb{R}^n$. Then by the fundamental theorem of calculus we have for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$

$$\begin{aligned}\nabla f(\mathbf{y}) &= \nabla f(\mathbf{x}) + \int_0^1 \nabla^2 f(\mathbf{x} + t(\mathbf{y} - \mathbf{x}))(\mathbf{y} - \mathbf{x})dt \\ &= \nabla f(\mathbf{x}) + \left(\int_0^1 \nabla^2 f(\mathbf{x} + t(\mathbf{y} - \mathbf{x}))dt\right)(\mathbf{y} - \mathbf{x})\end{aligned}.$$

To make the first line clearer, consider $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^n$. Then

$$\frac{d}{dt}\mathbf{g}(t\mathbf{r}) = \nabla \mathbf{g}(t\mathbf{r})\mathbf{r}$$

by the chain rule. Then, the fundamental theorem of calculus gives

$$\int_0^1 \nabla \mathbf{g}(t\mathbf{r})\mathbf{r}dt = \int_0^1 \frac{d}{dt}\mathbf{g}(t\mathbf{r})dt = \mathbf{g}(\mathbf{r}) - \mathbf{g}(\mathbf{0})$$

and if we let $\mathbf{g}(t\mathbf{r}) = \nabla f(\mathbf{x} + t\mathbf{r})$ where $\mathbf{r} = \mathbf{y} - \mathbf{x}$, the result follows.

Thus,

$$\begin{aligned}\|\nabla f(\mathbf{y}) - \nabla f(\mathbf{x})\| &= \left\| \left(\int_0^1 \nabla^2 f(\mathbf{x} + t(\mathbf{y} - \mathbf{x}))dt \right) (\mathbf{y} - \mathbf{x}) \right\| \\ &\leq \left\| \int_0^1 \nabla^2 f(\mathbf{x} + t(\mathbf{y} - \mathbf{x}))dt \right\| \|\mathbf{y} - \mathbf{x}\| \text{ by the definition of a matrix norm} \\ &\leq \left(\int_0^1 \left\| \nabla^2 f(\mathbf{x} + t(\mathbf{y} - \mathbf{x})) \right\| dt \right) \|\mathbf{y} - \mathbf{x}\| \\ &\leq L\|\mathbf{y} - \mathbf{x}\|\end{aligned}$$

establishing the desired result $f \in C_L^{1,1}$.

Now we prove the only if part. Suppose now that $f \in C_L^{1,1}$. Then by the fundamental theorem of calculus for any $\mathbf{d} \in \mathbb{R}^n$ and $\alpha > 0$ we have

$$\nabla f(\mathbf{x} + \alpha\mathbf{d}) - \nabla f(\mathbf{x}) = \int_0^a \nabla^2 f(\mathbf{x} + t\mathbf{d})\mathbf{d}dt$$

Thus,

$$\left\| \left(\int_0^\alpha \nabla^2 f(\mathbf{x} + t\mathbf{d})dt \right) \mathbf{d} \right\| = \|\nabla f(\mathbf{x} + \alpha\mathbf{d}) - \nabla f(\mathbf{x})\| \leq \alpha L\|\mathbf{d}\|$$

Dividing by α and taking the limit $\alpha \longrightarrow 0^+$, we obtain

$$\left\| \nabla^2 f(\mathbf{x})\mathbf{d} \right\| \leq L\|\mathbf{d}\|$$

implying that $\left\| \nabla^2 f(\mathbf{x}) \right\| \leq L$. □

Main Convergence Result

We now state an important result connecting functions in $C_L^{1,1}(\mathbb{R}^n)$ and gradient descent.

Lemma (Sufficient decrease of the gradient method). *Let $f \in C_L^{1,1}(\mathbb{R}^n)$. Let $\{\mathbf{x}^k\}_{k\geq 0}$ be the sequence generated by the gradient method for solving*

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x})$$

with one of the following stepsize strategies:

- *constant stepsize $\bar{t} \in \left(0, \frac{2}{L}\right)$,*
- *exact line search,*
- *backtracking procedure with parameters $s > 0, \alpha \in (0, 1)$, and $\beta \in (0, 1)$.*

Then

$$f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k) - M \left\| \nabla f(\mathbf{x}^k) \right\|^2$$

where

$$M = \begin{cases} \bar{t}\left(1 - \frac{\bar{t}L}{2}\right) & \text{constant stepsize,} \\ \frac{1}{2L} & \text{exact line search,} \\ \alpha \min\left\{s, \frac{2(1-\alpha)\beta}{L}\right\} & \text{backtracking.} \end{cases}$$

So each step of gradient descent decreases the objective function by at least $M \left\| \nabla f(\mathbf{x}^k) \right\|^2$. We will now show the convergence of the norms of the gradients $\left\| \nabla f\left(\mathbf{x}^k\right) \right\|$ to zero.

Theorem (Convergence of the Gradient Method)**.** *If*

- $f \in C_L^{1,1}\left(R^n\right)$
- f *is bounded below*[5] *over* $\mathbb{R}^n$

then:

1. *for any* k, $f\left(\mathbf{x}^{k+1}\right) < f\left(\mathbf{x}^k\right)$ *unless* $\nabla f\left(\mathbf{x}^k\right) = 0$.
2. $\nabla f\left(\mathbf{x}^k\right) \to 0$ *as* $k \to \infty$

where $\mathbf{x}^k$ *is the sequence generated by gradient descent using the stepsize strategies from the previous lemma.*

Proof. (1) By the sufficient decrease of the gradient method (previous lemma) we have that

$$f(\mathbf{x}^k) - f(\mathbf{x}^{k+1}) \geq M \left\| \nabla f(\mathbf{x}^k) \right\|^2 \geq 0$$

for some constant $M > 0$, and hence the equality $f\left(\mathbf{x}^k\right) = f\left(\mathbf{x}^{k+1}\right)$ can hold only when $\nabla f\left(\mathbf{x}^k\right) = 0$.

(2) Since the sequence $\left\{f\left(\mathbf{x}^k\right)\right\}_{k \geq 0}$ is nonincreasing and bounded below, it converges. Thus, in particular $f\left(\mathbf{x}^k\right) - f\left(\mathbf{x}^{k+1}\right) \to 0$ as $k \to \infty$, which combined with the sufficient decrease of the gradient method implies that $\left\| \nabla f\left(\mathbf{x}^k\right) \right\| \to 0$ as $k \to \infty$. □

Note that this theorem does not imply convergence to a global minimum, or even a local minimum, but merely convergence to a stationary point. This is typical for all first-order unconstrained optimization methods - it is impossible to guarantee converge to a local minimum. We require rather strict assumptions on the objective function to do better than this.

[5] That is, there exists $m \in \mathbb{R}$ such that $f(\mathbf{x}) > m$ for all $\mathbf{x} \in \mathbb{R}^n$

Example. In the Exercises, you will be asked to solve

$$\min x_1^2 + 2x_2^2$$

using gradient descent with backtracking and parameters $\mathbf{x}^0 = (2, 1)$, $s = 2$, $\alpha = 0.25$, $\beta = 0.5$, and $\varepsilon = 10^{-5}$. If you code this yourself, you should see that it converges in only two iterations! See Figure 4. Note the exact stepsize approach tends to find stationary points quicker than backtracking, but in this case the backtracking strategy got 'lucky' - it over shot the exact stepsize in the first iteration, and ended in a position which meant the second iteration took the algorithm directly to the minimum.

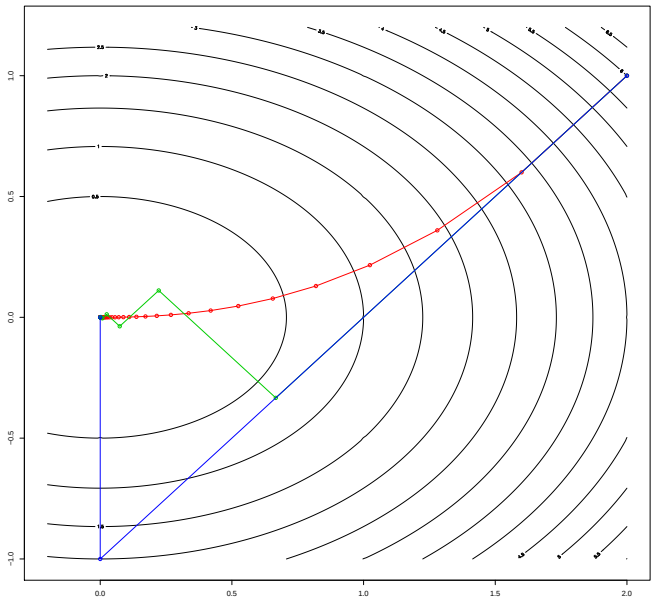

Figure 4: Gradient descent paths taken by the three different stepsize strategies: constant (red), exact (green), backtracking (blue). Note that the constant stepsize method takes a constant multiple of the gradient, which does not result in a constant step in $\mathbf{x}$.

However, if you try to solve

$$\min 0.01x_1^2 + x_2^2$$

with the same method and parameters, it will take about 200 iterations to reach the optimum $(0, 0)$. Can we detect key properties of the objective function that imply fast/slow convergence?

We will see that in the quadratic case, a fundamental characterization is given by the **condition number** of the associated quadratic form.

The Condition Number

We recall the definition of condition number.

Definition (Condition Number). *Let* $\mathbf{A}$ *be an* $n \times n$ *positive definite matrix. Then the condition number of* $\mathbf{A}$ *is defined by*

$$\kappa(\mathbf{A}) = \frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})} \geq 1$$

where $\lambda_{\max}(\mathbf{A})$ *and* $\lambda_{\min}(\mathbf{A})$ *are the largest and smallest eigenvalues, respectively.*

Matrices (or quadratic functions) with large condition number are called *ill-conditioned*. Matrices with small condition number are called *well-conditioned*. Among other things, the condition number gives an idea of the error amplification when working with the matrix $\mathbf{A}$. For an ill-conditioned matrix, small perturbations in the matrix entries can lead to large errors when solving a linear system, or computing $\mathbf{A}^{-1}$.

We continue with a technical lemma.

Lemma (Kantorovich Inequality). *Let* $\mathbf{A}$ *be a positive definite* $n \times n$ *matrix. Then for any* $0 \neq \mathbf{x} \in \mathbb{R}^n$ *the inequality*

$$\frac{\mathbf{x}^\top \mathbf{x}}{(\mathbf{x}^\top \mathbf{A}\mathbf{x})\,(\mathbf{x}^\top \mathbf{A}^{-1}\mathbf{x})} \geq \frac{4\lambda_{\max}(\mathbf{A})\lambda_{\min}(\mathbf{A})}{(\lambda_{\max}(\mathbf{A}) + \lambda_{\min}(\mathbf{A}))^2}$$

holds.

We are now in position to state a precise result regarding the minimization of quadratic functions via gradient descent and its rate of convergence based on the condition number of the matrix $\mathbf{A}$.

Theorem (Gradient Method for Minimizing $\mathbf{x}^\top \mathbf{A}\mathbf{x}$). *Let* $\{\mathbf{x}^k\}_{k\geq 0}$ *be the sequence generated by the gradient method with exact line search for solving the problem*

$$\min_{\mathbf{x}\in\mathbb{R}^n} \mathbf{x}^\top \mathbf{A}\mathbf{x} \quad (\mathbf{A} \succ 0)\,,$$

Then for any $k = 0, 1, \ldots$

$$f(\mathbf{x}^{k+1}) \leq \left(\frac{M-m}{M+m}\right)^2 f(\mathbf{x}^k)\,,$$

where $M = \lambda_{\max}(\mathbf{A})$, *and* $m = \lambda_{\min}(\mathbf{A})$.

This implies

$$0 = f(\mathbf{x}^*) \leq f(\mathbf{x}^k) \leq c^k f(\mathbf{x}^0) \quad \text{where} \quad c = \left(\frac{M-m}{M+m}\right)^2 = \left(\frac{\kappa(A)-1}{\kappa(A)+1}\right)^2,$$

so we say that the sequence of function values converges at a linear rate to the optimal value. Note that the speed of convergence depends upon c. If the condition number, $\kappa(\mathbf{A})$, is small (close to 1), then c will be small and convergence will be fast. If $\kappa(A)$ is large, i.e., $\mathbf{A}$ is ill-conditioned, then $c \approx 1$ and convergence may be very slow.

Proof. Gradient descent with exact line search for a quadratic function is

$$\mathbf{x}^{k+1} = \mathbf{x}^k - t^k \mathbf{d}^k, \text{ where } t^k = \frac{(\mathbf{d}^k)^\top \mathbf{d}^k}{2(\mathbf{d}^k)^\top \mathbf{A}\mathbf{d}^k},$$

and $\mathbf{d}^k = 2\mathbf{A}\mathbf{x}^k$ (see the exercise above).

Plugging in the expression for t^k gives

$$\begin{aligned} f(\mathbf{x}^{k+1}) = (\mathbf{x}^{k+1})^\top \mathbf{A}\mathbf{x}^{k+1} &= (\mathbf{x}^k)^\top \mathbf{A}x^k - \frac{1}{4}\frac{\left((\mathbf{d}^k)^\top \mathbf{d}^k\right)^2}{(\mathbf{d}^k)^\top \mathbf{A}\mathbf{d}^k} \\ &= (\mathbf{x}^k)^\top \mathbf{A}\mathbf{x}^k \left(1 - \frac{1}{4}\frac{\left((\mathbf{d}^k)^\top \mathbf{d}^k\right)^2}{\left((\mathbf{d}^k)^\top \mathbf{A}\mathbf{d}^k\right)\left((\mathbf{x}^k)^\top \mathbf{A}\mathbf{A}^{-1}\mathbf{A}\mathbf{x}^k\right)}\right) \\ &= \left(1 - \frac{\left((\mathbf{d}^k)^\top \mathbf{d}^k\right)^2}{\left((\mathbf{d}^k)^\top \mathbf{A}\mathbf{d}^k\right)\left((\mathbf{d}^k)^\top \mathbf{A}^{-1}\mathbf{d}^k\right)}\right) f(\mathbf{x}^k)\,. \end{aligned}$$

Finally, using Kantorovich's Inequality:

$$f(\mathbf{x}^{k+1}) \leq \left(1 - \frac{4Mm}{(M+m)^2}\right) f(\mathbf{x}^k) = \left(\frac{M-m}{M+m}\right)^2 f(\mathbf{x}^k) = \left(\frac{\kappa(\mathbf{A})-1}{\kappa(\mathbf{A})+1}\right)^2 f(\mathbf{x}^k)$$

□

This result is only for the quadratic function $f(\mathbf{x}) = \mathbf{x}^\top \mathbf{A}\mathbf{x}$. For non-quadratic functions, the asymptotic rate of convergence of $\mathbf{x}^k$ to a stationary point $\mathbf{x}^*$ is usually determined by the condition number of $\nabla^2 f(\mathbf{x}^*)$.

Example A severely ill-conditioned function is the so called Rosenbrock function:

$$\min \left\{ f(x_1, x_2) = 100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2 \right\}$$

The optimal solution to this problem is easily found to be $(x_1, x_2) = (1, 1)$, with optimal value 0. The gradient is given by

$$\nabla f(\mathbf{x}) = \begin{pmatrix} -400x_1\left(x_2 - x_1^2\right) - 2(1 - x_1) \\ 200\left(x_2 - x_1^2\right) \end{pmatrix},$$

and the Hessian

$$\nabla^2 f(\mathbf{x}) = \begin{pmatrix} -400x_2 + 1200x_1^2 + 2 & -400x_1 \\ -400x_1 & 200 \end{pmatrix}.$$

Evaluating the Hessian at the optimum we obtain

$$\nabla^2 f(1,1) = \begin{pmatrix} 802 & -400 \\ -400 & 200 \end{pmatrix}.$$

This has the high condition number $\kappa = 2508$, which is high. Solving the Rosenbrock problem with gradient descent and backtracking stepsize selection and parameters $\mathbf{x}^0 = (2,5), s = 2, \alpha = 0.25, \beta = 0.5, \varepsilon = 10^{-5}$, leads to 6890(!!) iterations. Figure 5 depicts the slow convergence due to poor conditioning of the Hessian around the optimum.

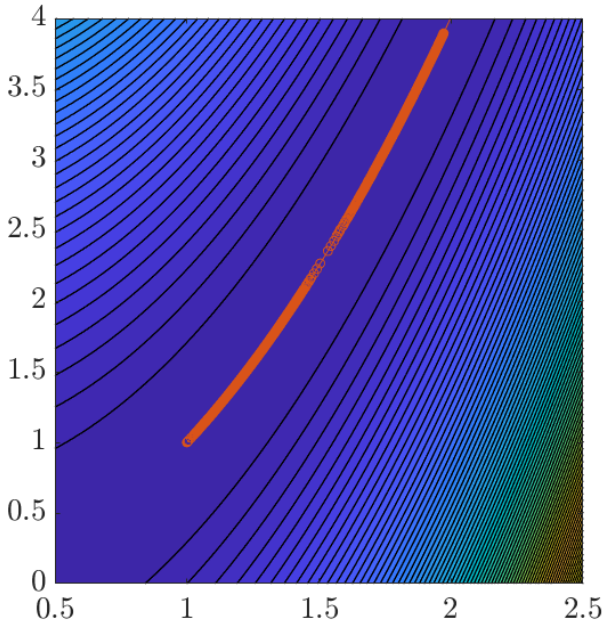

Figure 5: Gradient descent with backtracking line search for the Rosenbrock problem. Convergence towards the optimum $(1,1)$ is achieved after several thousands iterations. The method is extremely slow due to poor conditioning of the Hessian around the optimum.

Scaled Gradient Descent

A way to mitigate the slow convergence due to poor conditioning of the Hessian is to formulate a rescaled version of the problem in order to *condition it*. Consider the minimization problem

$$\min \{f(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\}$$

For a given nonsingular matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$, we make the linear change of variables $\mathbf{x} = \mathbf{S}\mathbf{y}$, and obtain the equivalent problem

$$\min \{g(\mathbf{y}) \equiv f(\mathbf{S}\mathbf{y}) : \mathbf{y} \in \mathbb{R}^n\} .$$

Since $\nabla g(\mathbf{y}) = \mathbf{S}^\top \nabla f(\mathbf{S}\mathbf{y}) = \mathbf{S}^\top \nabla f(\mathbf{x})$, by the chain rule, gradient descent for the rescaled problem reads

$$\mathbf{y}^{k+1} = \mathbf{y}^k - t^k \mathbf{S}^\top \nabla f(\mathbf{S}\mathbf{y}^k) .$$

Multiplying the latter equality by $\mathbf{S}$ from the left, and using the notation $\mathbf{x}^k = \mathbf{S}\mathbf{y}^k$

$$\mathbf{x}^{k+1} = \mathbf{x}^k - t^k \, \mathbf{S}\mathbf{S}^\top \, \nabla f(\mathbf{x}^k)$$

Defining $\mathbf{D} = \mathbf{S}\mathbf{S}^\top$, we obtain scaled gradient descent:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - t^k \mathbf{D} \nabla f(\mathbf{x}^k) .$$

Note that $\mathbf{D} \succ \mathbf{0}$, so the direction $-\mathbf{D}\nabla f\left(\mathbf{x}^k\right)$ is a descent direction:

$$f'(\mathbf{x}^k; -\mathbf{D}\nabla f(\mathbf{x}^k)) = -\nabla f(\mathbf{x}^k)^\top \mathbf{D} \nabla f(\mathbf{x}^k) < 0 .$$

We also allow different scaling matrices at each iteration.

Algorithm 3: Scaled Gradient Descent

Initialization: A tolerance parameter $\varepsilon > 0$ and $\mathbf{x}^0 \in \mathbb{R}^n$.
General Step: for any $k = 0, 1, 2, \ldots$ execute the following steps:

1 Pick a scaling matrix $\mathbf{D}^k \succ 0$

2 Pick a stepsize t^k by a line search procedure on the function

$$g(t) = f\left(\mathbf{x}^k - t\mathbf{D}^k \nabla f\left(\mathbf{x}^k\right)\right)$$

Set $\mathbf{x}^{k+1} = \mathbf{x}^k - t^k \mathbf{D}^k \nabla f\left(\mathbf{x}^k\right)$.

3 If $\left\|\nabla f\left(\mathbf{x}^{k+1}\right)\right\| \leq \varepsilon$, then STOP and $\mathbf{x}^{k+1}$ is the output.

Choosing the Scaling Matrix $\mathbf{D}^k$. The scaled gradient descent method with scaling matrix $\mathbf{D}$ is equivalent to gradient descent employed on the function $g(\mathbf{y}) = f\left(\mathbf{D}^{1/2}\mathbf{y}\right)$, where $\mathbf{D}^{1/2}$ is a matrix such that $\mathbf{D}^{1/2}\mathbf{D}^{1/2} = \mathbf{D}$, and $\mathbf{x} = \mathbf{D}^{1/2}\mathbf{y}$. Note that the gradient of g is

$$\nabla g(\mathbf{y}) = \mathbf{D}^{1/2} \nabla f\left(\mathbf{D}^{1/2}\mathbf{y}\right) = \mathbf{D}^{1/2} \nabla f(\mathbf{x})$$

and so gradient descent for $g(\mathbf{y})$ is

$$\mathbf{y}^{k+1} = \mathbf{y}^k - \mathbf{D}^{1/2} \nabla f(\mathbf{D}^{1/2}\mathbf{y})$$

and left multiplying by $\mathbf{D}^{1/2}$ and substituting $\mathbf{x} = \mathbf{D}^{1/2}\mathbf{y}$ then gives

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \mathbf{D}f(\mathbf{x}).$$

The Hessian of g is

$$\nabla^2 g(\mathbf{y}) = \mathbf{D}^{1/2}\nabla^2 f\left(\mathbf{D}^{1/2}\mathbf{y}\right)\mathbf{D}^{1/2} = \mathbf{D}^{1/2}\nabla^2 f(\mathbf{x})\mathbf{D}^{1/2}$$

The objective is usually to pick $\mathbf{D}^k$ so as to make $\nabla^2 g(\mathbf{y})$ as well-conditioned as possible. A well known choice is to pick $\mathbf{D}^k = \left(\nabla^2 f\left(\mathbf{x}^k\right)\right)^{-1}$, which makes

$$\nabla^2 g(\mathbf{y}) = (\mathbf{D}^k)^{1/2}\nabla^2 f(\mathbf{x}^k)(\mathbf{D}^k)^{1/2} = \mathbf{I},$$

which has a condition number of 1. The gradient descent iteration rule is then

$$\mathbf{x}^{k+1} = \mathbf{x}^k - t^k \nabla^2 f(\mathbf{x}_k)^{-1}\nabla f(\mathbf{x}^k)$$

which is Newton's method[6] for finding the zeros of the equation $\nabla f(\mathbf{x}) = 0$ (i.e., the first order optimality conditions). A drawback of this approach is that it requires us to compute the $n \times n$ Hessian matrix $\nabla^2 f$, and solve a $n \times n$ linear system, which can be prohibitively expensive to compute in many cases.

An alternative is to use a diagonal scaling: $\mathbf{D}^k$ is picked to be diagonal. For example,

$$\left(\mathbf{D}^k\right)_{ii} = \left(\frac{\partial^2 f\left(\mathbf{x}^k\right)}{\partial x_i^2}\right)^{-1}$$

Using diagonal scaling can be very effective when the decision variables are of different magnitudes.

[6] We will discuss this further next week.

Checklist

The idea of this checklist is to help you to self-evaluate your progress and understanding of the subject, and to give you some guidance on where to focus. If you can tick all the boxes it means you're doing alright, otherwise you need to study a bit more, grab a book, watch the videos, or seek help from classmates, the lecturers, or the demonstrators. Try to fill as many gaps as quickly as possible.

And remember to do the exercises!

Learning Outcome	Check
I understand solving $\nabla f(\mathbf{x}) = 0$ is not always feasible by hand and we need numerical methods.	
I am familiar with the idea of having an **iterative method** generating a sequence converging to a stationary point.	
I understand the gradient descent method is composed by a descent direction and a stepsize.	
I understand the three types of line search procedures presented for selecting a stepsize.	
I understand why $-\nabla f(\mathbf{x})$ is the right direction to take.	
I have completed the computer lab tasks.	
I understand the basics of gradient descent algorithm (inputs, outputs, iteration, stopping).	
I understand the Lipschitz gradient property and the convergence theorem for gradient descent.	
I understand the role of the condition number in the convergence speed of gradient descent and the simplest example of scaling.	

Exercises

1. Suppose that f is continuously differentiable, and that $\mathbf{d}$ is a descent direction of f at $\mathbf{x}$. Prove that there exists $\varepsilon > 0$ such that

$$f(\mathbf{x}) - f(\mathbf{x} + t\mathbf{d}) \geq -\alpha t \nabla f(\mathbf{x})^\top \mathbf{d} \quad \forall t \in [0, \varepsilon].$$

2. **Exact line search for quadratic functions.** Find the exact stepsize when $f(\mathbf{x})$ is a quadratic function

$$f(\mathbf{x}) = \mathbf{x}^\top \mathbf{A}\mathbf{x} + 2\mathbf{b}^\top \mathbf{x} + c$$

where $\mathbf{A}$ is an $n \times n$ positive definite matrix, $\mathbf{b} \in \mathbb{R}^n$, and $c \in \mathbb{R}$. Let $\mathbf{x} \in \mathbb{R}^n$ and let $\mathbf{d} \in \mathbb{R}^n$ be a descent direction of f at $\mathbf{x}$. In other words, your objective is to find a solution to

$$\min_{t \geq 0} f(\mathbf{x} + t\mathbf{d}) .$$

3. Consider

$$\min x_1^2 + 2x_2^2.$$

Write down the gradient descent algorithm using

- a fixed stepsize of $t = 0.1$
- exact line search
- backtracking and parameters $\mathbf{x}^0 = (2, 1)$, $s = 2$, $\alpha = 0.25$, $\beta = 0.5$, and $\varepsilon = 10^{-5}$

Implement each case on a computer, check you get the same results as shown in the notes, and record how many steps are required for the algorithm to converge. Plot the three different trajectories on the same figure.

For the fixed stepsize method, what is the largest stepsize that can used and the algorithm still converge?

4. Now solve

$$\min 0.01x_1^2 + x_2^2$$

using gradient descent with backtracking using the same parameters as above. How many steps did this take till convergence?

Implement scaled gradient descent for this problem using the diagonal scaling matrix given in the notes. Try using a fixed step size of $\bar{t} = 1$ - how many iterations are required for the method to converge?

5. Show that $f(x) = \sqrt{1 + x^2} \in C_1^{1,1}$.

6. Let $f(\mathbf{x}) = \mathbf{x}^\top \mathbf{A}\mathbf{x} + 2\mathbf{b}^\top \mathbf{x} + c$, where $\mathbf{A}$ is a symmetric $n \times n$ matrix etc. Show that the smallest Lipschitz constant of ∇f is $2||\mathbf{A}||$.

7. In the notes we stated that for gradient descent we have

$$f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k) - M\|\nabla f(\mathbf{x}^k)||^2.$$

If $f^* = \min f(\mathbf{x})$, show that

$$M \sum_{k=0}^{N} ||\nabla f(\mathbf{x}^k)||^2 \leq f(\mathbf{x}^0) - f^*$$

for some constant M. Now let $g_N^* = \min_{0 \leq k \leq N} \|\nabla f(\mathbf{x}^k)\|$. Show that

$$g_N^* \leq \frac{1}{\sqrt{N+1}} \left[\frac{1}{M} (f(\mathbf{x}^0) - f^*) \right]^{\frac{1}{2}}$$

This result tells us about the rate of convergence of $||\nabla f(\mathbf{x})||$.

8. Implement the Rosenbrock example from the notes.

9. It is possible to prove that having a Lipschitz continuous gradient with constant L is equivalent to

$$|f(\mathbf{x}) - f(\mathbf{x}_0) - \nabla f(\mathbf{x}_0)^\top (\mathbf{x} - \mathbf{x}_0)| \leq \frac{L}{2} \|\mathbf{x} - \mathbf{x}_0\|_2^2.$$

By removing the absolute signs, show that we can bound any $f \in C^{1,1}$ above and below by a quadratic function that are tight at $\mathbf{x}_0$. In other words, show that there exists quadratic functions $u(\mathbf{x})$ and $l(\mathbf{x})$ with

$$l(\mathbf{x}) \leq f(\mathbf{x}) \leq u(\mathbf{x}) \qquad \text{and} \qquad f(\mathbf{x}_0) = u(\mathbf{x}_0) = l(\mathbf{x}_0).$$

Draw a picture.