Some indications and tips for the gradient algorithms

1) Gradient

The gradient is equivalent to the derivative in n-dimension :

depends on x

$$^t\nabla f(x) = \left(\frac{\partial f}{\partial x_1}(x) \quad \frac{\partial f}{\partial x_2}(x) \quad \dots \quad \frac{\partial f}{\partial x_n}(x)\right)$$

defined everywhere
f can be derivated

In 1D : $^t\nabla f(x) = \left(\frac{\partial f}{\partial x_1}(x)\right) = \left(f'(x)\right)$

If the gradient is $\neq \vec{0}$ then the point x is not a local minimum.

We can follow the opposite of the gradient to decrease the function f.

$(\exists \varepsilon > 0 \;/\; \forall x_\varepsilon = x - \varepsilon \vec{\nabla} f(x) \quad f(x_\varepsilon) < f(x))$

(Similarly, we can follow $\nabla f(x)$ to increase f)

The classical gradient algorithm is

- Let x be any point
- While $\nabla f(x) \neq 0$

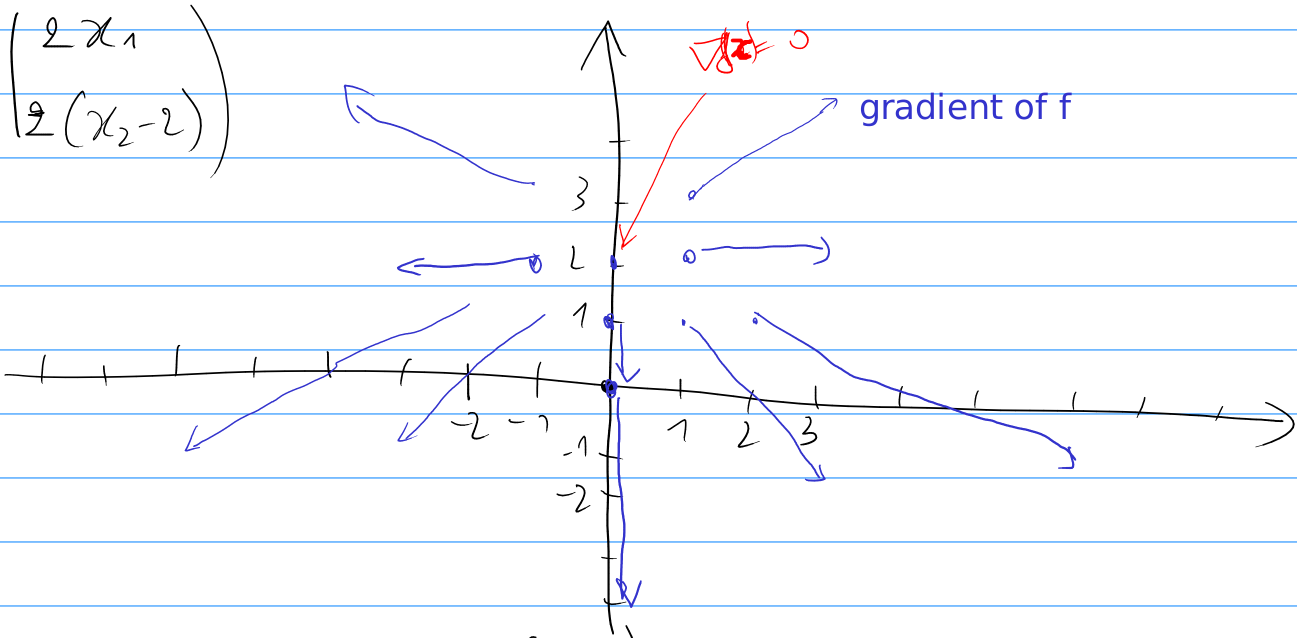
  Find $x_2 = \underbrace{\arg\min_{\alpha \geq 0} f(x - \alpha \nabla f(x))}_{\text{Point minimizing f on the direction } \nabla f(x)}$

  $x \leftarrow x_2$

The reduced gradient algorithm and the projected gradient algorithm work the same way if there is no constraint.

Example with $\min \left(x_1^2 + (x_2 - 2)^2 \right)$

$$\nabla f(x) = \begin{pmatrix} 2x_1 \\ 2(x_2-2) \end{pmatrix}$$



If we follow the gradient from $\begin{pmatrix} 1 \\ 1 \end{pmatrix}$

$$\nabla f\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ -2 \end{pmatrix} \Rightarrow -\nabla f\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -2 \\ 2 \end{pmatrix}$$

$$f(x - \alpha \nabla f(x)) = f\begin{pmatrix} 1 - 2\alpha \\ 1 + 2\alpha \end{pmatrix}$$

$$= (1-2\alpha)^2 + (1+2\alpha-2)^2$$

$$= (1-2\alpha)^2 + (2\alpha-1)^2$$

$$= 2(2\alpha-1)^2$$

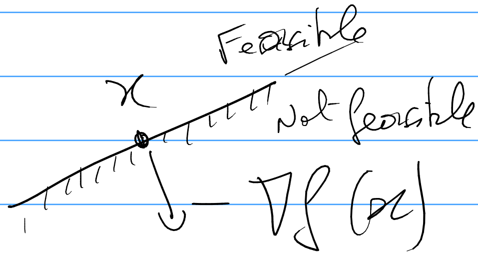
This is minimum if $\alpha = \frac{1}{2}$

$$\Rightarrow x_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \frac{1}{2}\begin{pmatrix} -2 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

Finally $\nabla f(x_2) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Rightarrow$ end

The reduced gradient and projected gradient algorithms are used when there are some contraints. The gradient algorithm may not work in that case.

For instance



---

2) Projected gradient when $\vec{d} = 0$

At some point we can have $\vec{d} = 0$

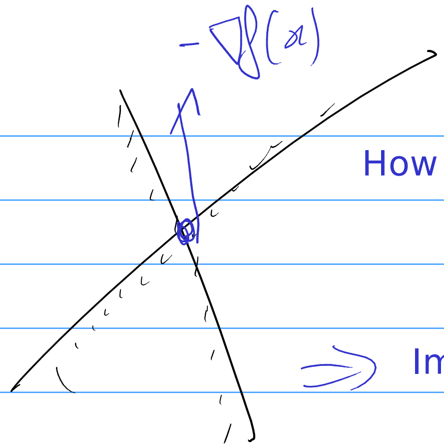
The projection is $\vec{0}$ in two cases.

this is a particular case of that

- You project $-\nabla f(x)$ on $\{\vec{0}\}$
- You project $-\nabla f(x)$ on L with $L \perp -\nabla f(x)$

The first case happens when you project the gradient on too many constraints.

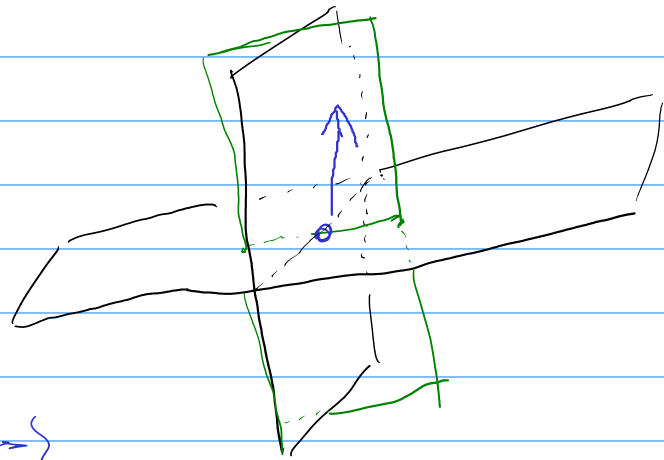
In 2D



How can we project a vector on 2 lines

$\Rightarrow$ Impossible : $L = \{\vec{0}\}$

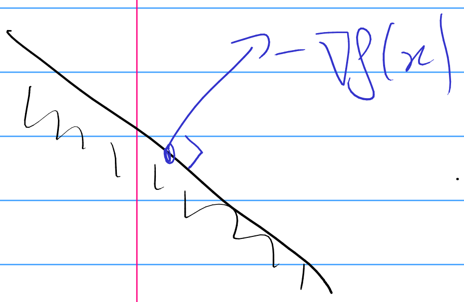
In 3D



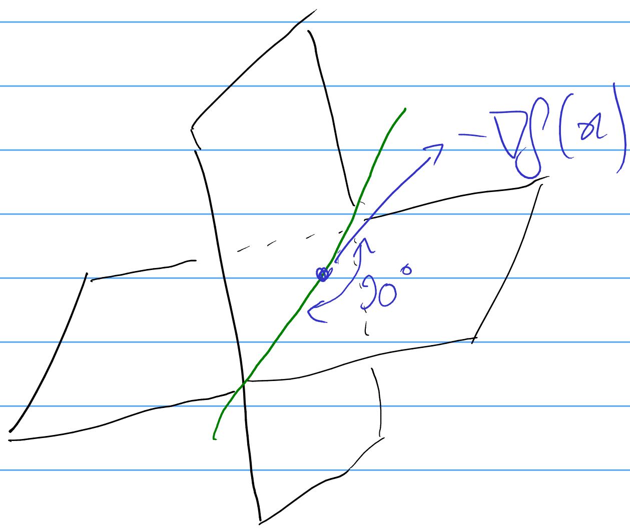
How to project a vector on 3 planes?

$\Rightarrow$ Impossible $L = \{\vec{0}\}$

On the second case, the gradient is orthogonal to the intersection of the constraints.



2D



3D

3) Sign of λ

When $\vec{d} = 0$ we can write

$$-\nabla f = \sum_{i \in I(x)} \lambda_i \nabla g_i(x) + \sum_{j \in J} \mu_j \nabla h_j(x)$$

Indeed, the gradient of each constraint is orthogonal to that contraint. If the grad
of f is orthogonal to the intersection of the contraints, then it is a linear combina
of the gradients of the contraints.

The sign of λ_i is important

$$\lambda_i \geq 0 \quad \forall i \Rightarrow KKT$$

$\Rightarrow$ end of the algorithm

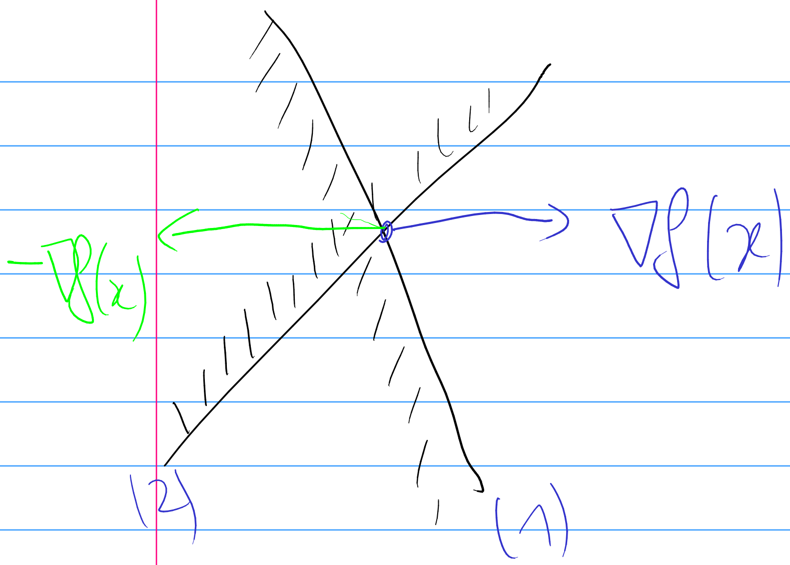
$\lambda_i < 0 \quad \exists i \Rightarrow$ there exists a useless contrain

A useless constraint is a constraint you can (temporarily) remove.
==> That constraint make the projection nul
==> It prevents us to find a better solution
==> The fact that lambda_i is negative means that KKT is not satisfied : there
exists a better solution locally.
==> Removing the constraint make the projection not nul, we can then follow
new direction.

How to detect a useless constraint on the drawing ?
==> By trying to (temporarily) remove every constraint one by one.
==> If projecting on the remaining constraints produces a valid direction (not
violating the removed constraint) then that constraint is useless.

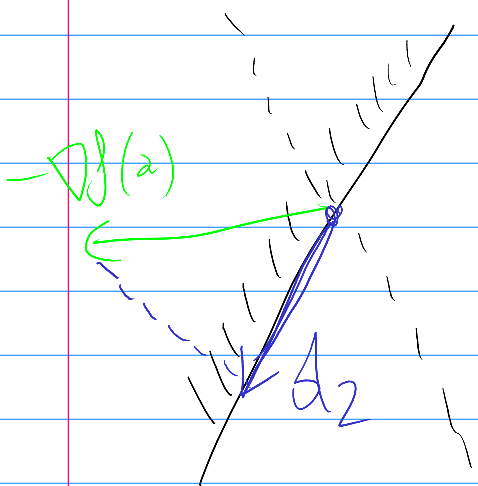
If a constraint is useless : lambda_i < 0
otherwise lambda_i >= 0.

Examples



If you project $-\nabla f(x)$ on (1) and (2), you get $\vec{0}$

If you remove (1)



We can now project on (2) to get d_2

However, if we re-add (1), we see that we cannot follow d2 as it would violate (1).

$\Rightarrow$ (1) is not useless.

If you remove (2)

We can now project on (1) to get d1

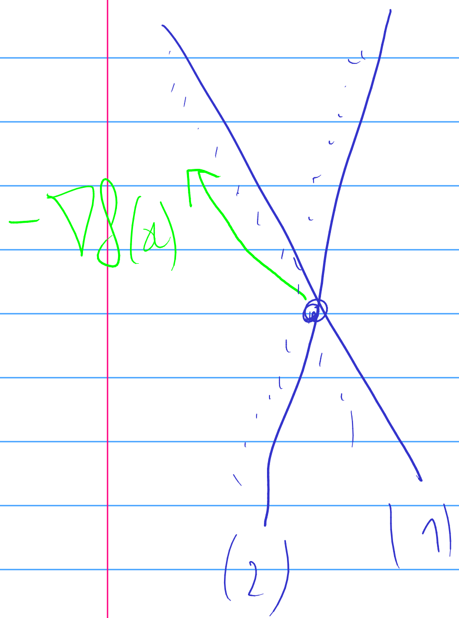
However, if we re-add (2), we see that we cannot follow d1 as it would violate (2).

$\Rightarrow$ (2) is also not useless

$\Rightarrow$ In some way we are stuck, we cannot follow any direction to decrease f this means that KKT is satisfied.

$\Rightarrow$ in other words: λ_1 and $\lambda_2 \geq 0$

Now consider that example:



Again $\vec{\delta} = 0$

If you remove (1)

It seemes we can follow d2 without violating (1)

$\Rightarrow$ (1) is useless

If you remove (2)

We cannot follow d1 as it would violate (2)

$\Rightarrow$ (2) is not useless

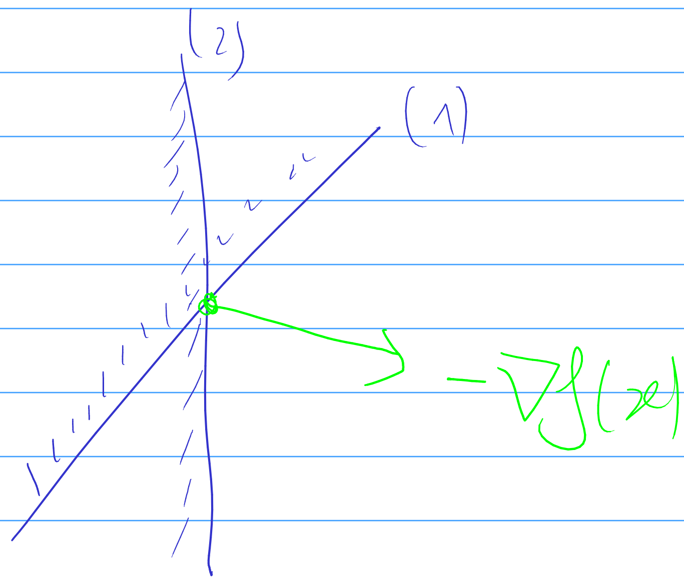
We can follow some direction to decrease f ==> We are not stuck

$\Rightarrow$ KKT is not satisfied

$\Rightarrow$ Some λ is < 0

Here $\lambda_1 < 0$ and $\lambda_2 > 0$

We can have more than one negative lambda $\lambda_i < 0$



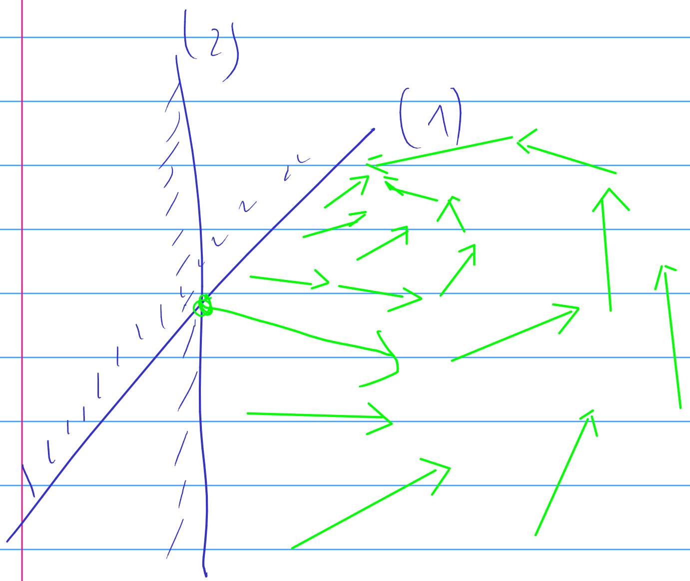
Clearly (1) and (2) are useless and can be removed.

Recall that the removal of a constraint is always temporary.

You must re-add the constraint after the projection is done!

(the reason is that f is not linear and the gradient can indicate a direction that is locally optimal but that does not point at an optimal solution following the gradient continuously can force you to make a half-turn)

You may have something like this:



opposite of the gradient