Descent methods

for unconstrained optimization

Gilles Gasso

INSA Rouen - ASI Departement
Laboratory LITIS

September 25, 2023

Plan

Unconstrained optimization

Elements of the problem

- $\boldsymbol{\theta} \in \mathbb{R}^d$: vector of unknown real parameters
- $J : \mathbb{R}^d \to \mathbb{R}$: the function to be minimized.
- Assumption: J is differentiable all over its domain
  $\text{dom} J = \left\{\boldsymbol{\theta} \in \mathbb{R}^d \mid J(\boldsymbol{\theta}) < \infty\right\}$

Problem formulation

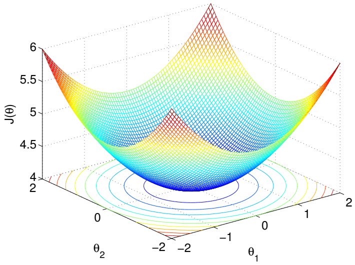
$$(P) \quad \min_{\boldsymbol{\theta} \in \mathbb{R}^d} J(\boldsymbol{\theta})$$

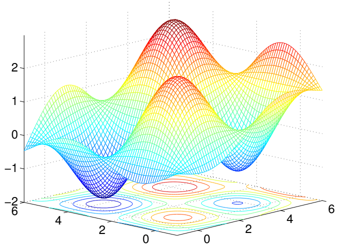
Unconstrained optimization

Examples

$$J(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^\top \boldsymbol{P}\boldsymbol{\theta} + q^\top \boldsymbol{\theta} + r$$

with $\boldsymbol{P}$ a positive definite matrix





$$J(\boldsymbol{\theta}) = \cos(\theta_1 - \theta_2) + \sin(\theta_1 + \theta_2) + \frac{\theta_1}{4}$$

Different solutions

Global solution

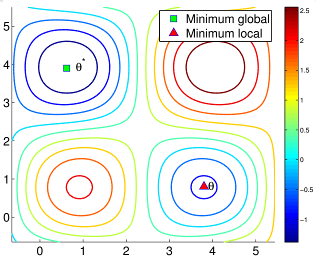
$\boldsymbol{\theta}^*$ is said to be the global minimum solution of the problem if
$J(\boldsymbol{\theta}^*) \leq J(\boldsymbol{\theta}), \quad \forall \boldsymbol{\theta} \in domJ$

Local solution

$\hat{\boldsymbol{\theta}}$ is a local minimum solution of problem (P) if it holds
$J(\hat{\boldsymbol{\theta}}) \leq J(\boldsymbol{\theta}), \ \forall \boldsymbol{\theta} \in domJ$ *such that* $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\| \leq \epsilon, \epsilon > 0$

Illustration

$J(\boldsymbol{\theta}) = \cos(\theta_1 - \theta_2) + \sin(\theta_1 + \theta_2) + \frac{\theta_1}{4}$



Optimality conditions

- How do we assess a solution to the problem?

First order necessary condition

Theorem [First order condition]

Let $J : \mathbb{R}^d \to \mathbb{R}$ be a differential function on its domain. A vector $\boldsymbol{\theta}_0$ is a (local or global) solution of the problem (P), if it necessarily satisfies the condition $\nabla J(\boldsymbol{\theta}_0) = 0$.

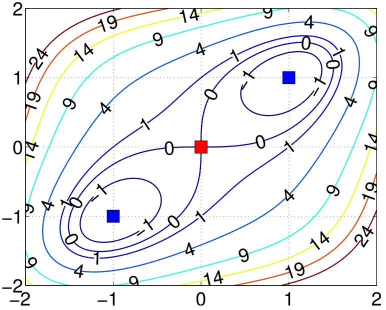
Vocabulary

- Any vector $\boldsymbol{\theta}_0$ that verifies $\nabla J(\boldsymbol{\theta}_0) = 0$ is called a stationary point or critical point
- $\nabla J(\boldsymbol{\theta}) \in \mathbb{R}^d$ is the gradient vector of J at $\boldsymbol{\theta}$.
- The gradient is the unique vector such that the directional derivative can be written as:

$$\lim_{t \to 0} \frac{J(\boldsymbol{\theta} + t\mathrm{h}) - J(\boldsymbol{\theta})}{t} = \nabla J(\boldsymbol{\theta})^\top \mathrm{h}, \quad \mathrm{h} \in \mathbb{R}^d, \quad t \in \mathbb{R}$$

Example of a first order optimality condition

- $J(\boldsymbol{\theta}) = \theta_1^4 + \theta_2^4 - 4\theta_1\theta_2$
- Gradient $\nabla J(\boldsymbol{\theta}) = \begin{pmatrix} 4\theta_1^3 - 4\theta_2 \\ -4\theta_1 + 4\theta_2^3 \end{pmatrix}$
- Stationary points that verify $\nabla J(\boldsymbol{\theta}) = 0$.
- Three solutions $\boldsymbol{\theta}^{(1)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\boldsymbol{\theta}^{(2)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $\boldsymbol{\theta}^{(3)} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$



Remarks

- $\boldsymbol{\theta}^{(2)}$ and $\boldsymbol{\theta}^{(3)}$ are local minimal but not $\boldsymbol{\theta}^{(1)}$
- every stationary point can be deemed a local extremum

We need another optimality condition

How to ensure that a stationary point is a minimum solution?

Hessian matrix

Twice differential function

$J : \mathbb{R}^d \to \mathbb{R}$ is said to be a twice differentiable function on its domain dom$J$ if, at every point $\boldsymbol{\theta} \in$, there exists a unique symmetric matrix $\boldsymbol{H}(\boldsymbol{\theta}) \in \mathbb{R}^{d\times d}$ called Hessian matrix such that
$J(\boldsymbol{\theta} + \mathrm{h}) = J(\boldsymbol{\theta}) + \nabla J(\boldsymbol{\theta})^\top \mathrm{h} + \mathrm{h}^\top \boldsymbol{H}(\boldsymbol{\theta})\mathrm{h} + \|\mathrm{h}\|^2 \varepsilon(\mathrm{h})$.
$\varepsilon(\mathrm{h})$ *is a continuous function at* 0 *with* $\lim_{\mathrm{h}\to 0} \varepsilon(\mathrm{h}) = 0$

- $\boldsymbol{H}(\boldsymbol{\theta})$ is the second derivative matrix

$$\boldsymbol{H}(\boldsymbol{\theta}) = \begin{pmatrix} \frac{\partial^2 J}{\partial_{\theta_1}\partial_{\theta_1}} & \frac{\partial^2 J}{\partial_{\theta_1}\partial_{\theta_2}} & \cdots & \frac{\partial^2 J}{\partial_{\theta_1}\partial_{\theta_d}} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial^2 J}{\partial_{\theta_d}\partial_{\theta_1}} & \frac{\partial^2 J}{\partial_{\theta_d}\partial_{\theta_2}} & \cdots & \frac{\partial^2 J}{\partial_{\theta_d}\partial_{\theta_d}} \end{pmatrix}$$

- $\boldsymbol{H}(\boldsymbol{\theta}) = \nabla_{\theta^\top}(\nabla_\theta J(\boldsymbol{\theta}))$ is the Jacobian of the gradient function

Examples

Example 1

- Objective function
  $J(\boldsymbol{\theta}) = \theta_1^4 + \theta_2^4 - 4\theta_1\theta_2$

- Gradient
  $\nabla J(\boldsymbol{\theta}) = \begin{pmatrix} 4\theta_1^3 - 4\theta_2 \\ -4\theta_1 + 4\theta_2^3 \end{pmatrix}$

- Hessian matrix
  $\boldsymbol{H}(\boldsymbol{\theta}) = \begin{pmatrix} 12\theta_1^2 & -4 \\ -4 & 12\theta_2^2 \end{pmatrix}$

Exemple 2

- Quadratic objective function
  $J(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^\top \boldsymbol{P}\boldsymbol{\theta} + \mathrm{q}^\top\boldsymbol{\theta} + r$

- Directional derivative
  $D(\mathrm{h}, \boldsymbol{\theta}) = \lim_{t\to 0} \frac{J(\boldsymbol{\theta}+t\mathrm{h}) - J(\boldsymbol{\theta})}{t}$

  $D(\mathrm{h}, \boldsymbol{\theta}) = (\boldsymbol{P}\boldsymbol{\theta} + \mathrm{q})^\top \mathrm{h}$

- Gradient $\nabla J(\boldsymbol{\theta}) = \boldsymbol{P}\boldsymbol{\theta} + \mathrm{q}$

- Hessian matrix $\boldsymbol{H}(\boldsymbol{\theta}) = \boldsymbol{P}$

Second order optimality condition

Theorem [Second order optimality condition]

Let $J : \mathbb{R}^d \to \mathbb{R}$ be a twice differentiable function on its domain. If $\boldsymbol{\theta}_0$ is a minimum of J, then $\nabla J(\boldsymbol{\theta}_0) = 0$ and $\boldsymbol{H}(\boldsymbol{\theta}_0)$ is a positive definite matrix.

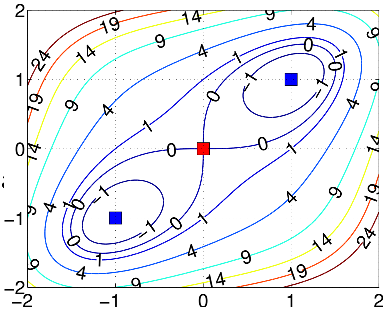
Remarks

- $\boldsymbol{H}$ is positive definite if and only if all its eigenvalues are positive
- $\boldsymbol{H}$ is negative definite if and only if all its eigenvalues are negative
- For $\theta \in \mathbb{R}$, this condition means that the gradient of J at the minimum is null, $J'(\theta) = 0$ and its second derivative is positive i.e. $J''(\theta) > 0$
- If at a stationary point $\boldsymbol{\theta}_0$, $\boldsymbol{H}(\boldsymbol{\theta}_0))$ is negative definite, $\boldsymbol{\theta}_0$ is a local maximum of J

Illustration of the second order optimality condition

- $J(\boldsymbol{\theta}) = \theta_1^4 + \theta_2^4 - 4\theta_1\theta_2$
- Gradient : $\nabla J(\boldsymbol{\theta}) = \begin{pmatrix} 4\theta_1^3 - 4\theta_2 \\ -4\theta_1 + 4\theta_2^3 \end{pmatrix}$
- Stationary points : $\boldsymbol{\theta}^{(1)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\boldsymbol{\theta}^{(2)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

  $\boldsymbol{\theta}^{(3)} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$



- Hessian matrix $\boldsymbol{H}(\boldsymbol{\theta}) = \begin{pmatrix} 12\theta_1^2 & -4 \\ -4 & 12\theta_2^2 \end{pmatrix}$

	$\boldsymbol{\theta}^{(1)}$	$\boldsymbol{\theta}^{(2)}$	$\boldsymbol{\theta}^{(3)}$
Hessian	$\begin{pmatrix} 0 & -4 \\ -4 & 0 \end{pmatrix}$	$\begin{pmatrix} 12 & -4 \\ -4 & 12 \end{pmatrix}$	$\begin{pmatrix} 12 & -4 \\ -4 & 12 \end{pmatrix}$
Eigenvalues	$4, -4$	$8, 16$	$8, 16$
Type of solution	Saddle point	Minimum	Minimum

Necessary and sufficient optimality condition

Theorem [2nd order sufficient condition]

Assume the hessian matrix $\boldsymbol{H}(\boldsymbol{\theta}_0)$ of $J(\boldsymbol{\theta})$ at $\boldsymbol{\theta}_0$ exists and is positive definite. Assume also the gradient $\nabla J(\boldsymbol{\theta}_0) = 0$. Then $\boldsymbol{\theta}_0$ is a (local or global) minimum of problem (P).

Theorem [Sufficient and necessary optimality condition]

Let J be a convex function. Every local solution $\hat{\boldsymbol{\theta}}$ is a global solution $\boldsymbol{\theta}^*$.

Recall

A function $J : \mathbb{R}^d \rightarrow \mathbb{R}$ is convex if it verifies

$$J(\alpha\boldsymbol{\theta} + (1-\alpha)\mathrm{z}) \leq \alpha J(\boldsymbol{\theta}) + (1-\alpha)J(\mathrm{z}), \quad \forall \boldsymbol{\theta}, \mathrm{z} \in \mathrm{dom} J, \quad 0 \leq \alpha \leq 1$$

How to find the solution(s)?

- We have seen how to assess a solution to the problem
- A question to be addressed now is how to compute a solution?

Principle of descent algorithms

Direction of descent

Let the function $J : \mathbb{R}^d \to \mathbb{R}$. The vector $\mathrm{h} \in \mathbb{R}^d$ is called a descent direction in $\boldsymbol{\theta}$ if there exists $\alpha > 0$ such that $J(\boldsymbol{\theta} + \alpha \mathrm{h}) < J(\boldsymbol{\theta})$

Principle of descent methods

- Start from an initial point $\boldsymbol{\theta}_0$
- Design a sequence of points $\{\boldsymbol{\theta}_k\}$ with $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \alpha_k \mathrm{h}_k$
- Ensure that the sequence $\{\boldsymbol{\theta}_k\}$ converges to a stationary point $\hat{\boldsymbol{\theta}}$

- h_k: direction of descent
- α_k: step size

General approach

General algorithm

1: Let $k = 0$, initialize $\boldsymbol{\theta}_k$

2: **repeat**

3: Find a descent direction $\mathrm{h}_k \in \mathbb{R}^d$

4: Line search: find a step size $\alpha_k > 0$ in the direction h_k such that $J(\boldsymbol{\theta}_k + \alpha_k \mathrm{h}_k)$ decreases "enough"

5: Update: $\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k + \alpha_k \mathrm{h}_k$ and $k \leftarrow k + 1$

6: **until** $\|\nabla J(\boldsymbol{\theta}_k)\| < \epsilon$

- The methods of descent differ by the choice of:
  - h: gradient algorithm, Newton, Quasi-Newton algorithm
  - α: backtracking...

Gradient Algorithm

Theorem [descent direction and opposite direction of gradient]

Let $J(\boldsymbol{\theta})$ be a differential function. The direction $\mathrm{h} = -\nabla J(\boldsymbol{\theta}) \in \mathbb{R}^d$ is a descent direction.

Proof.

J being differentiable, for any $t > 0$ we have
$J(\boldsymbol{\theta} + t\mathrm{h}) = J(\boldsymbol{\theta}) + t\nabla J(\boldsymbol{\theta})^\top \mathrm{h} + t\|\mathrm{h}\|\epsilon(t\mathrm{h})$. Setting $\mathrm{h} = -\nabla J(\boldsymbol{\theta})$, we get
$J(\boldsymbol{\theta} + t\mathrm{h}) - J(\boldsymbol{\theta}) = -t\|\nabla J(\boldsymbol{\theta})\|^2 + t\|\mathrm{h}\|\epsilon(th)$. For t small enough $\epsilon(t\mathrm{h}) \to 0$ and so $J(\boldsymbol{\theta} + t\mathrm{h}) - J(\boldsymbol{\theta}) = -t\|\nabla J(\boldsymbol{\theta})\|^2 < 0$. It is then a descent direction. □

Characteristics of the gradient algorithm

- Choice of the descent direction at $\boldsymbol{\theta}_k$: $\mathrm{h}_k = -\nabla J(\boldsymbol{\theta}_k)$
- Complexity of the update: $\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k - \alpha_k \nabla J(\boldsymbol{\theta}_k)$ costs $\mathcal{O}(d)$

Newton algorithm

- 2nd order approximation of J at $\boldsymbol{\theta}_k$

$$J(\boldsymbol{\theta} + \mathrm{h}) \approx J(\boldsymbol{\theta}_k) + \nabla J(\boldsymbol{\theta}_k)^\top \mathrm{h} + \frac{1}{2}\mathrm{h}^\top \boldsymbol{H}(\boldsymbol{\theta}_k)\mathrm{h}$$

  with $\boldsymbol{H}(\boldsymbol{\theta}_k)$ the positive definite Hessian matrix
- The direction h_k which minimizes this approximation is obtained by

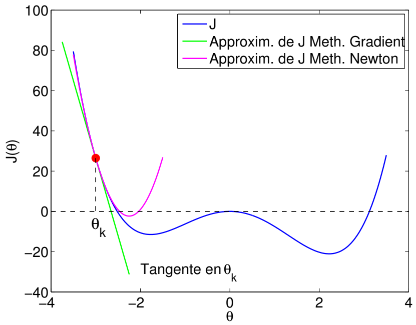
$$\nabla J(\boldsymbol{\theta} + \mathrm{h}_k) = 0 \quad \Rightarrow \quad \mathrm{h}_k = -\boldsymbol{H}(\boldsymbol{\theta}_k)^{-1}\nabla J(\boldsymbol{\theta}_k)$$

Features

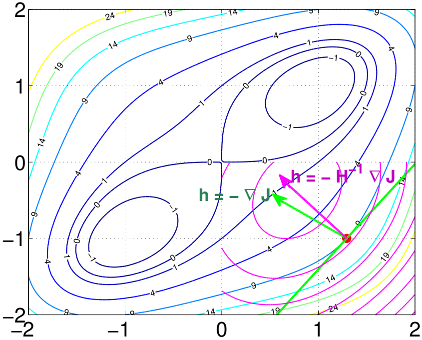
- Descent direction at $\boldsymbol{\theta}_k$: $\mathrm{h}_k = -\boldsymbol{H}(\boldsymbol{\theta}_k)^{-1}\nabla J(\boldsymbol{\theta}_k)$
- Complexity of the update: $\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k - \alpha_k \boldsymbol{H}(\boldsymbol{\theta}_k)^{-1}\nabla(\boldsymbol{\theta}_k)$ costs $\mathcal{O}(d^3)$ flops
- $\boldsymbol{H}(\boldsymbol{\theta}_k)$ is not always guaranteed to be positive definite matrix. Hence we cannot always ensure that h_k is a direction of descent

Illustration of gradient and Newton methods

Local approximation of the two methods in 1D



Directions of descent in 2D



Quasi-Newton method

Main features

- Descent direction at $\boldsymbol{\theta}_k$: $\mathrm{h}_k = -\mathrm{B}(\boldsymbol{\theta}_k)^{-1}\nabla J(\boldsymbol{\theta}_k)$
- $\mathrm{B}(\boldsymbol{\theta}_k)$ is an positive definite approximation of the Hessian matrix
- Complexity of the update: most of the times $\mathcal{O}(d^2)$

Line search

Assume the direction of descent h_k at $\boldsymbol{\theta}_k$ is fixed. We aim to find the step size $\alpha_k > 0$ in the direction h_k such that the function $J(\boldsymbol{\theta}_k + \alpha_k h_k)$ decreases enough (compared to $J(\boldsymbol{\theta}_k)$)

Several options

- Fixed step size: use a fixed value $\alpha > 0$ at each iteration k

$$\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k + \alpha h_k$$

- Optimal step size α_k^*

$$\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k + \alpha_k^* h_k \quad \text{with} \quad \alpha_k^* = \arg\min_{\alpha > 0} J(\boldsymbol{\theta}_k + \alpha h_k)$$

- Variable step size: the choice α_k is adapted to the current iteration

$$\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k + \alpha_k h_k$$

Line search

Pros and cons

- Fixed step size strategy: often not very effective
- Optimal step size: can be costly in calculation time
- Variable step: most commonly used approach
  - The step is often imprecise
  - A trade-off between computation cost and decrease of J

Variable step sizeh

Armijo's rule

Determine the step size α_k in order to have a sufficient decrease of J i.e.

$$J(\boldsymbol{\theta}_k + \alpha_k \mathrm{h}) \leq J(\boldsymbol{\theta}_k) + c\,\alpha_k \nabla J(\boldsymbol{\theta}_k)^\top \mathrm{h}_k$$

- Usually c is chosen in the range $\left[10^{-5}, 10^{-1}\right]$
- Having h_k the direction of descent, we have $\nabla J(\boldsymbol{\theta}_k)^\top \mathrm{h}_k < 0$, which ensures the decrease of J

Backtracking

1: Fix an initial step $\bar{\alpha}$, choose $0 < \rho < 1$, $\alpha \leftarrow \bar{\alpha}$
2: **repeat**
3: $\quad \alpha \leftarrow \rho\alpha$
4: **until** $J(\boldsymbol{\theta}_k + \alpha \mathrm{h}) > J(\boldsymbol{\theta}_k) + c\,\alpha \nabla J(\boldsymbol{\theta}_k)^\top \mathrm{h}_k$

Choice of the initial step

- Newton method: $\bar{\alpha} = 1$
- Gradient method: $\bar{\alpha} = 2\frac{J(\boldsymbol{\theta}_k) - J(\boldsymbol{\theta}_{k-1})}{\nabla J(\boldsymbol{\theta}_k)^\top \mathbf{h}_k}$

Interpretation: as long as J does not decrease, we decrease the value of the step size

Summary of descent methods

General algorithm

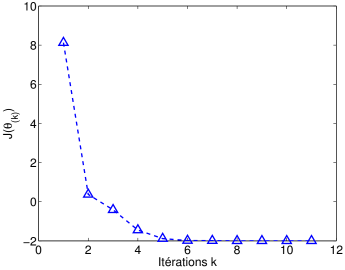
1: Initialize θ_k
2: **repeat**
3: Find direction of descent $\mathrm{h}_k \in \mathbb{R}^d$
4: Line search: find the step $\alpha_k > 0$
5: Update: $\boldsymbol{\theta}_{k+1} \leftarrow \boldsymbol{\theta}_k + \alpha_k \mathrm{h}_k$
6: **until** convergence

Method	Direction of descent h	Complexity	Convergence
Gradient	$-\nabla J(\boldsymbol{\theta})$	$\mathcal{O}(d)$	linear
Quasi-Newton	$-B(\boldsymbol{\theta})^{-1}\nabla J(\boldsymbol{\theta})$	$\mathcal{O}(d^2)$	superlinear
Newton	$-\boldsymbol{H}(\boldsymbol{\theta})^{-1}\nabla J(\boldsymbol{\theta})$	$\mathcal{O}(d^3)$	quadratic

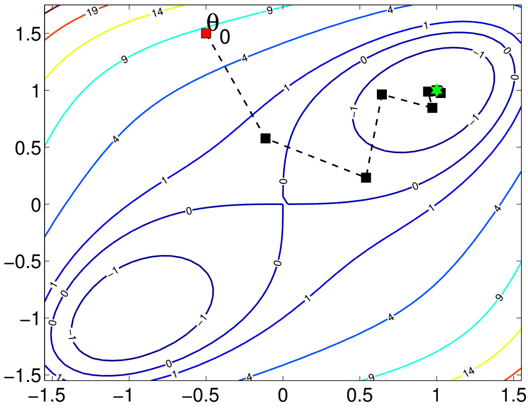
- Step size computation: backtracking (common) or optimal step size
- Complexity of each method: depends on the complexity of calculating h, the search for α, and the number of iterations performed until convergence

Gradient method

J along the iterations

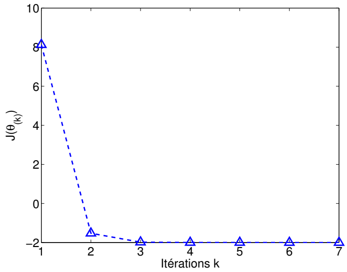


Evolution of the iterates



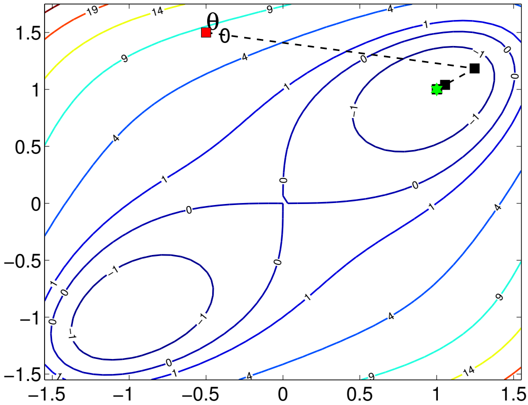
Newton method

J along the iterations



- At each iteration we considered the matrix $\boldsymbol{H}(\boldsymbol{\theta}) + \lambda \mathrm{I}$ instead of $\boldsymbol{H}$ to guarantee the positive definite property of Hessian

Evolution of the iterates



Conclusion

- Unconstrained optimization of smooth objective function
- Characterization of the solution(s) requires checking the optimality conditions
- Computation of a solution using descent methods
    - Gradient descent method
    - Newton method
- Not covered in this lecture:
    - Convergence analysis of the studied algorithms
    - Non-smooth optimization
    - Gradient-free optimzation