ESQC 2024

Mathematical Methods Lecture 4

By Simen Kvaal

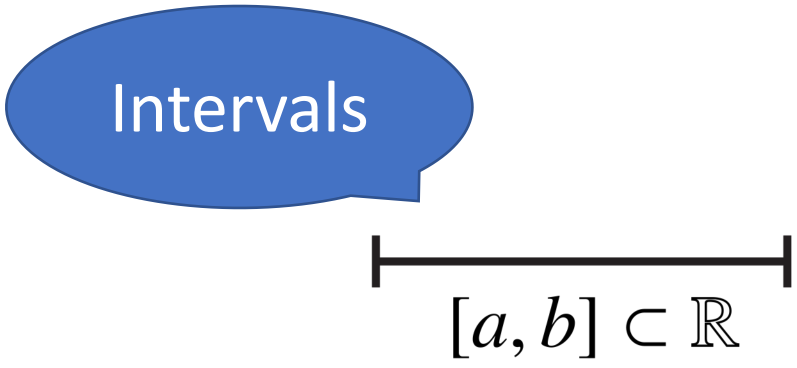
Vector Calculus

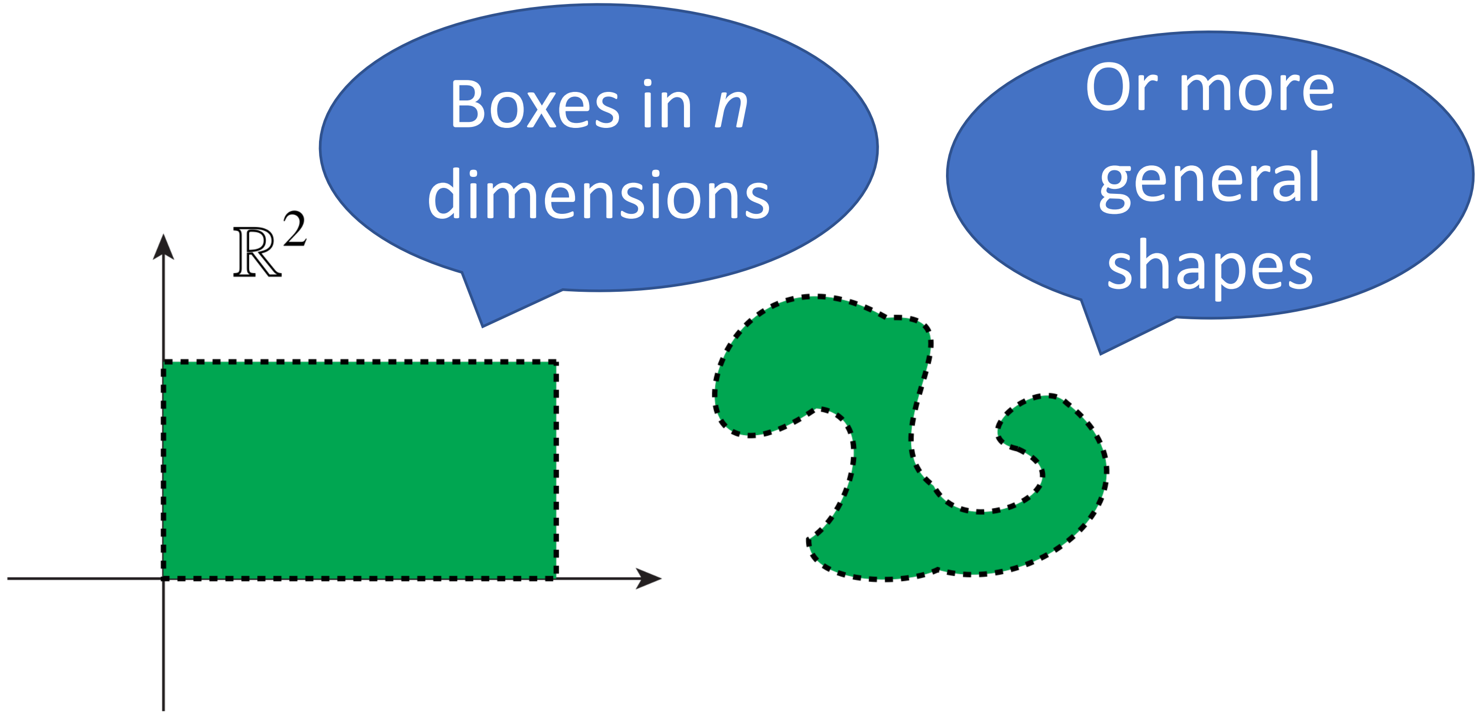
Functions of several variables

- We turn to the study of functions **between vector spaces:**

$$\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m \qquad\qquad \mathbf{f} : \Omega(\subset \mathbb{R}^n) \to \mathbb{R}^m$$

- A typical **domain** Ω:

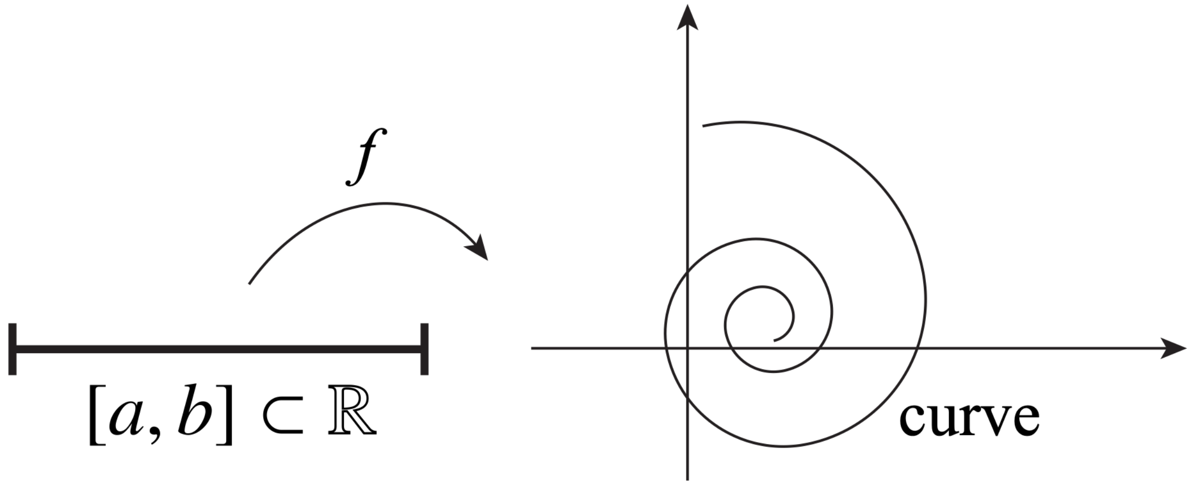




Examples

- Curves:

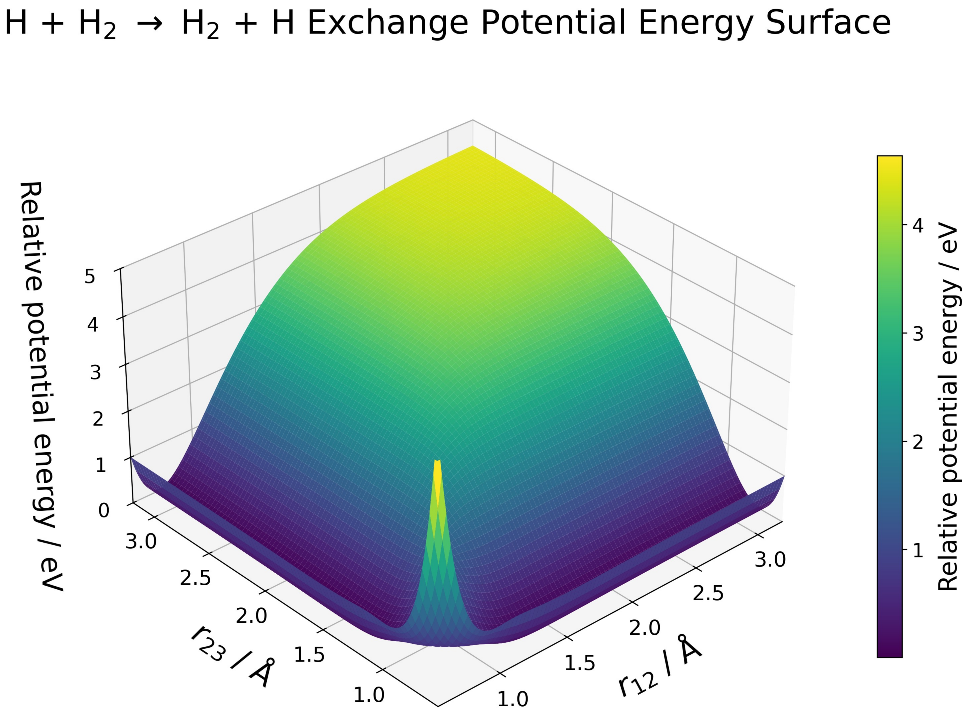
$$f : \mathbb{R} \to \mathbb{R}^m$$



- Scalar fields:

$$f : \mathbb{R}^n \to \mathbb{R}$$

$f(x, y) = \text{intensity}$



$V(r_{12}, r_{23}) = \text{potential energy}$

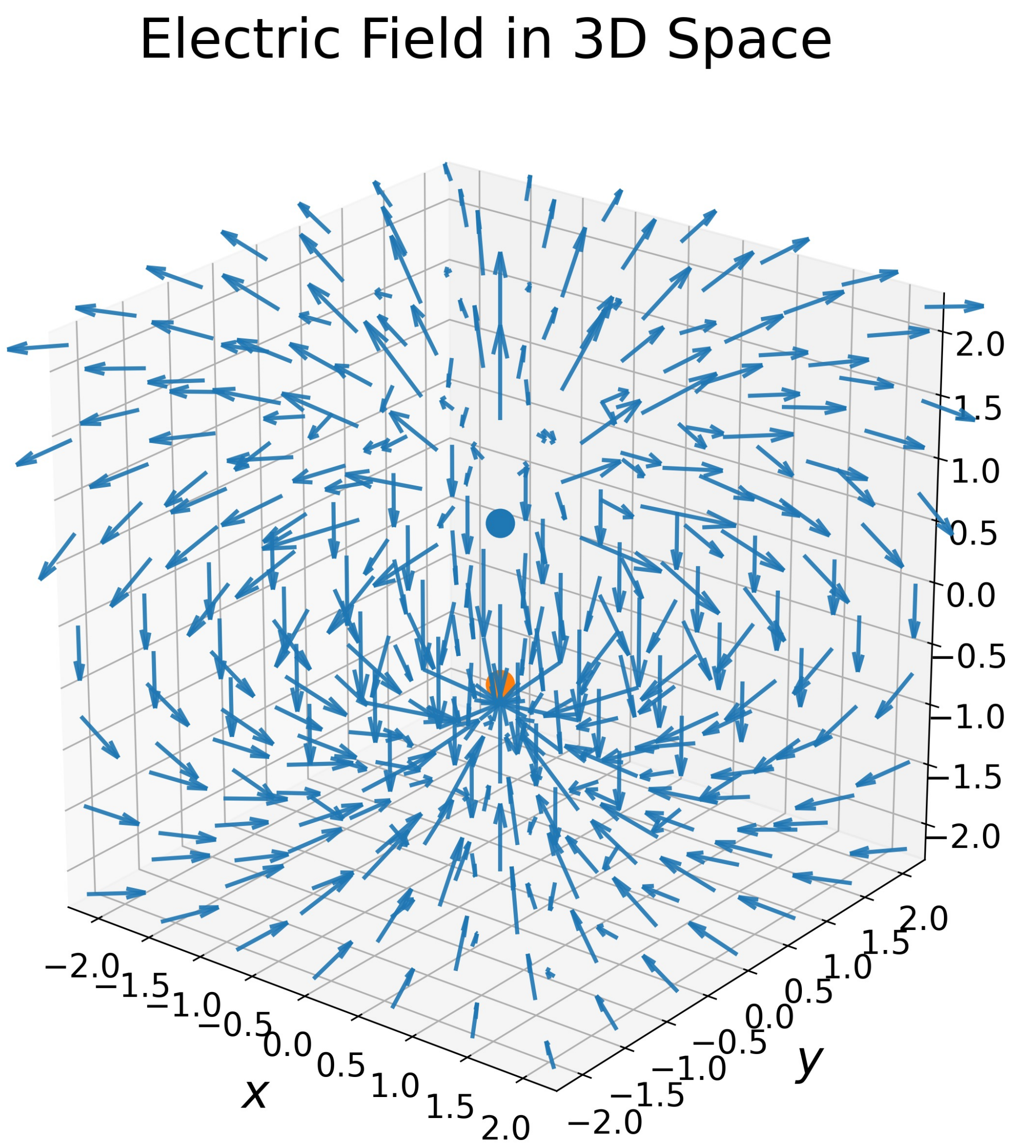
Examples 2

- Vector fields:

$$\mathbf{E} : \mathbb{R}^3 \to \mathbb{R}^3$$

$$\mathbf{E}(\mathbf{r}) \quad \text{is electric field at } \mathbf{r}$$

- 3-vector fields appear in electromagnetism, fluid mechanics, ...



Partial derivatives

Extremely useful in optimization

What is the «slope» of a multivariable function?

- In ordinary calculus:

$$f'(x) = \frac{df}{dx}(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

- But in 2D there are infinitely many directions to go.
- For example, the coordinate directions:

$$\frac{\partial f}{\partial x_1}(\mathbf{x}) \quad \frac{\partial f}{\partial x_2}(\mathbf{x})$$

Partial derivatives are ordinary derivatives of slices

- Freeze a value for x_1 and define a **slice:**

$$g(x_2) = f(x_1, x_2)$$

- The partial derivative is the **ordinary derivative** of the slice function:

$$\frac{\partial f}{\partial x_2}(x_1, x_2) = \frac{dg}{dx_2}(x_2)$$

- Gradient:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} \quad \text{a vector!}$$

Quick demo

- `lecture-4-hhh.ipynb`

Directional derivatives

- It is almost **trivial** to see that the directions 1 and 2 are arbitrary.
- Why not choose another direction $\mathbf{u}$?
- This leads to the **directional derivative**

$\mathbf{x} + t\mathbf{u}$ is a line through $\mathbf{x}$

$$D_{\mathbf{u}} f(\mathbf{x}) = \lim_{h \to 0} \frac{f(\mathbf{x} + h\mathbf{u}) - f(\mathbf{x})}{h}.$$

Derivative of a slice along **u**

$$D_{\mathbf{u}} f(\mathbf{x}) = \nabla f(\mathbf{x}) \cdot \mathbf{u} \quad \text{by the chain rule}$$

Gradient contains all directional derivatives!

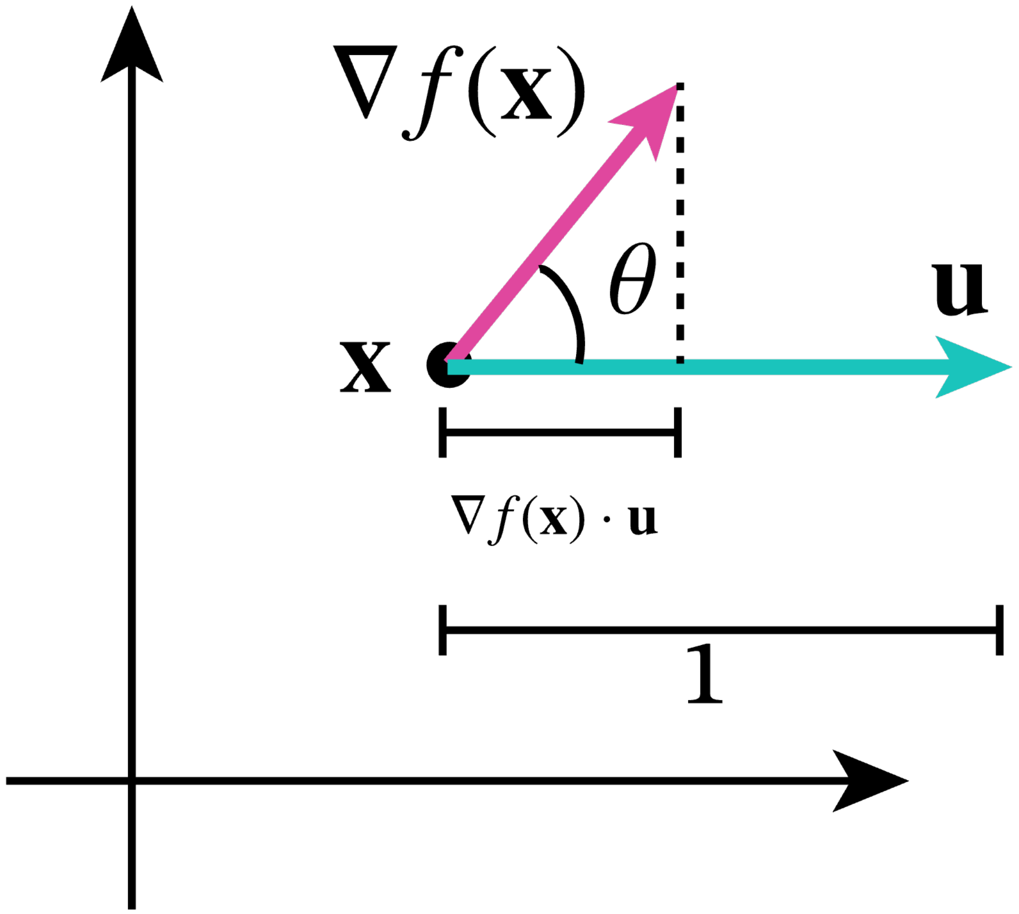
Meaning of the gradient

- Directional derivative is a dot product:

$$D_{\mathbf{u}} f = \nabla f \cdot \mathbf{u} = ||\nabla f|| \cos \theta$$

- The cosine is at max if angle is 0
- The gradient points in the **direction of steepest ascent**

- **Demo:** `lecture-4-gradient.ipynb`



Finding a local minimum: gradient flow

- A local minimum must have:

$$\nabla f(\mathbf{x}_*) = 0$$

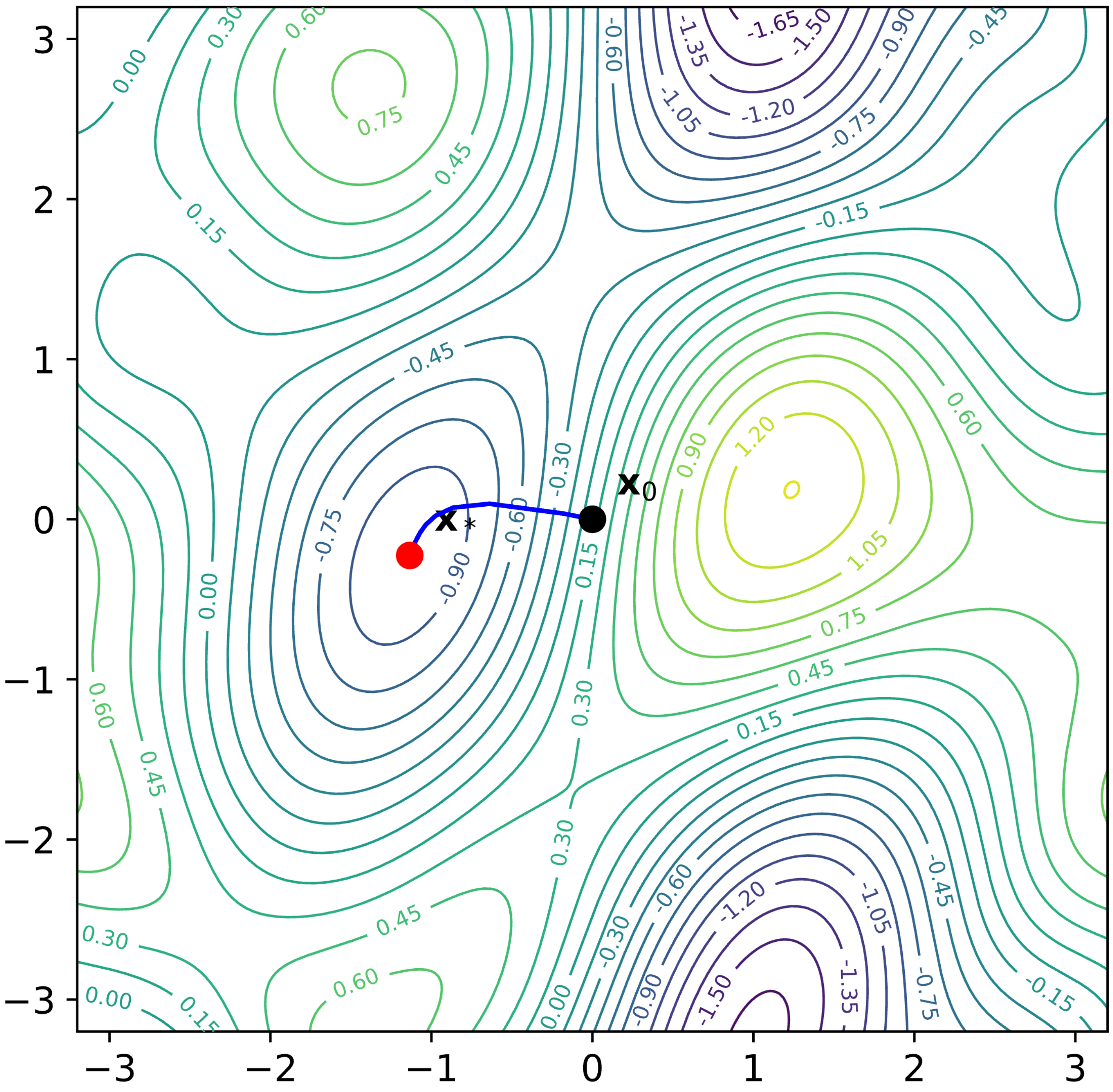
$$-\nabla f(\mathbf{x}) = \text{direction of } \textbf{steepest descent}$$

- If we always walk downhill, we must come to the bottom eventually!
- Define a **curve** as follows:

$$\mathbf{x}(0) = \mathbf{x}_0$$

$$\frac{d}{dt}\mathbf{x}(t) = -\nabla f(\mathbf{x}(t)) \quad \text{gradient flow}$$

- Starting point for **many algorithms:** geometry opt., machine learning ...



$\mathbf{x}_0$
$\mathbf{x}_*$

Differentiability and linear algebra

Looking back at calculus …

- The derivative had an important interpretation:

$$f(x+h) \approx f(x) + f'(x) \cdot h$$

- $f'(x)$ was the **best linear approximation** near x

- **What replaces this in vector calculus?**

Carl Gustaf Jacob Jacobi
(1804-1851)

Jacobi =
«Jacob's»

The Jacobian matrix

A vector of scalar functions

- We consdier now a **general function**

$$\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n \qquad \mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(x_1, \cdots, x_n) \\ \vdots \\ f_m(x_1, \cdots, x_n) \end{bmatrix}$$

- Each f_i now has a gradient. We obtain a **matrix of partial derivatives**

Jacobi's's matrix

$$J_{\mathbf{f}}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \qquad \text{i.e.,} \quad J_{\mathbf{f}}(\mathbf{x})_{ij} = \frac{\partial f_i}{\partial x_j}$$

From the Jacobian to linear approximation

- We consider a single component f_i
- Choose an origin and a direction in which to walk:

$$\mathbf{x} + t\mathbf{u} \text{ is a line through } \mathbf{x}$$

- Compute the **linear approximation** of the slice function:

Matrix-vector product!

$$\begin{aligned} f_i(\mathbf{x} + t\mathbf{u}) &\approx f_i(\mathbf{x}) + tD_{\mathbf{u}} f_i(\mathbf{x}) \\ &= f_i(\mathbf{x}) + t\nabla f_i(\mathbf{x}) \cdot \mathbf{u} \\ &= f_i(\mathbf{x}) + t\sum_{k=1}^{n} \frac{\partial f_i}{\partial x_k}(\mathbf{x})\, u_k = f_i(\mathbf{x}) + t\sum_{k=1}^{n} J_{ik}(\mathbf{x}) u_k \end{aligned}$$

Small changes are linear

- We obtain the vector equation:

$$\mathbf{f}(\mathbf{x} + \mathbf{h}) \approx \mathbf{f}(\mathbf{x}) + J_{\mathbf{f}}(\mathbf{x})\,\mathbf{h}$$

- A derivative in many dimensions is **linear transformation!**
- **h** is a small change in the input
- **The second term is the small change in the function – which is a linear transformation!**

- **Demo:** `lecture-4-parrot-transform.ipynb`

What is a differentiable function?

- It may be that the first-order Taylor approximation **fails**
- For example: Partial derivatives may not exist!

- We say that **f** is differentiable at **x** if the first-order Taylor expansion holds true!

$$\mathbf{f}(\mathbf{x}+\mathbf{h}) \approx \mathbf{f}(\mathbf{x}) + J_{\mathbf{f}}(\mathbf{x})\,\mathbf{h}$$

f is differentiable

Example of a non-differentiable function

$$f(x, y) = \frac{xy^2}{(x^2 + y^2)^{3/2}}$$

- Run `nondiff_function.py`

Function composition and the chain rule

- Recall the **chain rule** from calculus:
  - h is the **composition** of two functions:

$$h(x) = f(g(x)) \quad h'(x) = f'(g(x))g'(x) \qquad \leftarrow \text{chain rule}$$

- Multivariable chain rule:

$$\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^o \quad \mathbf{f} : \mathbb{R}^o \to \mathbb{R}^m \quad \mathbf{h}(\mathbf{x}) = \mathbf{f}(\mathbf{g}(\mathbf{x}))$$

$$J_{\mathbf{h}}(\mathbf{x}) = J_{\mathbf{f}}(\mathbf{g}(\mathbf{x}))\, J_{\mathbf{f}}(\mathbf{x})$$

One linear transformation following after another

A main component of neural networks

Under a mathematical microscope,
Vector Calculus is linear algebra.

The Hessian

The second derivative of a scalar function

Taylor expansion to one more order

- We focus again on **scalar functions**
- Jacobian = gradient in this case:

$$f(\mathbf{x} + \mathbf{h}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x}) \cdot \mathbf{h}$$

- Repeated differentiation of the slice function leads to **second order polynomial,**

$$f(\mathbf{x} + \mathbf{h}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x}) \cdot \mathbf{h} + \frac{1}{2}\mathbf{h}^T H_f(\mathbf{x})\mathbf{h}$$

- H_f is called the **Hessian matrix**

Hessian matrix

- The Hessian matrix is the second derivative of a scalar function:

$$f(\mathbf{x}+\mathbf{h}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x}) \cdot \mathbf{h} + \frac{1}{2}\mathbf{h}^T H_f(\mathbf{x})\mathbf{h}$$

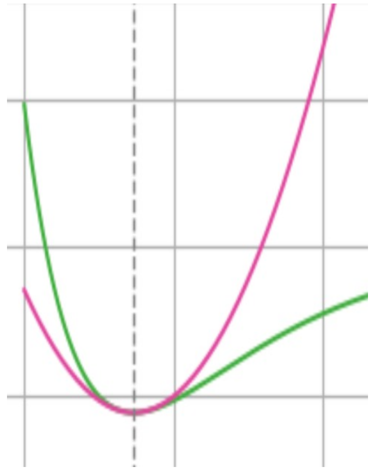
$$H_f(\mathbf{x})_{ij} = \frac{\partial^2 f(\mathbf{x})}{\partial x_i \partial x_j} \qquad \leftarrow \text{mixed partial derivatives}$$

- We say that *f* is **twice differentiable** if the Taylor approximation holds
- In that case **the Hessian is symmetric**

Maxima, minima, saddle points

The Hessian encodes curvature

$f''(x_*) > 0$



- Recall from calculus, near a **stationary point:**

$$f'(x_*) = 0 \qquad f(x_* + h) \approx f(x_*) + \frac{1}{2} h^2 f''(x_*)$$

- Multivariable case, **stationary point**:

$$\nabla f(\mathbf{x}_*) = 0 \qquad f(\mathbf{x}_* + \mathbf{h}) \approx f(\mathbf{x}_*) + \frac{1}{2} \mathbf{h}^T H_f(\mathbf{x}_*) \mathbf{h}$$

Encodes the nature of the stationary point

Maxima, minima, saddle points 1

- How does the second order term behave?

$$\nabla f(\mathbf{x}_*) = 0 \qquad f(\mathbf{x}_* + \mathbf{h}) \approx f(\mathbf{x}_*) + \frac{1}{2}\mathbf{h}^T H \mathbf{h}$$

- Since H is symmetric we have **orthonormal basis of eigenvectors:**

$$H\mathbf{x}_k = \lambda_k \mathbf{x}_k \qquad \mathbf{h} = \sum_{k=1}^{n} h_k \mathbf{x}_k$$

Exapansion of the small change in eigendirections

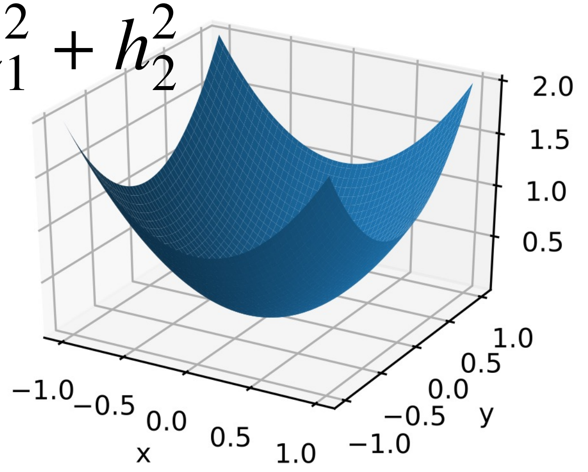
$$f(\mathbf{x}_* + \mathbf{h}) \approx f(\mathbf{x}_*) + \frac{1}{2}\sum_{k=1}^{n} h_k^2 \lambda_k$$

Maxima, minima, saddle points 2

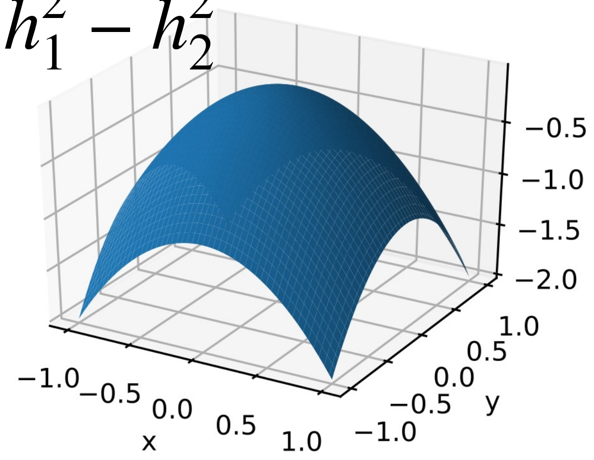
$$f(\mathbf{x}_* + \mathbf{h}) \approx f(\mathbf{x}_*) + \frac{1}{2}\sum_{k=1}^{n} h_k^2 \lambda_k$$

- All eigenvalues positive $\Rightarrow$ local minimum
- All eigenvalues negative $\Rightarrow$ local maxium
- Positive and negative eigenvalues $\Rightarrow$ saddle point

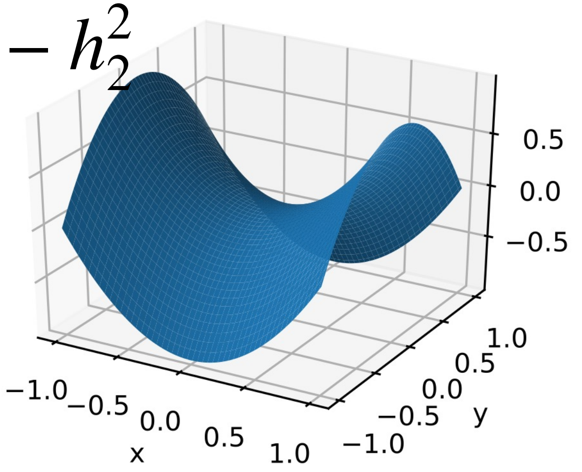
$h_1^2 + h_2^2$

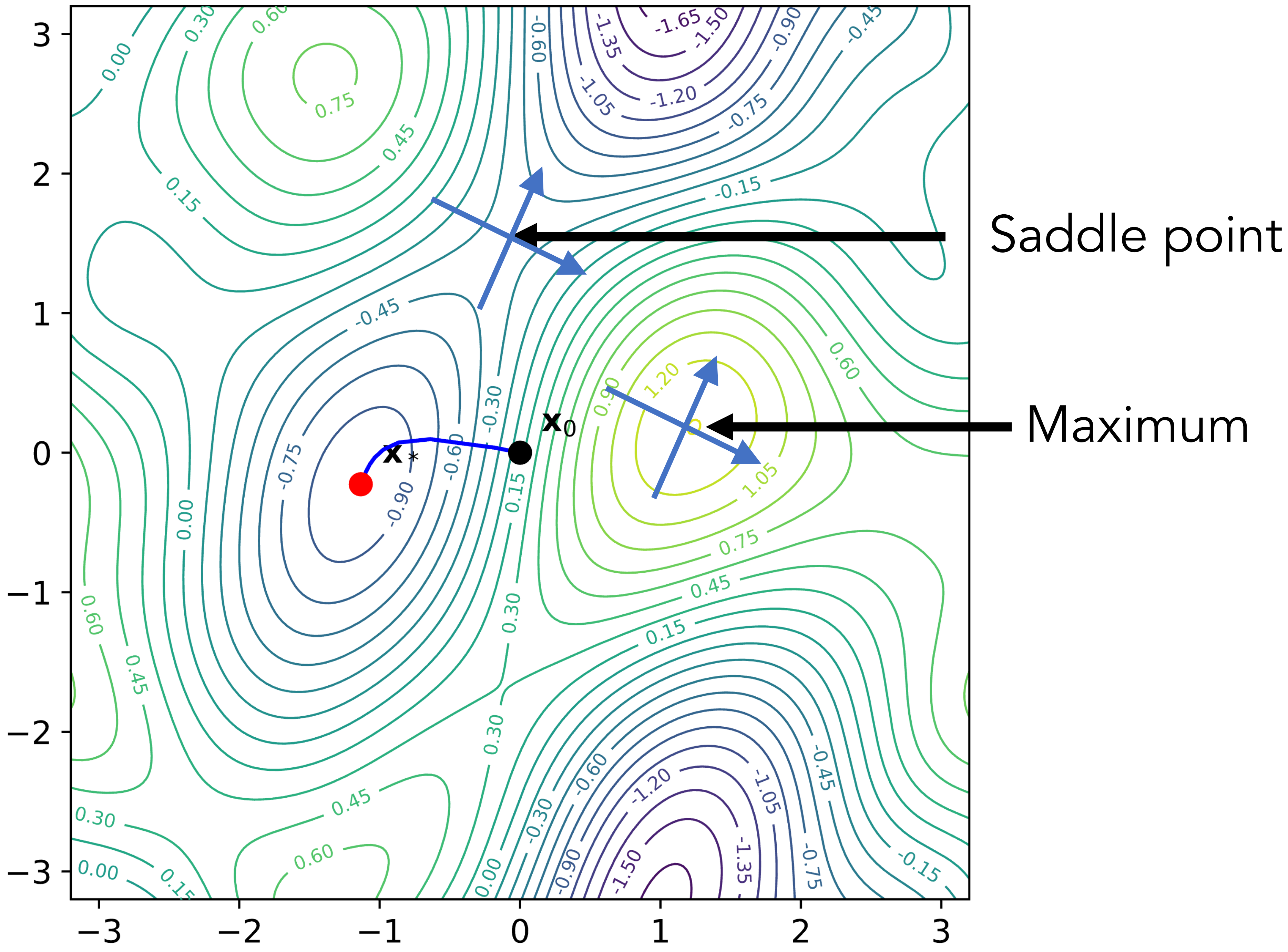


$-h_1^2 - h_2^2$



$h_1^2 - h_2^2$





Saddle point
Maximum
$\mathbf{x}_0$
$\mathbf{x}_*$

End of lecture 4

- Other topics of vector calculus:
  - Constrained optimization and Lagrange multipliers
  - Integration and the theorems of Gauss and Stoke