Numerical methods
Bachelor in Physics

Practicum 12: Gradient descent

Contact: jeroen.mulkers@uantwerpen.be U.305
misa.andelkovic@uantwerpen.be U.209

Many quantitative problems are optimization problems. In these problems, a certain value is to be minimized or maximized by varying input variables. Some examples of optimization problems are: profit maximization, strength optimization of bridge structures, optimization of traffic flow in infrastructure design, the principle of least action in Physics, and the training of an artificial neural network.

In this worksheet, we will study the *gradient descent* method. This is a simple method to minimize a function with continuous variables. First, we will study how this method can be used to determine the minimum of a convex quadratic function. Next, we will compute the profile of a magnetic domain wall by minimizing the magnetic free energy. We end this worksheet by studying the bending of sunbeams in the atmosphere starting from Fermat's principle.

1 Gradient descent

Consider a function $f(\mathbf{x})$ which depends on one or multiple variables (here placed in a single vector $\mathbf{x}$). To find the minimum of this function, we choose an arbitrary starting point $\mathbf{x}_0$, and walk downward along the gradient ∇f in a stepwise fashion until we reach the minimum:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \gamma \nabla f(\mathbf{x}_n). \tag{1}$$

You can easily check that $f(\mathbf{x}_{n+1}) < f(\mathbf{x}_n)$ if the step-size factor γ is chosen small enough. Since the gradient is zero at function extrema, one can argue that the gradient descent has converged to a local minimum if the gradient is close to zero: $\|\nabla f(\mathbf{x}_n)\| < TOL$, with TOL a predefined tolerance.

As an example, let's consider the quadratic function $f(x, y) = x^2 + 2y^2 - xy - 7x$ which is convex and has a single minimum in (4,1). The Matlab script *quadratic.m* visualizes the gradient descent method applied on this function. Run this script and study the implementation of the gradient descent method in *descent.m*. Check what happens if you vary γ between 0.1 and 0.5.

Task 1 Implement the Barzilai-Borwein step in *descent.m* and check if this improves the speed of descent.

2 Magnetic domain wall

Neighbouring magnetic moments in ferromagnetic materials have the tendency to align to each other. The interaction between neighbouring magnetic moments $\mathbf{S}^{(i)}$ and $\mathbf{S}^{(j)}$ can be modelled by the following classical Heisenberg Hamiltonian:

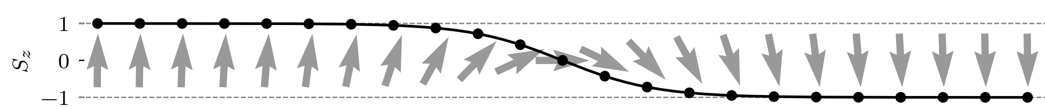
$$E_{ij}^{\text{Exchange}} = -J\mathbf{S}^{(i)} \cdot \mathbf{S}^{(j)}, \tag{2}$$

with J the exchange interaction strength. Note that the energy is minimal if both magnetic moments point in the same direction.

Some ferromagnetic films have a perpendicular magnetic anisotropy (PMA). Due to this interaction, magnetic moments have the tendency to point out of plane. This interaction can be modelled as follows:

$$E_i^{\text{PMA}} = -K\left(S_z^{(i)}\right)^2 \tag{3}$$

with anisotropy strength K. Note that the energy is minimal if the magnetic moment is perpendicular to the film (parallel to the z axis). In these so-called PMA materials, the magnetization configuration of the low lying energy states consist out of magnetic domains with *up* or *down* magnetization. These domains are outlined by magnetic domain walls in which the magnetization varies smoothly from *up* to *down* or vice versa (see figure below). In what follows, we will study the profile of such a domain wall.



Let's consider a ferromagnetic chain of N magnetic moments $\{\mathbf{S}^{(i)} : i = 1 \dots N\}$ with PMA. The Heisenberg-like Hamiltonian of this system is given by:

$$E = -J\sum_{i=1}^{N-1} \mathbf{S}^{(i)} \cdot \mathbf{S}^{(i+1)} - K\sum_{i=1}^{N}\left(S_z^{(i)}\right)^2. \tag{4}$$

To make the problem easier, let's assume that $\|\mathbf{S}^{(i)}\| = 1$ and that the magnetic moments can only rotate in the x, z plane. Under these assumptions, it is possible to describe the magnetic moments by polar angles θ_i instead of a 3D vector: $\mathbf{S}^{(i)} = (\sin\theta_i, 0, \cos\theta_i)$. The total magnetic energy is now given by

$$E = -J\sum_{i=1}^{N-1} \cos(\theta_i - \theta_{i+1}) - K\sum_{i=1}^{N} \cos^2\theta_i. \tag{5}$$

Task 2

(a) Create a function which returns the energy E (a single scalar) and the energy gradient $\partial E/\partial\theta_i$ (a vector) for a given magnetization configuration θ_i (a vector of polar angles).

(b) Construct a chain with $N = 50$ magnetic moments which vary smoothly from up (left boundary) to down (right boundary): $\theta_i = \pi(i-1)/(N-1)$. This configuration is a stretched domain wall which minimizes the exchange energy (but not the anisotropy energy). Plot the profile of the z component of the domain wall $S_z^{(i)} = \cos\theta_i$. .

(c) Now, let's freeze the direction of the magnetic moments at the two boundaries (Dirichlet boundary conditions) by setting the last and first element of the energy gradient in the energy function to zero (why?).

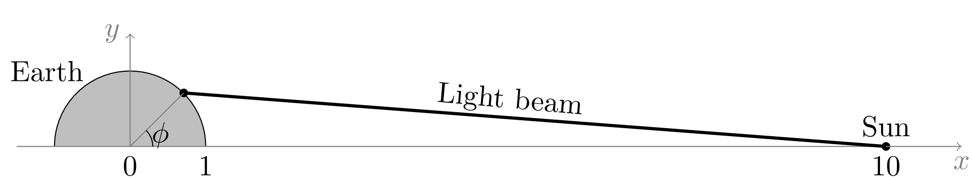
(d) Use *descent.m* to minimize the energy of a chain with $J = 1$, $K = 0.1$, and $N = 50$ magnetic moments. Use the magnetization configuration constructed in (b) as initial configuration.

(e) Vary K to study the effect of the perpendicular magnetic anisotropy on the domain wall profile. Plot the profiles in a single figure.

3 Fermat's principle

Fermat's principle is the principle that the path taken between two points by a ray of light is the path that can be traversed in the least time. Determining the trajectory of a light beam is thus essentially a minimization problem in which the total time is minimized by varying the path.

One could argue that light from the sun travels in a straight line to the earth at the speed of light c, as shown in the figure below (yes, I know the sun is actually much larger than the Earth and much further away, but bear with me). This assumption, however, is not entirely true. Light travels slightly slower in the Earth's atmosphere than in space. In this exercise, we will see how this leads to bending of sunbeams in the atmosphere.



The total travel time T to go from one point to another point along a path $\mathcal{L}$ is given by the line integral

$$T = \int_{\mathcal{L}} \frac{1}{v} \mathrm{d}l, \tag{6}$$

with v, the velocity along the path, and $\mathrm{d}l$, an infinitesimal part of the path. One can approximate this numerically by splitting the path in straight segments in which light travels at the same speed:

$$T \approx \sum_{i=1}^{N-1} \frac{\Delta l_i}{v_i} \tag{7}$$

with the length of the ith segment (in 2D)

$$\Delta l_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{8}$$

and the speed in the ith segment

$$v_i = \frac{v(x_i, y_i) + v(x_{i+1}, y_{i+1})}{2}. \tag{9}$$

which is equal to c far away from Earth, and which becomes smaller than c when approaching the Earth's surface.

According to the figure, we fix the first point somewhere at the surface of the Earth, let's say $(x_1, y_1) = (\cos\phi, \sin\phi)$ and the last point at the sun $(x_N, y_N) = (10, 0)$. Furthermore, we will use fixed equidistant x positions. Now we can vary $\{y_i : i = 2 \ldots N-1\}$ to minimize the total time T.

Task 3

(a) Run and study the script *sunbeam.m* which computes the bending of a sunbeam in the atmosphere.

(b) Place the sun on the x axis infinitely far away from the Earth. Hint: use the Neumann boundary condition $\partial y/\partial x = 0$ at $x = x_N$ instead of a Dirichlet boundary condition (why?).

(c) Add the viewing direction on the Earth to see the sun. Use a first-order forward difference based on the first two points.

(d) Same as (c) but use a second-order forward difference based on the first three points.