KU LEUVEN

SUMMER OF SCIENCE

MAGNETISM AND SUPERCONDUCTIVITY

Summer of Science 2025

Summer of Science @ KU Leuven
28 July - 1 August 2025

Contents

Chapter I

Introduction to Magnetism

1 The fundamental forces of nature

Everything that you see (and don't see) in nature can be categorised into four fundamental forces. Their relative strength and range can be summarized as follows:

Table I.1: Fundamental forces of nature

	Gravity	Electromagnetism	Strong nuclear	Weak nuclear
Relative strength	Very weak	Moderate	Very strong	Weak
Range of force	Long	Long	Very short	Very short

By looking at Table I.1, you see that **gravity** is very weak compared to the other forces. That's why this force, that influences mass, is only significant for really heavy objects, like planets or stars. It keeps most matter close to or at Earth's surface and it keeps the Earth in orbit around our Sun. Apart from that, and perhaps the tides, it does not influence your life too much.

The two nuclear forces, strong and weak, is not very visible in your life either. Even though the **strong nuclear force** is very strong, its range is very short (subatomic scales). The same is true for its weaker brother, the **weak nuclear force**. The strong nuclear force prevents atomic nuclei from falling apart. The weak nuclear force is the origin of decays of neutrons into a protons and an electrons (β-radiation).

Since **electromagnetism** is moderate in strength (much stronger than gravity) and also has a long range (many, many times longer than the nuclear forces), it is the force that influences (almost) everything you see and experience in daily life. We give some examples of physical phenomena that are mitigated by electromagnetism:

- Visible light
- The full electromagnetic spectrum
- Electricity
- Magnetism
- The attraction of electrons to the nucleus in an atom
- Forces like normal force, friction, ... are interactions between the electrons in the different objects
- ...

2 Electric and magnetic fields

It is possible to write down mathematical expressions for all the interactions in Table I.1 in terms of forces. However, over time scientist found that electricity and magnetism are much easier described by **electric and magnetic fields**. But what is a "field"?

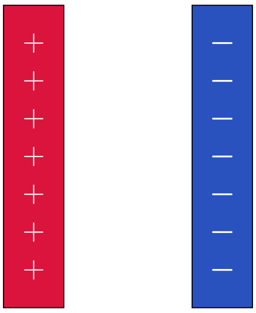
Well the correct terminology would be "a field of force". So the direction and magnitude of a field is the level of force experienced by another object upon which it acts. The best thing about fields is that they are easy to draw.

Exercise 1. *Draw the electric and magnetic field lines for the following systems. Also pay attention to the density of field lines.*

1. pos. charge

2. neg. charge

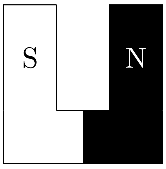
3. Two plates



Drawing magnetic field lines

1. Bar magnet

2. Horseshoe magnet



3. Two bar magnets

How can we interpret these fields? The arrows show the direction of the force and the spacing of the lines indicate the strength of the field in that point (more lines per area = more electromagnetic force exerted at that point). In the figures you can also see that why unlike poles in magnets attract: field lines flowing from north to south and are closely packed in between the poles. So the magnetic field in between the poles exerts a magnetic force upon objects (or even the both bar magnets) that pushes the object (or both bar magnets) from north to south.

3 Maxwell's equations

There are mathematical equations that explain how these electric and magnetic fields are created by charges and currents. These are called **Maxwell's equations**, after 19th century physicist James Clerk Maxwell. For now, you don't have to understand the mathematical symbols, we will describe them intuitively. Later in the course, we will explore the mathematical side of these equations in more detail. There are four equations:

3.1 Gauss' law

$$\vec{\nabla} \cdot \vec{E} = -\frac{\rho}{\epsilon_0} \tag{3.1}$$

This describes how electric fields are created: a positive point charge (e.g. a proton) radiates an electric field **E** outwards A negative point charge (e.g. electron) radiates an electric field **E** inwards. This explains why the same charges repel each other and opposite charges attract.

3.2 Faraday's law

$$\vec{\nabla} \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \tag{3.2}$$

This equations explains how a changing magnetic field **B** induces an electric current. This is the principle behind a magnetic induction generator (dynamo).

3.3 Ampere-Maxwell's law

$$\vec{\nabla} \times \vec{B} = \mu_0 (\vec{J} + \frac{1}{c^2} \frac{\partial \vec{E}}{\partial t}) \tag{3.3}$$

An electric current **J** or changing electric flux $\frac{\partial \mathbf{E}}{\partial t}$ produces a circulating magnetic field.

Exercise 2. *Eddy currents: with equation 3.2 and 3.3, how can you explain why a falling magnet in a copper tube is slowed down?*

....

....

3.4 Magnetic monopoles do not exist

$$\vec{\nabla} \cdot \vec{B} = 0 \tag{3.4}$$

The final equation looks very similar to equation (3.2). Indeed, this equation describes magnetic poles: the magnetic fields **B** radiates from magnetic north poles towards south poles. But, this equation is set to 0. Consequently, no magnetic fields is allowed to radiate only outwards or only inwards. This means that objects that are just a north or only a south pole (magnetic monopoles) do not exist! There is no magnetic equivalent of point charges. North and South always must come together and cannot exist separately. The combination of North and South pole is called a dipole. All magnets must be dipoles. Even when you break a magnet in two, you will still get two smaller magnets with both poles.

Let us now take a closer look at magnets: what are they, really, and how can we describe them scientifically?

4 Classical description of magnets

From the Ampere-Maxwell equation (3.3), we saw that moving electric charges can generate a magnetic field. This is why, before the early 1900's, physicists believed that the moving electrons in magnets generated the magnetic fields of that material. In fact, they believed that the electrons were orbiting the nucleus of the atom like planets in our solar system. So, it looked very obvious to them that this rotation of an electron generated a magnetic dipole. Since materials consisted of trillions of electrons, this could lead to macroscopic magnetic fields.

5 Quantum mechanical description of magnets

5.1 What is quantum mechanics?

Quantum mechanics is a more fundamental theory than classical physics. It can also explain submicroscopic physics (at the scale of the atom and even subatomic). An important discovery in quantum mechanics, is that particles have both particle and wave characteristics. An important consequence of this is that electrons in an atom do not have an exact location around the nucleus.

Why is it called "**quantum** mechanics"? It starts from the principles that several physical quantities, like charge, energy, angular momentum, ..., are **quantized**!

Example: quantization of charge There is a fundamental charge $q = 1.6 \cdot 10^{-19} C$. A proton has a positive charge q and an electron $-q$. Every neutral atom (equal amounts of protons and electrons) has a total charge of 0. An atom missing two electrons, has a total charge of $2 \cdot q$. Since everything is made out of a certain number of protons and electrons, everything will have a total charge that is a sum of q or $-q$.

This was an easy example that was already used in classical physics. But this idea is now used also for quantities that in classical physics could have any value possible: for example angular momentum has a quantum of $\hbar = 1.056... \cdot 10^{-34} (J \cdot s)$/rad.

5.2 Magnetic spin

Unlike in the classical model, electrons do not orbit the nucleus like planets in a solar system. In fact, due to their wave-like character, they are never in a single location in the atom. You have to describe their location as a probability function. This is what we call **orbitals**. So, where does magnetism in materials come from, if it is not orbiting electrons?

In 1925, George Uhlenbeck and Samuel Goudsmit discover that just like charge, magnetic moments are quantized! This means that every electron or proton is its own **tiny magnet** or **spin**[1]. Why do protons and electrons have a tiny magnetic moment? That's just the way it is. A similar question would be: why does an electron have a negative charge? That is just how nature works, and we are simply describing it.

Importantly, the electron's tiny magnet is 1840 times bigger than the proton's tiny magnet. As a consequence, the magnetism of materials will be entirely determined by the magnetic moments and configuration of its electrons.

---

[1] The word spin is an analogy to a spinning ball. A charged ball can spin clockwise or counter clockwise, which in the classical picture generates a magnetic field pointing down and pointing up, respectively. However, the electrons and protons are not actually spinning, and are not particles like a ball. So this is the worst analogy every.

6 Magnetic materials

The next question we can ask ourselves is: “If magnetism comes from the magnetic moment from electrons, and all materials are made of electrons. Why are not all materials magnetic?” Well, they all are magnetically actually! There are just different types of magnetic behavior.

Exercise 3.a) *Solenoid with soft iron core. Describe the effect you see when the following materials are put in the solenoid:*

1. Plexiglass:

   ....

2. Aluminium:

   ....

3. Copper:

   ....

4. Nickel:

   ....

5. Iron:

   ....

Exercise 3.b) *Magnetic materials in two solenoids. The first solenoid will or will not magnetise the material you will put in its core. The magnetised material will induce a current in the second solenoid. This current/voltage can be read out. For each material how does the current in the second solenoid change?*

1. Plexiglass:

   ....

2. PVC:

   ....

3. copper:

   ....

4. Aluminium:

   ....

5. Iron:

   ....

Exercise 3.c) *Magnetic properties of water. Fill a test tube with distilled water and let it float on the mini raft. Make sure it is stationary, and carefully hold a big magnet very close without touching it. What happens?*

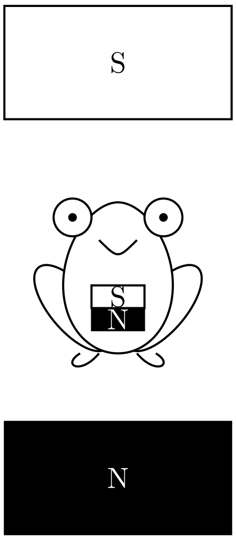
. . . .

. . . .

6.1 Diamagnetism

Diamagnetism is a property of all materials. An external magnetic field induces an opposite magnetisation, causing **a repulsive force**. However, it is very weak, that's why we need very strong magnetic fields to see it. If it is the only contribution in a material in a magnetic field, we call this material **diamagnetic**[2].

Since diamagnetism results in a repulsive force, diamagnetic objects in a strong enough vertical magnetic field (stronger than the gravitational force on the object) will start to levitate. In the video of the levitating live frog, the frog was around 1.5 cm and the applied field was about 16 Tesla. To put that in perspective, that is around 16,000 times stronger than the average fridge magnet, about 320,000 times stronger than Earth's magnetic field and 10 times stronger than an average MRI magnet.



Diamagnetism arises purely due to quantum mechanical effects and the theoretical derivation is out of the scope of this course. However, a classical model[3] can give the same prediction (but overestimates the magnitude of diamagnetism for many materials). In the classical theory of diamagnetism, we model electrons in the atom as current loops with effectively zero resistance. Bringing this material in an large external field will cause the electrons of the material to change course. The altered motion of the electrons will result in a small magnetisation that opposes the external magnetic field. This is very weak, but it will cause the material to be repelled from the external magnet.

---

[2] 'dia-' is Greek for *through* or *across*

[3] This model is called Larmor diamagnetism and we will do a simple mathematical derivation in section 8.2

Interestingly, this model of electrons moving in zero resistance loops is very similar to the (semi-) classical model for superconductors. In fact, superconductors are essentially perfect diamagnets and can repel all the applied magnetic field.

6.2 Paramagnetism

On top of having a diamagnetic response, some materials **align their magnetisation along the applied field direction**. These are materials are called **paramagnetic**. This effect is most of the time many times higher than diamagnetism, so you will usually not notice the diamagnetic response of a paramagnetic material.

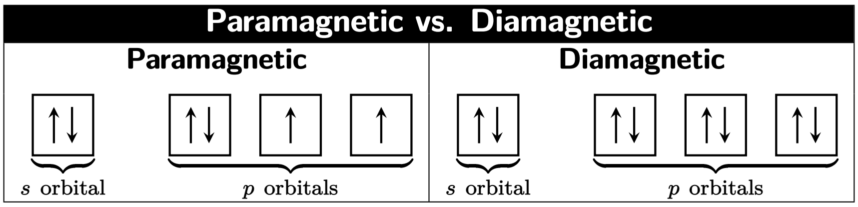
Exercise 4. *Looking back at materials you described in exercise 3, can you tell which materials have a diamagnetic and which have paramagnetic behaviour?*

....

....

....

When is a material paramagnetic and when is it only diamagnetic? We saw that the total magnetic moment of a material arises form the sum of tiny magnetic moments of the electrons. However, if two electrons occupy the same orbital shell, they will have an opposite tiny magnetic moment direction (one spin-up and one spin-down according to the Pauli exclusion principle). The net magnetic moment of these paired electrons will be zero. So the only net magnetic moment will arise from unpaired electrons! Hund's rules [4] tells us that if we fill orbital by first filling the shells with spin-up electrons and afterwards fill them with extra spin-down electrons. If we ran out of electrons to fill the orbital before all shells are filled, we end up with unpaired electrons in some shells. If this happens, we have a net magnetic moment for the atom.



To conclude, paramagnetic materials are are mostly made of atoms with partially filled outer orbitals. The total magnetic moment is the sum of all the net magnetic moments of all atoms. The resulting magnetisation will align with the external magnetic field to minimise energy.

6.3 Ferromagnetism

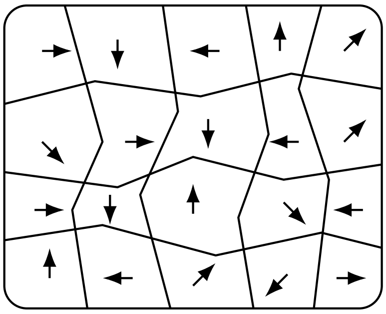
Ferromagnets are a special type of paramagnets. Like paramagnetic materials, they magnetise along the direction of the external magnetic field. Yet, when the external magnetic field is removed afterwards, the material **can remain magnetised** without the external field. This is the typical permanent magnet like the horse shoe or fridge magnet. The name "ferro" is derived from the Latin word for iron ("Ferrum"), which is indeed a ferromagnet. Other typical ferromagnetic materials are nickel, cobalt and neodymium.

---

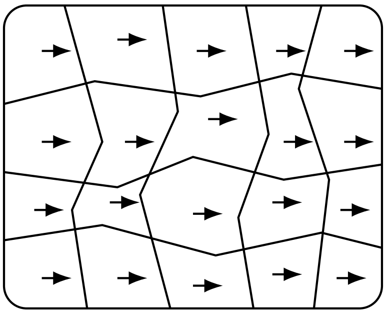
[4] For more information see section 7

Question: What makes a ferromagnet keep it magnetisation outside an external field?
The electrons should be able to tell each other which direction they should align their magnetic moment towards without the magnetic field being present. This is done by an interaction that we call **exchange interaction**. For a positive exchange interaction term, the system will minimise its energy if the magnetic spins are parallel to each other and the material will become a **ferromagnet**. How this interaction can minimise the energy and how one can derive the interaction term, we will see in section 9.

Question: Why can you also find unmagnetised iron and how can it get magnetised?
In un- or demagnetised ferromagnets (meaning that it does not have a net magnetisation) the exchange interaction still exists. However, it only forces the spins of a small regions of atoms to align[5]. This small region is called **a magnetic domain** or **a Weiss domain**. A neighbouring magnetic domain doesn't need to have the same direction. If all these magnetic domains are randomly orientated relatively to each other, the total net magnetisation of the material is zero.



Unmagnetized



Magnetized

However, after we apply a large external magnetic field, most domains will orient themselves along this field. When the external field is removed, the exchange interaction will maintain the orientation of most domains. This results in a net magnetisation in absence of external field!

Enough heat can demagnetise the ferromagnet. How can this be explained? Well, temperature is also a form of energy and it makes the magnetic moments of the domains vibrate randomly. For high enough temperature the vibrational energy because of temperature will be higher than the exchange interaction energy, and the spins do not tend to align with each other anymore. At this temperature, the ferromagnet acts as a paramagnet. The temperature when this phase transition happens is called the Curie[6] temperature.

---

[5] Assuming that we are under the Curie temperature of the material, see section 9.3 for more information

[6] Called after Pierre Curie, husband of Marie Curie.

Chapter II

Theory of magnetism

This chapter will elaborate on more theoretical aspects of magnetism. I think a small warning is in order: *this stuff is quite hard!* Now don't let that deter you, but don't be too hard on yourself if you get confused. I still get confused about it too! Do not hesitate to reach out during or after the lectures if you have any questions or unresolved confusion.

Now, let's see. The only logical place to start is with electromagnetism, as it is regularly framed: with *Maxwell's equations*;

7 Magnetism in equations

The equations describing electromagnetism involve only two players: the **electric and magnetic fields**, $\vec{E}(\vec{x})$ and $\vec{B}(\vec{x})$. These are *vector fields*, meaning they have a vector value at each point in space. The behavior of $\vec{E}$ and $\vec{B}$ is fully determined by the **Maxwell equations**:

$$\begin{aligned}
\textbf{1.}\quad \vec{\nabla}\cdot\vec{E} &= \frac{\rho}{\varepsilon_0}\\
\textbf{2.}\quad \vec{\nabla}\times\vec{E} &= -\frac{\partial\vec{B}}{\partial t}\\
\textbf{3.}\quad \vec{\nabla}\cdot\vec{B} &= 0\\
\textbf{4.}\quad \vec{\nabla}\times\vec{B} &= \mu_0\vec{J}+\mu_0\varepsilon_0\frac{\partial\vec{E}}{\partial t}
\end{aligned}$$

Each of these equations has an important meaning:

1. The only sources/sinks for **electric** field lines are positive and negative charges.
2. A varying magnetic field generates an electric field.
3. There are no sources/sinks for magnetic fields lines. All lines must close. **There is no magnetic charge.**
4. Moving charges or changing electric fields generate magnetic fields.

Out of these four, property **3.** is the most disturbing: there exist no North poles without South poles, **all magnets are dipoles**.

Now I mentioned something interesting: *a magnet.* You all know what magnets are, you have several of them in your house, and you probably even have some in your pocket right now (phones usually have magnets in them, e.g. in their speakers). You are maybe so used to them that you forgot how weird they really are, the ancient Greeks already knew about them[1], *what's new*? Well, I'm here to remind you how strange magnets really are, by letting you in on a few secrets.

7.1 Secret no. 1: The force between magnets

Weirdly, it is quite difficult to calculate the force between two magnets. There is no simple formula for it[2], and the force is highly dependent on the precise shape, material and mutual orientation of the magnets. Most university students couldn't even tell you *why* two magnets attract.

So let's study a simple example. Since all magnets are dipoles, let's construct the simplest model of a dipole we can imagine: a rectangular wire carrying a current I.

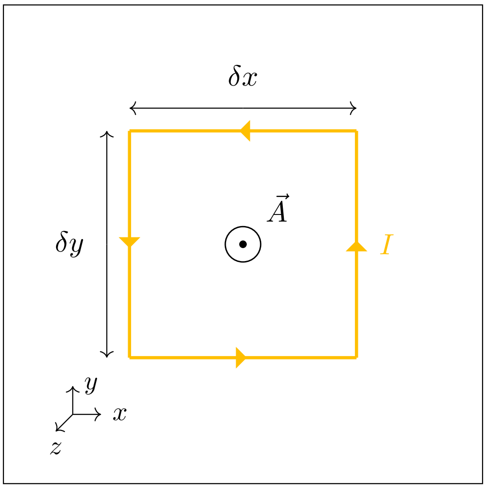


Figure 1: A simple model of a magnetic dipole: a rectangular loop carrying a current.

The magnetic dipole moment of this wire of this loop is:

$$\vec{m} = I\vec{A} = I\,\delta x\,\delta y\,\hat{z}. \tag{7.1}$$

The force on a straight piece of wire in an external magnetic field (the *Lorenz force*) is:

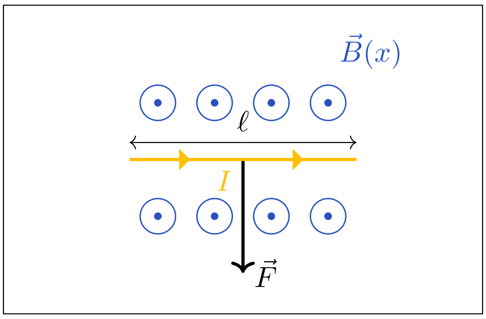
$$\vec{F} = I\,\vec{\ell} \times \vec{B} \tag{7.2}$$

where the vector product results in a force perpendicular to the wire, which schematically looks as follows:

---

[1] Magnets are named after the ancient Greek region of *Magnesia*, where magnetic stones were first found. Effectively they are called 'stones from Magnesia'; likewise, electricity was named by the ancuent Greeks after *amber*, a stone with strong triboelectric effect, which becomes charged when rubbed. Amber in greek is $\bar{\eta}\lambda\epsilon\kappa\tau\rho o\nu$ (elektron).

[2] If you don't believe this, ask your favorite local physicist. Chances are they will be embarrassed and start waving their hands as an answer.



Let us take

$$\vec{B}(\vec{x}) = \Big(B_x(\vec{x}),\, B_y(\vec{x}),\, B_z(\vec{x})\Big), \tag{7.3}$$

then we can compute the Lorentz force in all three directions:

$F_x =$ ....

$F_y =$ ....

$F_z =$ ....

Recall from the preliminaries the definition of a derivative:

$$\frac{df(x)}{dx} = \frac{f(x+\delta x) - f(x)}{\delta x}. \tag{7.4}$$

So we get

$F_x =$ ....

$F_y =$ ....

$F_z =$ ....

We conclude that in our setup the force is

$$\vec{F} = m\vec{\nabla}B_z. \tag{7.5}$$

More generally, when your loop does not lie in the $x-y$ plane, you get

$$\vec{F} = \vec{\nabla}(\vec{B}\cdot\vec{m}) \tag{7.6}$$

This was an easy calculation, but the takeaways are nonetheless deep and important:

- The force is proportional to the *gradient* of the magnetic field (rate of change in space).
- If the field is homogeneous, a current loop will only rotate to align with the magnetic field. It will not be repelled/attracted, cf. a compass and the Earth's magnetic field.
- When the field is induced by another magnet, the $\vec{B}$ field will *always* be a dipole field; dipole fields behave as $B \sim 1/r^3$. Then its derivative is proportional to the force so that

  $$F \propto \vec{\nabla}B \sim \frac{1}{r^4}. \tag{7.7}$$

  This means that it is very short ranged: the force drops to zero very quickly if you increase the distance r. (This is something you recognise from real life: magnets don't fly across the room towards the fridge!)

7.2 Secret no. 2: Magnets are complicated!

Now I have been talking about 'magnets' for a while, but what *is* a 'magnet'? You might say something like "a magnet is a material with magnetic properties". But that definition would be way too broad, because then *almost everything* would be a magnet! The objects we usually call magnets, the ones that stick to your refrigerator, are better referred to as **permanent magnets** or **ferromagnets**.

But then, what are these magnetic properties all materials have? In general, any material with magnetic properties can be categorized into one of three categories of magnetic behaviour. These are based on the *response* of the object to an external magnetic field.

Each material is made up out of atoms, which themselves are made up of a nucleus with electrons around it. These atoms have magnetic moments. The details of precisely which magnetic moment an atom has requires serious study of quantum mechanics, but the main point is this: magnetic moments arise when charged particles have **angular momentum**. In quantum physics, angular momentum comes in two forms:

- **orbital angular momentum**: arising from the "motion"[3] of the electron around the nucleus.
- **spin angular momentum**[4]: an intrinsic property of fundamental particles. Electrons have **spin 1/2**, as do protons. This means that each electron has an associate dipole moment, regardless of its motion.

What you should pick up from this, is that each material is *composed of small magnets*! In most objects, these small magnets are quite disordered, and their magnetic fields roughly cancel and average to zero. However, when you bring all these tiny magnets into a magnetic field, you expect them to respond to it in some way. Note that this is all *extremely complicated*, as quantum many-body physics is notoriously difficult.

What we *can* do here is to categorize these responses. To do this, we define a **response coefficient** χ:

$$\vec{M} = \chi \frac{\vec{B}}{\mu_0}, \tag{7.8}$$

which relates the **magnetization** of a material ($\vec{M}$) to the applied magnetic field ($\vec{B}$). The magnetization is the vector sum of all the dipole moments or the tiny magnets:

$$\vec{M} = \sum_i \vec{\mu}_i. \tag{7.9}$$

The different categories or types of magnetic behavior depend on the values of χ for that material:

- **Diamagnetism:** the system *repels* external magnetic fields, but outside a magnetic field it is not magnetized:

  $$\chi < 0 \qquad \vec{M} = 0. \tag{7.10}$$

  Examples of diamagnetic materials include copper, the levitating frog[5] (water), ...

---

[3] It is alright to *think* of an electron whizzing around the nucleus, but reality is more complicated. The orbital angular momentum is a property of the *orbital state* or *shell* the electron is in.

[4] The discovery of spin as an intrinsic magnetic moment, with the picture of the electron spinning around its own axis is usually attributed to George Uhlenbeck and Samuel Goudsmit in 1925. However, they were not the first to come up with the idea. Still a doctoral student, a few months earlier, Ralph Kronig had suggested the same idea, but the idea was rejected by Pauli, who influenced Kronig not to publish his idea.

[5] A frog was famously levitated in a magnetic field by physicist Andre Geim, which earned him the Ig-Nobel prize (for experiments which *can* not or *should* not be repeated) in 2000. Ten years later Geim also won the actual Nobel prize for discovering graphene, making him the first winner of both the Ig-Nobel prize and the Nobel Prize.

- **Paramagnetism:** the system *attracts* external magnetic fields, but outside a magnetic field it is not magnetized:
$$\chi > 0 \qquad \vec{M} = 0. \tag{7.11}$$
  Most metals with unpaired electrons are paramagnetic.
- **Ferromagnetism:** ferromagnets are paramagnets that can *keep a nonzero* $\vec{M}$ even in absence of an external magnetic field
$$\chi > 0 \qquad \vec{M} > 0. \tag{7.12}$$
  Examples include iron, nickel, ...

Whether any given atom behaves dia- or paramagnetically depends on the configuration of electrons in its orbit. This is something you might recognise from your chemistry classes: electrons orbit around atoms in **orbitals** or **shells**. Most atoms will have a bunch of completely filled or **closed shells**, and only a single partially filled shell, called the **valence shell**. The closed shells will have no net magnetic moment, so the magnetic properties come entirely from the valence shell.

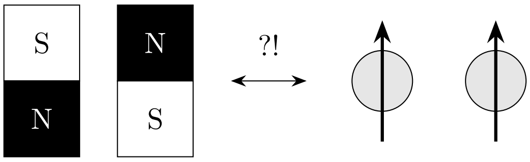
A note on chemistry: Chemistry was invented as a way to understand the quantum physics of atoms and molecules *without* understanding quantum physics. For this reason, there are a lot of 'rules' which give the correct answers, but with little to no explanation why. The answers usually lie in serious quantum mechanics. We will try to strike a balance here[6].

The order in which the orbitals fill up is determined by **Madelung's rule**, as illustrated in the table below:

n	He	Be	Mg	Ca	Sr	Ba	Ra
1	$1s^2$						
2	$2s^2$	$2p^6$					
3	$3s^2$	$3p^6$	$3d^{10}$				
4	$4s^2$	$4p^6$	$4d^{10}$	$4f^{14}$			
5	$5s^2$	$5p^6$	$5d^{10}$	$5f^{14}$			

There are also rules for how each orbital is filled, called **Hund's rules**:

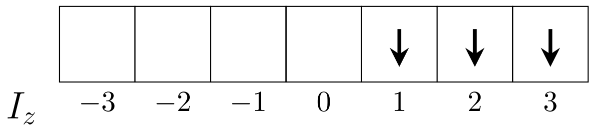
- **Hund's first rule:** electrons tend to align their spins. This is the opposite of what you would expect from ordinary small magnets. The reason for this is that the magnetic moment of an electron is tiny, and Coulomb interaction dominates.



---

[6] The mechanisms can usually be explained in terms of minimization of energy.

- **Hund's second rule:** electrons tend to *maximize* their total orbital angular momentum. This is again driven by the Coulomb interaction. Each 'position' inside an orbital has an associated orbital angular momentum, and the highest ones are filled up first. See the example of three electrons in an f-orbital in the picture below. The slot with $l = 3$ is filled up first, etc.



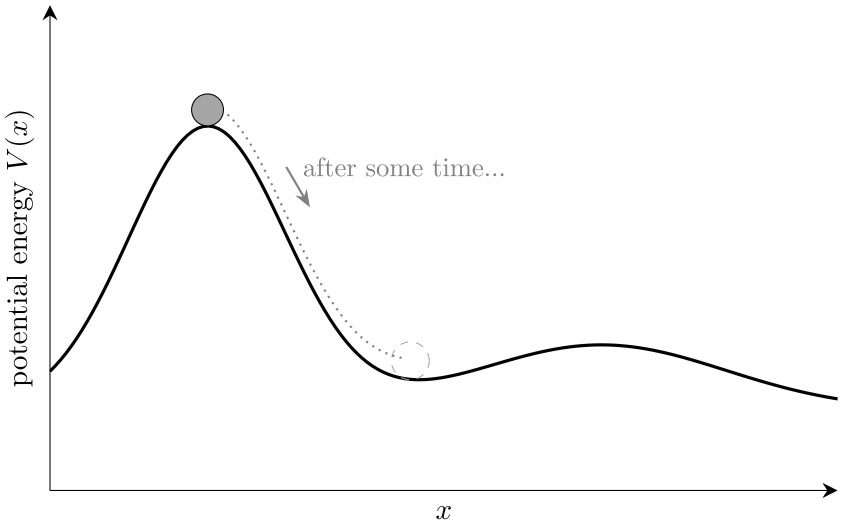
Now, you might be wondering why all the spins are down-spins. Well, ...

- **Hund's third rule:** Given that Hund's first and second rule are satisfied, the orbital and spin angular momentum will either align or anti-align, depending on whether the shell is more or less than half-full. In the case above, the spins anti-align with the orbital angular momentum because the orbital is less than half-full. The mechanism behind Hund's third rule is not the Coulomb force, but **spin-orbit coupling**. This is beyond the scope of this course, but deserves to be mentioned.

The main point of going over these rules is to point out that there are many effects and interactions to take into account when trying to figure out the electron configuration of a *single atom*. You can now appreciate that atoms can have wildly different configurations and properties, even if they are close in the periodic table. We will continue to study single atoms, but in a bit more detail.

8 Atoms in an external field

Briefly remind yourselves that most of physics is in some way an energy minimization. Think of how a ball rolls down a hill until it finds its resting point in a *local minimum*. What you should thus keep in mind is that the **ground state** of a system is almost always its minimum (reachable) energy configuration (e.g. at the bottom of the hill).



In quantum physics, the way we describe how a physical system behaves is by specifying a

Hamiltonian or **energy function**. This is the only thing you need to describe the system, nothing else.

A note on quantum mechanics: You will be hearing the word “quantum” a lot, but we will *not* teach you quantum mechanics. All you really need to know now, is that things are determined by energy functions, and that ground states minimize these energy functions. To give a little bit more detail: in quantum mechanics, things are described by **wave functions**, $|\psi\rangle$, and these behave according to the Schrödinger equation

$$i\frac{\partial}{\partial t}|\psi\rangle = \hat{E}\,|\psi\rangle\,, \tag{8.1}$$

where $\hat{E}$ is the energy function/operator

For a single, free particle, such as an electron, the energy function looks like this:

$$E_0(\vec{p},\vec{r}) = \frac{\vec{p}\cdot\vec{p}}{2m} + V(\vec{r}). \tag{8.2}$$

Here, the first term represents kinetic energy. Do you recognise this term? (**Hint:** $\vec{p} = m\vec{v}$)

....

The second term represents the potential energy, which in the case of an isolated single atom can only be due to the nucleus or other electrons.

Adding a magnetic field complicates things slightly. The simplest thing we can do is called a **minimal coupling**, which is done by replacing

$$\vec{p} \to \vec{p} - q\vec{A}, \tag{8.3}$$

where q is the charge of the particle and $\vec{A}$ is the **vector potential**, related to the magnetic field as

$$\vec{B} = \vec{\nabla} \times \vec{A} \tag{8.4}$$

This leaves us with the following energy function

$$E(\vec{p},\vec{r}) = \frac{(\vec{p}+e\vec{A})^2}{2m} + g\mu_B\vec{B}\cdot\vec{\sigma} + V(\vec{r}), \qquad \mu_B = \frac{e\hbar}{2m}. \tag{8.5}$$

The second term included here is the so-called **Zeeman term**, which accounts for the difference in energy when an electron has its magnetic moment aligned or anti-aligned with $\vec{B}$. The vector $\vec{\sigma}$ is the spin vector of the electron. Note that the energy is *lower* when $\vec{B}\cdot\vec{\sigma} = -1$, i.e. when the spin points *opposite* to the magnetic field. This is because the electron’s charge is $-e$, leaving the magnetic moment anti-aligned with the spin[7].

This form of the energy function is not very illuminating, so we’ll massage it a bit to get something more readily interpretable. First, we need to set an appropriate vector potential[8] $\vec{A}$. We choose the following[9]:

$$\vec{A} = \frac{1}{2}\vec{B} \times \vec{r}. \tag{8.6}$$

---

[7] Blame Benjamin Franklin for the confusing signs; he chose the convention that electrons carry negative charge and protons positive charge.

[8] There is some freedom here, a **gauge freedom**, just as you are allowed to add some constant to any potential energy and still have the same physics.

[9] This choice is called the *symmetric gauge*.

Exercise: Prove that this choice of $\vec{A}$ satisfies $\vec{\nabla} \times \vec{A} = \vec{B}$.

....

....

....

....

We can plug our choice of $\vec{A}$ into the energy function and expand the square to get

$$E(\vec{p}, \vec{r}) = \underline{\frac{\vec{p} \cdot \vec{p}}{2m}} + V(\vec{r}) + \underline{\frac{e}{2m} \vec{p} \cdot (\vec{B} \times \vec{r}) + g \mu_B \vec{B} \cdot \vec{\sigma}} + \underline{\frac{e^2}{2m} \frac{1}{4} |\vec{B} \times \vec{r}|^2}. \tag{8.7}$$

Note that the part underlined in red is just the energy of the 'free' particle. The part underlined in green can be simplified as follows:

$$\frac{e}{2m} \vec{p} \cdot (\vec{B} \times \vec{r}) = \frac{e}{2m} \vec{B} \cdot (\vec{r} \times \vec{p}) = \mu_B \vec{B} \cdot \vec{L} \qquad \text{where} \qquad \hbar \vec{L} = \vec{r} \times \vec{p}.$$
$$\text{which lets the green term combine to } \rightarrow \underline{\ldots} = \mu_B \vec{B} \cdot (\vec{L} + g \vec{\sigma}). \tag{8.8}$$

We conclude that the energy function is equal to

$$E(\vec{p}, \vec{r}) = \underline{E_0(\vec{p}, \vec{r})} + \underline{\mu_B \vec{B} \cdot (\vec{L} + g \vec{\sigma})} + \underline{\frac{e^2}{2m} \frac{1}{4} |\vec{B} \times \vec{r}|^2} = E_0 + E_{\text{para}} + E_{\text{dia}}. \tag{8.9}$$

Let us discuss the three different terms in this energy function

- The red term is just the energy of a 'free' particle.
- The green term is called the **paramagnetic term**, as it lowers the energy when a $\vec{B}$-field is applied by aligning the magnetic moment (i.e. anti-aligning $\vec{L}$ and $\vec{\sigma}$ with $\vec{B}$[10]).
- The blue term is called the **diamagnetic term**. It is quadratic in B, and can therefore never be negative. It will always increase the energy when a magnetic field is applied. For *small* B it is weaker than the paramagnetic term.

We will now study the paramagnetic and diamagnetic terms in more detail.

8.1 Langevin[11]/Curie[12] paramagnetism

Let us first take a look at the paramagnetic term. Our goal is to derive the magnetic response coefficient χ of a single atom. To do this, we have to make use of **statistical mechanics**. This is a toolbox made to deal with either large numbers of particles, or interactions with so-called 'baths' which usually refer to the environment of the system.

---

[10] Ben Franklin strikes again.

[11] After Paul Langevin, who was a student of Pierre Curie. Paul Langevin is also famous for having an affair with Marie Curie (after Pierre died) while he had a wife himself. It was quite the scandal. Funnily enough, the grandchildren of Paul and Marie got married and had a child who also became a physicist!

[12] After Pierre Curie. This work dates from before Pierre married his mega-brilliant wife[13]. Pierre got run over by a horse-drawn carriage in between his wife's two Nobel prizes. (Look both ways when crossing the street!)

[13] Marie Sklodowska-Curie won the Nobel prize in both physics *and* chemistry

A note on statistical mechanics: This is another topic which we cannot and will not explain in detail. It is a pity, because the methods of statistical mechanics are both powerful and beautiful. To you, it may seem like some black magic box, but we'll again try to highlight the intuition and physical ideas behind it.

The main object we need from the statistical mechanics toolbox is the **partition function**. It is the most fundamental object in *equilibrium* statistical mechanics. All thermodynamic properties[14] of the system can be derived from the partition function.

Intuitively, the partition function tells us the 'volume the system occupies in phase space' at a certain temperature T[15]. It is calculated as follows:

$$\mathcal{Z} = \sum_{\text{conf.}} e^{-E_{\text{conf}}/k_B T} \tag{8.10}$$

Here the sum is taken over all possible configurations, and k_B is the **Boltzmann constant**. From the partition function we can calculate the **free energy** of a system

$$F \equiv -k_B T \log \mathcal{Z}, \tag{8.11}$$

from which, in turn, we can compute the magnetic moment

$$M = -\frac{\partial F}{\partial B}. \tag{8.12}$$

Let's apply these tools to our single atom. We will focus only on the paramagnetic term, and drop both E_0 and the diamagnetic term. We define the **total angular momentum** as

$$\tilde{g}\vec{J} \equiv \vec{L} + g\vec{\sigma}, \tag{8.13}$$

such that

$$E(\vec{p},\vec{r}) = \mu_B \tilde{g} \vec{B} \cdot \vec{J}. \tag{8.14}$$

Here $\tilde{g}$ is called the Landé g-factor. We were studying one particle with a spin J, so in our case the partition function is

$$\mathcal{Z} = \sum_{j=-J}^{J} e^{-\mu_B \tilde{g} B j / k_B T}. \tag{8.15}$$

Let's assume for simplicity that $J = 1/2$, so that there is no orbital angular momentum, only spin angular momentum. Then, j can be $\pm 1/2$ and $\tilde{g} = 2$, such that there are only two configurations ($\pm$). The Partition function is then quite simple,

$$\mathcal{Z} = e^{-\beta\mu_B B} + e^{\beta\mu_B B} \qquad \text{where} \qquad \beta \equiv \frac{1}{k_B T}. \tag{8.16}$$

From the partition function, we get the following expression for the free energy:

$$F = -k_B T \log\left(e^{-\beta\mu_B B} + e^{\beta\mu_B B}\right). \tag{8.17}$$

We can then differentiate with respect to B to get the magnetic moment:

$$M = k_B T \beta \mu_B \frac{e^{\beta\mu_B B} - e^{-\beta\mu_B B}}{e^{\beta\mu_B B} + e^{+\beta\mu_B B}} = \mu_B \tanh(\beta\mu_B B). \tag{8.18}$$

We now have the magnetization as a function of the B-field, from which we can extract the response coefficient:

$$\chi = \lim_{H\to 0} \frac{\partial M}{\partial H} = \frac{\mu_0 \mu_B^2}{k_B T} \qquad \text{where} \qquad B \approx \mu_0 H. \tag{8.19}$$

We conclude that the response coefficient χ is indeed **positive**, as expected for a paramagnet. The expression for the response coefficient is known as **Curie's law**.

---

[14] *Thermodynamic* properties are macroscopic properties such as temperature, free energy, heat capacity, etc.

[15] Don't worry about this if this confuses you and/or means nothing to you.

8.2 Larmor[16] diamagnetism

Earlier, we ignored the diamagnetic term in the energy function, saying it would be ‘small’ compared to the paramagnetic term. Is this always true? We can quite easily imagine scenarios where it is not! When

$$\vec{J} \equiv \vec{L} + g\vec{\sigma} = 0, \tag{8.20}$$

the paramagnetic term vanishes, and the diamagnetic term will dominate. This happen for example when the atom has only shells that are completely full (noble gases), or when a shell has one electron less than being half-filled

Let us here study the effect of the diamagnetic term and ignore the paramagnetic term. Let us take the B-field to lie only in the z-direction ($\vec{B} = B\hat{z}$). Then the diamagnetic energy contribution is on average

$$E_{\text{dia}} = \frac{e^2}{8m}\langle|\vec{B} \times \vec{r}|\rangle = \frac{e^2B^2}{8m}\langle x^2 + y^2\rangle. \tag{8.21}$$

We can use the fact that the atom is rotationally symmetric to say that

$$\langle x^2 + y^2\rangle = \frac{2}{3}\langle x^2 + y^2 + z^2\rangle = \frac{2}{3}\langle r^2\rangle. \tag{8.22}$$

Then

$$E_{\text{dia}} = \frac{e^2B^2}{12m}\langle r^2\rangle. \tag{8.23}$$

Exercise: Compute the magnetic moment and the response coefficient

$$M = -\frac{\partial E}{\partial B} \qquad = \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$

$$\chi = \lim_{H\to 0}\frac{\partial M}{\partial H} \qquad = \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$

As expected, this time χ is negative, meaning that this term is indeed responsible for diamagnetic behavior. Although we could not see this directly in the energy function, the diamagnetic term will cause the moments to anti-align with the magnetic field, since that is energetically favorable.

---

[16] Joseph Larmor was quite a brilliant physicist. He invented Lorentz transformations, which are very important for the theory of relativity, two years before Lorentz did. He always rejected relativity though, in favour of *aether theory*.

9 Magnetic order: spin lattices and the Ising model

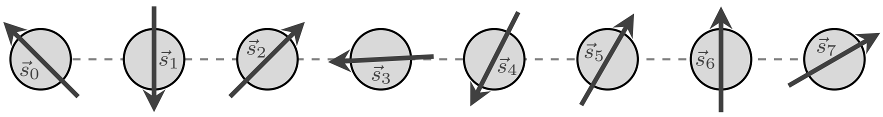
In the previous section we learned how atoms can have paramagnetic and diamagnetic properties, and how these are encoded in and described by energy functions. This was all very fun and cool, but it still teaches us nothing about the ‘magnets’ that we all love and that stick to our fridge. This is because everything we did so far concerned a *single atom’s* response to magnetic fields. We saw that they can indeed align or anti-align their magnetic dipole moments to the field, but there is no concept of ‘permanent magnetivity’ there.

The reason you cannot build a permanent magnet with a single atom is that ferromagnetism is a **collective behavior** of many atoms interacting with each other. A single magnetic moment of order $\mu_b \propto 9.2 \times 10^{-24} J/T$ is very small, so necessarily there must be a lot of moments adding up to make a permanent magnet. (cf. Avogadro’s number: 6.022×10^{23})

Let us look at a simple model that includes many such magnetic dipoles.

9.1 Spin Chains

Imagine a chain of spins, each somehow kept in place. Since the spins cannot move, this material must be an insulator. Now assume that there is an **interaction between neighbors** in the chain, which depends on their mutual orientation.



We may write the energy function of such a spin chain as follows:

$$E(s_i) = -J \sum_{\langle i,j \rangle} \vec{s}_i \cdot \vec{s}_j. \tag{9.1}$$

Where J represents the strength of the interaction, the $\vec{s}_i$ are the spin vectors at each site, and the $\langle i,j \rangle$ represents the set of all neighbours. This is called the **Heisenberg**[17] **model**. Depending on the sign of J, the interaction between neighbors will prefer alignment or anti-alignment, with different ground states (lowest energy state)

- $J > 0 \rightarrow$ Ferromagnet ($M > 0$) $\ldots \uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow \ldots$
- $J < 0 \rightarrow$ Antiferromagnet ($M = 0$) $\ldots \uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow \ldots$
  $\Big($Ferrimagnet ($M > 0$) $\ldots \Big\uparrow \downarrow \Big\uparrow \downarrow \Big\uparrow \downarrow \Big\uparrow \downarrow \Big\uparrow \downarrow \ldots\Big)$

You can clearly see an example of **spontaneous magnetic order** here. Without even trying so hard, we set up a model whose ground state is a ferromagnetic state. In fact, we are only one step removed from one of the most important models/concepts in modern physics.

To see what we’re missing, let’s make the simple observation that the Heisenberg model is rotationally symmetric. That is, if you rotate all spins by some fixed angle α, the energy of the system is unchanged. In real systems (crystals) this is almost never the case and there will be some preferred direction. However small this effect may be, it will **break the symmetry**. We can incorporate this in the energy function by adding a term

$$E(s_i) = -J \sum_{\langle i,j \rangle} \vec{s}_i \cdot \vec{s}_j - \kappa \sum_i (s_i^z)^2. \tag{9.2}$$

---

[17] Werner Heisenberg is more famous for other things, like inventing quantum mechanics, and having a really cool name.

This results in a preference of the spins to (anti-)align with the z-axis. κ is a parameter controlling the strength of this preference, and is called the **anisotropic energy**.

One can quite easily write down other anisotropic terms to add to the energy function (can you think of one?), which will lead to different behavior.

9.2 The Ising[18] model

If the anisotropic energy κ is very large, every spin will effectively be forced to point fully in the z-direction (up or down), i.e.

$$s_i^z = \pm s_i. \tag{9.3}$$

This allows us to somewhat simplify the energy function

$$E(\sigma_i) = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j + g\mu_B B \sum_i \sigma_i, \tag{9.4}$$

where we also brought back the magnetic field B.

This energy function represents the **Ising model**, an extremely important model in statistical physics. The Ising model can be thought of as the 'hydrogen atom' of statistical physics, because despite its simplicity it contains many of the fascinating features of very complex systems. In this way, it opens a metaphorical window to understanding not only magnets, but also phase transitions, quantum computing, AI, ...

The model has been exactly solved in 1D (Ising, 1925), in 2D (Onsager, 1944)[20], and in $d > 4$ but not yet in 3D. People are still actively working on the Ising model, with more than 1000 scientific publications on the topic in 2023.

We will now enter into some detail regarding the Ising model, and obtain an approximate solution using techniques similar the those we borrowed from statistical physics earlier. But you can ask: what does "solve" really mean? What do we want from the Ising model? We would like to get an expression for the magnetization of the system as a function of the temperature. If we have this, we can see if the model exhibits ferromagnetic (or other) behavior at a certain temperature. So, what we really want is an expression for $M(J, B, T)$.

A note on temperature: What is temperature? (Do you know?) For an ideal gas, T is the average velocity of the particles. In a solid, it may be some measure of the vibration of the atoms, but this is already very vague. The best we can do here is say that $k_B T$ sets the energy scale of the fluctuations in the system. This scale then influences the occupation of higher-energy states and the volume in phase space. This is why it pops up in the partition function $\mathcal{Z}$.

Solving the Ising model is very hard. It has only been done *exactly* in 1D and 2D[21], and Ising recieved his PhD for this result. What we will do here is make an approximation, called the **mean field approximation**. The mean field scheme consists of two steps:

1. Examine one cell, and approximate all other cells as having some mean (or average) value. Then, figure out the behavior of this one cell.
2. Impose self-consistency i.e. require that the mean value used earlier is the actual mean value.

---

[18] Properly pronounced "*ee-zing*"[19]. Ising was a student of Lenz who worked out a solution in 1D to this model for his Ph.D in 1925. The actual inventor of the model is Wilhelm Lenz.

[19] I myself often slip and say "*eye-zing*". Luckily, people understand you both ways.

[20] Wolfgang Pauli wrote after WWII that "Nothing much of interest has happened in physics during the war, aside from Onsager's solution". This is a bold claim if you keep in mind how the war ended.

[21] and $d \geq 4$ using mean-field theory.

9.3 Mean field theory for the Ising model

Our starting point is again the Ising model energy function

$$E(\sigma_i) = -\frac{1}{2} J \sum_{\langle i,j\rangle} \sigma_i \sigma_j + g\mu_B B \sum_i \sigma_i, \tag{9.5}$$

where we let the σ_i take the values $\pm 1/2$ and choose $J > 0$. From this, we can write down the energy of one spin

$$E_i(\sigma_j) = \left(g\mu_B B - J \sum_{\langle i,j\rangle} \sigma_j \right) \sigma_i = g\mu_B B_{\text{eff},i} \sigma_i. \tag{9.6}$$

Now comes the approximation: instead of using the proper $B_{\text{eff},i}$ that is different for each spin σ_i, we use the mean value $\langle B_{\text{eff}} \rangle$, leading to the following energy function.

$$E_i(\sigma_j) = g\mu_B \langle B_{\text{eff}} \rangle \sigma_i. \tag{9.7}$$

This equation might ring a bell: it's the same energy function as the one we used to find the paramagnetic response coefficient! The partition function for this energy function was

$$\mathcal{Z}_i = e^{-\beta g\mu_B \langle B_{\text{eff}} \rangle /2} + e^{\beta g\mu_B \langle B_{\text{eff}} \rangle /2} \qquad \text{remember that } \beta = \frac{1}{k_B T}, \tag{9.8}$$

and from this we can calculate the average value of the spins σ_i: $\langle \sigma \rangle = -M/g\mu_B$

$$\begin{aligned} \langle \sigma \rangle = -\frac{M}{g\mu_B} &= \frac{1}{g\mu_B} \frac{\partial F}{\partial B} \\ &= -\frac{1}{\beta g\mu_B} \frac{\partial}{\partial B} \left[\log \mathcal{Z} \right] \\ &= -\frac{1}{\beta g\mu_B} \frac{1}{\mathcal{Z}} \frac{\partial \mathcal{Z}}{\partial B} \\ &= -\frac{1}{2} \frac{e^{\beta g\mu_B \langle B_{\text{eff}} \rangle /2} - e^{-\beta g\mu_B \langle B_{\text{eff}} \rangle /2}}{e^{\beta g\mu_B \langle B_{\text{eff}} \rangle /2} + e^{-\beta g\mu_B \langle B_{\text{eff}} \rangle /2}} \\ &= -\frac{1}{2} \frac{\sinh(\beta g\mu_B \langle B_{\text{eff}} \rangle /2)}{\cosh(\beta g\mu_B \langle B_{\text{eff}} \rangle /2)} \\ &= -\frac{1}{2} \tanh(\beta g\mu_B \langle B_{\text{eff}} \rangle /2) \end{aligned} \tag{9.9}$$

This was already the first step in the mean field scheme, but notice that our expression for $\langle \sigma \rangle$ contains $\langle B_{\text{eff}} \rangle$, which in turn contains $\langle \sigma \rangle$:

$$\begin{aligned} g\mu_B \langle B_{\text{eff}} \rangle &= g\mu_B B - J \sum_{\langle i,j\rangle} \langle \sigma_j \rangle \\ &= g\mu_B B - Jz \langle \sigma \rangle \qquad \text{(if you assume all spins have equal statistics)}^{22} \end{aligned} \tag{9.10}$$

Here z is the number of neighbors the spin has (depends on the dimension)[23].

Thus, the expression for $\langle \sigma \rangle$ itself contains $\langle \sigma \rangle$! This brings us naturally to the second step in the mean field scheme: of course, when we plug $\langle \sigma \rangle$ into itself, we want to give the same result. That is, we want our solution to be **self-consistent**. The self-consistency equation is

$$\langle \sigma \rangle = -\frac{1}{2} \tanh\left(\frac{\beta}{2} \left[g\mu_B B - Jz \langle \sigma \rangle \right] \right), \tag{9.11}$$

---

[22] Which we do!

[23] Notice that we did not specify the dimension until now. The *only* difference between dimensions in this calculation is the number of neighbours each spin has!

which we will solve *by drawing*. For simplicity we set $B = 0$ (Can you see what changes if $B \neq 0$?)

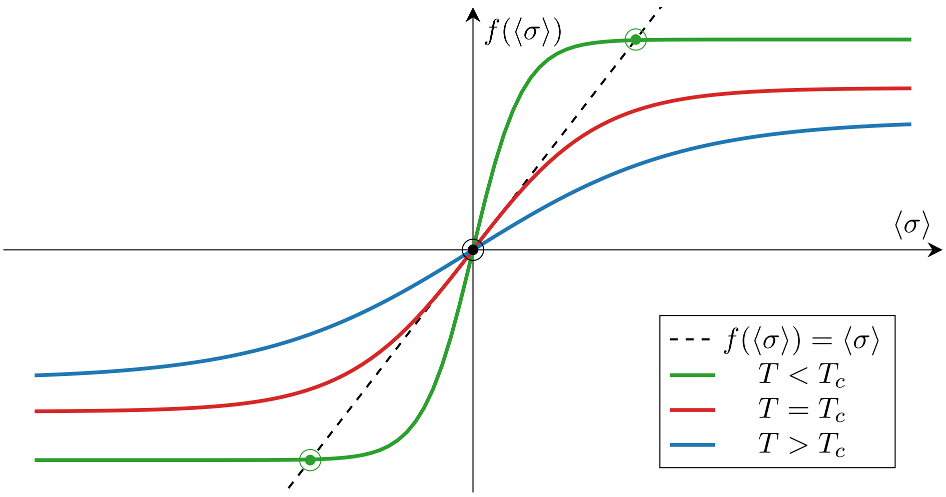


Figure 2: Plot of the tanh function for high temperatures (blue), low temperatures (green) and the critical temperature (red).

We have drawn the function $f(\langle\sigma\rangle) = \tanh\left(\beta\frac{Jz\langle\sigma\rangle}{2}\right)$ as a function of $\langle\sigma\rangle$, for three different values of β (or T). We have also indicated the curve $f(\langle\sigma\rangle) = \langle\sigma\rangle$ with dashed lines. There are two different regimes, separated by the red curve:

- **The blue curve, high values of T:** this curve crosses the dashed line only once, at $\langle\sigma\rangle = 0$. So, the only solution to the self-consistency equation is $\langle\sigma\rangle = 0$. At these temperatures, the system is not ferromagnetic. (But it *is* paramagnetic![24])
- **The green curve, low values of T:** this curve has not one crossing with the dashed line, but three! This means that the self-consistency equation has a solution with $\langle\sigma\rangle \neq 0$, so the system can have nonzero magnetization even in the absence of a magnetic field! This is the very definition of a ferromagnetic state!
- **The red curve, $T = T_c$:** the red curve marks the boundary between the paramagnetic and ferromagnetic regimes. The red line occurs at the temperature where the slope of the curve at the origin is equal to one. This temperature is called the **Curie temperature**[25], and every permanent magnet has one.

To calculate the Curie temperature, we can compute the slope of the curve by taking a derivative and then setting $\langle\sigma\rangle = 0$:

$$\frac{d}{d\langle\sigma\rangle}\tanh\left(\frac{B}{2}Jz\langle\sigma\rangle\right)\Big|_{\langle\sigma\rangle=0} = \frac{\beta Jz}{2}\operatorname{sech}^2\left(\frac{B}{2}Jz\langle\sigma\rangle\right)\Big|_{\langle\sigma\rangle=0} = \frac{\beta Jz}{2}. \tag{9.12}$$

We want this slope to be equal to 1, so we get

$$k_B T_c = \frac{Jz}{2} \tag{9.13}$$

Up until now, we have not actually specified in how many dimensions we are working. In 4D, the mean field method gives you the exact result. In 1D, 2D and 3D, it is wrong.

---

[24] Can you see why?

[25] Again after Pierre Curie

In 1D, mean field theory predicts a Curie temperature of $k_B T = J$ (because $z = 2$). However, it has been known for 99 years now that there is **no phase transition in 1D**. In 2D and 3D the wrongness is more subtle, but the qualitative behavior is still the same. Let us take a closer look at this phase transition:

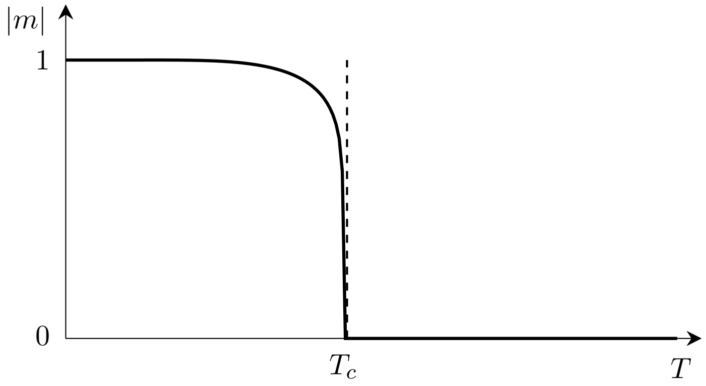


Figure 3: Magnetization per spin for the 2D Ising model as a function of the temperature, for $h = 0$ and $J = 1$.

Here we plot the solution of the Ising model as a function of temperature. You will observe this phase transition by simulating the Ising model on your computer later!

To conclude, the Ising model is a model that has generated invaluable insights and describes the ferro-paramagnetic phase transition pretty well. Still, it is clearly not a fully accurate description of the ferromagnets we know and love. In the next section, we'll briefly go over the shortcomings of the Ising model.

9.4 Real ferromagnets

Magnetic domains

You might think that in a ferromagnet, just as in the Ising model, *all* the spins will align. This is, however, not what we see in real ferromagnets. Let's try to understand why with a drawing:

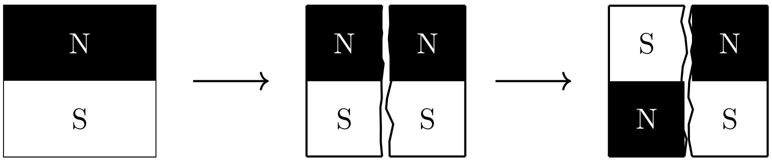


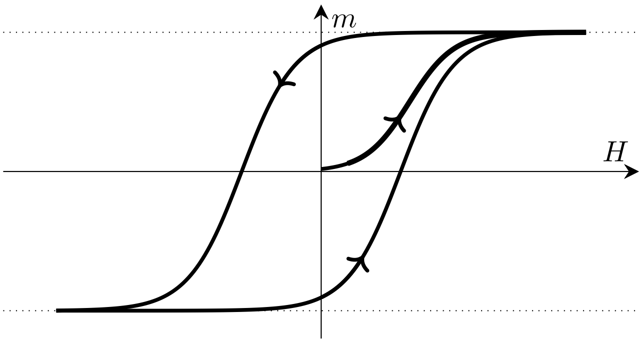
Figure 4: Imagine a fully aligned magnet, and imagine we cut it in half. Intuitively, you may feel it is more natural to flip one of the halves!

Indeed, when discussing Hund's first rule, we ignored dipole effects in favor of Coulomb interactions. But, when you put $\sim 10^{23}$ of them close together, the situation changes. At some point, the energy gained by flipping a whole lot of spins will outweigh the cost of forming the $\uparrow\downarrow$ border.

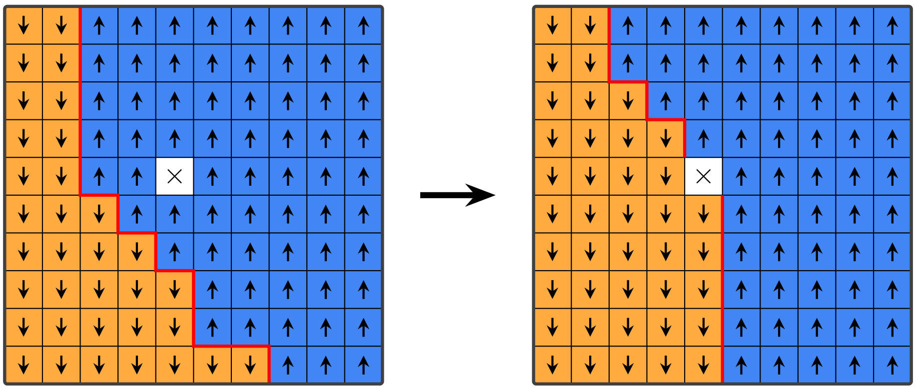
The small regions that form in this way are called **Weiss domains**. The border between such regions is called a **domain wall**.

Hysteresis

An important property of real ferromagnets is the fact that they exhibit a hysteresis effect. This is related to the 'activation energy' needed to realign a Weiss domain, i.e. to reconfigure the spins. This is what allows permanent magnets to stay magnetized!



Strong hysteresis is achieved by making domains hard to reconfigure. One way to do this is by adding disorder and/or impurities. Adding impurities gives the domain walls 'anchors'. It pins them to the impurities, with stronger hysteresis as a result.



Final note: Insulators and conductors

To end with, I would like to say a word about conductors. As you might realize, these spin lattice models we have been discussing are all models of *insulators*. We chose them to be that way when we chose the spins to have fixed positions. This choice simplified our task of setting up a mathematical model significantly, because the energy function does not depend on the movement or positions of the spins, letting us focus purely on their orientation.

As you might also realize, there exist ferromagnets that conduct. The most famous ferromagnet, *after which ferromagnets are named* is iron, which is of course a conductor. The difference in the case of conductors is that now the electrons can move around, generating a new layer of complexity.

There exist techniques to deal with this complexity (to some extent), but these techniques all belong to the field of *quantum many-body physics*[26]. While there are a lot of worthwhile things to say about this topic, we are limited in time and space, and we won't have the time to get

---

[26] Which is known for being very hard.

into it properly[27].

Let me just say a few words about it before we finish.

Itinerant ferromagnetism The name for ferromagnetism when the electrons are free to move around is *itinerant ferromagnetism*[28]. There is an immediate puzzle that pops up, since in a *free electron gas* the lowest-energy configuration is when the electrons pair up and have equal numbers of *up* and *down* spins. So what gives? The responsible interaction is again the Coulomb interaction, similarly to in Hund's first rule.

Quantitative calculations and simulations have been performed in simplified models, the most famous of which is the **Hubbard model**[29], which, like the Ising model, is a lattice model, with an on-site repulsion (an energy penalty if two electrons are on the same site).

Ferromagnetism has been observed/calculated in the Hubbard model, and to this day it gives some of the best insight we have on how electron interactions can lead to aligned spins.

---

[27] That would take a full-semester university course!

[28] From the Latin word *itiner*, which means *journey*.

[29] After the British physicist John Hubbard. This model is widely used throughout condensed matter physics, but also more specifically to the study of *High-T_c Superconductors*, more on that later.

Chapter III

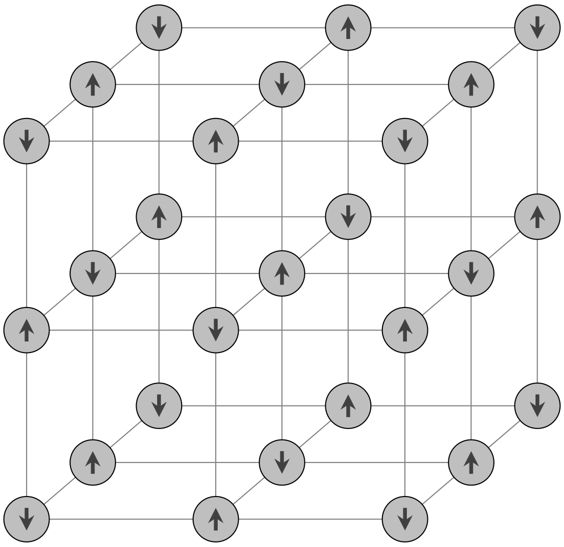
The Ising model and Metropolis algorithms

10 Recap: the two dimensional Ising model

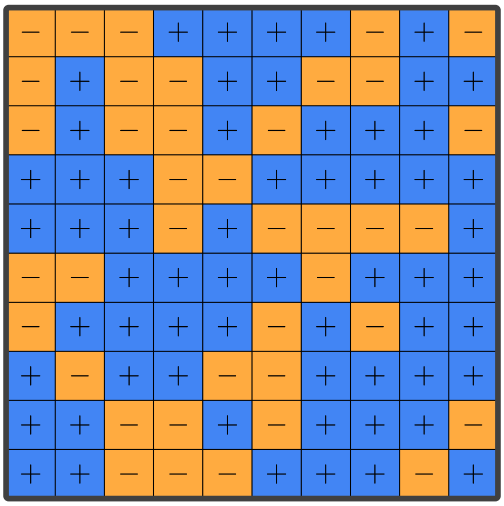
In the previous lectures, you saw that the Ising model is a very nice and relatively easy example of a statistical model. The reason that physicists like it so much, is that it is analytically solvable. This means that you can calculate all the relevant properties of this model exactly, and you don't need to make approximations. You can use the Ising model to describe different things, but we will mostly think of it as a model for certain types of magnets.

In a magnet, the particles are organized in a lattice structure. Each particle has its own magnetic moment. For one particle, the magnetic moment is tiny, and you will probably not notice it. However, when all the particles in the lattice work together, these tiny magnetic moments all sum up to form one big magnetic field inside and around the magnet.

In real life the lattice structure of a magnet is usually 3D, since a magnet is usually a 3D object. However, since a 3D lattice is hard to visualize, we will work with 2D lattices here. We will use the Ising model and a computer algorithm to simulate the behaviour of a 2D magnet.



We will consider a two-dimensional $N \times N$ lattice. At each site sits a **spin**, that can either be up or down ($+$ or $-$). We label the different spins in the lattice by σ_i. A specific combination of up and down spins is called a **state**. You can find an example of such a state for a 10×10 lattice in the figure below. We will give every state a name, and we usually use Greek letters for this. Let us call this state μ.



−	−	−	+	+	+	+	−	+	−
−	+	−	−	+	+	−	−	+	+
−	+	−	−	+	−	+	+	+	−
+	+	+	−	−	+	+	+	+	+
+	+	+	−	+	−	−	−	−	+
−	−	+	+	+	+	−	+	+	+
−	+	+	+	+	−	+	−	+	+
+	−	+	+	−	−	+	+	+	+
+	+	−	−	+	−	+	+	+	−
+	+	−	−	−	+	+	+	−	+

Figure 5: A 10×10 lattice filled with specific choices for the spins. This is one possible state of the lattice, which we call μ.

Every such state of the lattice has an **energy** associated to it. As you saw in the previous lectures, the energy of a state μ in the Ising model is given by[1]

$$\boxed{E_\mu = -J \sum_{<i\,j>} \sigma_i \, \sigma_j \; - h \sum_j \sigma_j} \qquad (J > 0). \tag{10.1}$$

As you can see, the energy consists of two parts:

- The first part originates from the **interaction** between neighboring states. Spins like to be aligned, they want to point in the same direction as the spins around them. If two neighboring spins point in the same direction (say they both point up), they will have a lower energy than when one points up and the other points down. In the Ising model, only direct neighbors are considered when calculating this energy term. This means that for a two dimensional square lattice, every spin has four neighbors to consider.

  You can calculate this energy term as follows: Consider every pair of neighboring spins in the lattice (every grid line between two cells). For every pair, multiply their spins. For example, if one of the spins (σ_1) is up and the other (σ_2) is down then $\sigma_1 . \sigma_2 = (+1).(-1) = -1$ [2]. You add up all these results and multiply by $-J$. Because of the minus sign in front, any pair of same spins will lower the energy, while every pair of different spins raises the energy. The number J determines the importance of aligning the spins, or the strength of the interaction between neighbors. If you make J very large, the spins will very much want to align themselves.

---

[1] Here the symbol $\sum\limits_{<i\,j>}$ means that you sum over all pairs of neighboring spins σ_i and σ_j, and $\sum\limits_{\sigma_j}$ just means the sum over all spins σ_j. We also redefined $h \equiv g\mu_B B$, and the minus sign difference is just a convention.

[2] Note that we now use $\sigma_i = \pm 1$ and not $\sigma_i = \pm 1/2$. This is just because it is easier to calculate with.

- The second part of the energy describes the influence of an **external magnetic field**. If there is a magnetic field present outside the lattice, the spins will want to align with that. If h is positive, all spins will want to be up and vice versa. This term is even easier to compute: you just add up all the spins, and then multiply by $-h$.

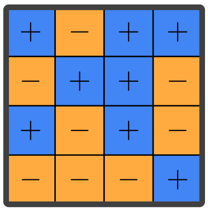
  The sum of all the spins is also called the **total magnetization** M of the lattice. You can easily see that the sum of all spins is also equal to the number of $+$ spins minus the number of $-$ spins.

  $$\boxed{M = \sum_i \sigma_i = N_+ - N_-} \tag{10.2}$$

  As you can see, the second term in the energy is negative when the total magnetization has the same sign as h, and positive when they have an opposite sign.

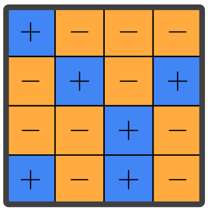
Note that we will always put $k_B = 1$ unlike in the previous lectures!

Exercise: Calculate the energy of the following states of a 4×4 lattice with empty boundary conditions (the spins at the edge have only 3 or 2 neighbors instead of 4):



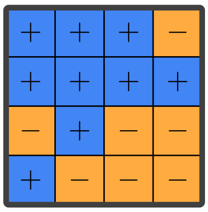
$+$	$-$	$+$	$+$
$-$	$+$	$+$	$-$
$+$	$-$	$+$	$-$
$-$	$-$	$-$	$+$

$E =$



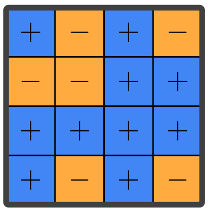
$+$	$-$	$-$	$-$
$-$	$+$	$-$	$+$
$-$	$-$	$+$	$-$
$+$	$-$	$+$	$-$

$E =$



$+$	$+$	$+$	$-$
$+$	$+$	$+$	$+$
$-$	$+$	$-$	$-$
$+$	$-$	$-$	$-$

$E =$



$+$	$-$	$+$	$-$
$-$	$-$	$+$	$+$
$+$	$+$	$+$	$+$
$+$	$-$	$+$	$-$

$E =$

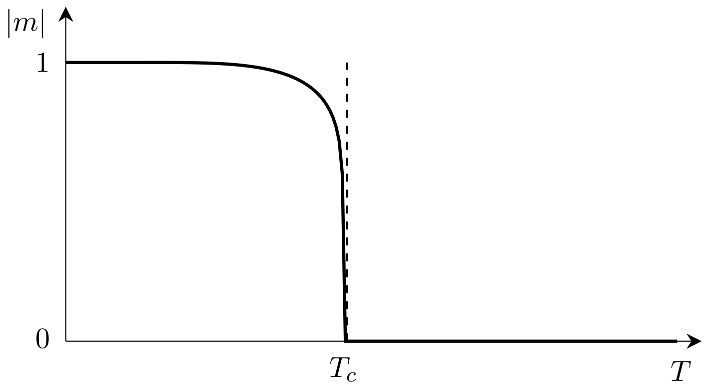
The Ising model is a way to describe microscopic properties of certain magnets. However, from these microscopic properties we can also derive some macroscopic properties. Let us, for example, look at the magnetization of the lattice. We already know the total magnetization M, but often we use the **magnetization per spin** or **relative magnetization** (m) instead. This is defined as the total magnetization divided by the total number of spins. For an $N \times N$ lattice, this is

$$\boxed{m = \frac{M}{N^2} = \frac{N_+ - N_-}{N^2}} \tag{10.3}$$

You can try to calculate the expectation value for m analytically for the Ising model, and you will find that it is impossible to solve this for general values of J and h. You can do one of two things to get a solution for m: you can solve the Ising model numerically, or you can put $h = 0$. We will do the second. If you put $h = 0$ (so no external magnetic field), then you can prove that

$$|m| = \left[1 - (\sinh(2J/T))^{-4}\right]^{1/8} \tag{10.4}$$

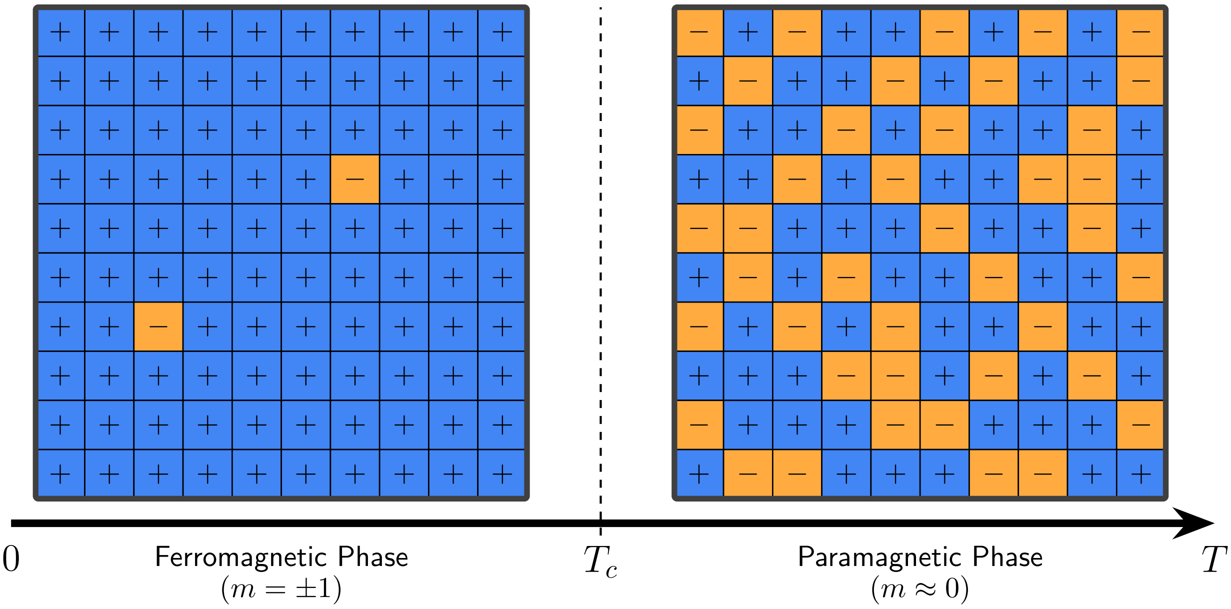
This is a rather complicated function, so let us plot it to illustrate its behaviour. The analytic solution for m for $h = 0$ and $J = 1$ looks something like this:



As you can see, the magnetization is zero for high temperatures (above $T = 2.26...$). A lattice with $m = 0$ should have a perfect mix between up and down spins ($+1$ and -1), so that the average value of a spin is perfectly zero. For very low temperatures, the absolute value of m is 1, so m is either $+1$ or -1. This would happen if the lattice would be entirely filled with up spins (for $m = +1$) or entirely filled with down spins (for $m = -1$).

This means that the lattice behaves very differently for low temperatures vs temperatures above $T = 2.26....$ We call this phenomenon a **phase transition**. The temperature at which this happens is called the **critical temperature** T_c. You can compare this to the phase transitions in for example a bottle of water: at very low temperatures, the water is solid (ice). Once you pass $T = 0°C$, the ice will melt and the water is a liquid. If you go even higher, once you pass $T = 100°C$ your water will become a gas.

The phase transition of a lattice in the Ising model is similar, but different. Below the critical temperature, the lattice is mostly filled with one type of spin. We call this the **ferromagnetic phase**. It behaves like the sort of magnet you are used to: it will create its own magnetic field because all the spins work together. Above the critical temperature, all the spins point in random directions. We call this the **paramagnetic phase**. You will not notice a magnetic field around the lattice, because the spins are not working together at all.



+	+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	−	+	+	+
+	+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+	+
+	+	−	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+	+

−	+	−	+	+	−	+	−	+	−
+	−	+	+	−	+	−	+	+	−
−	+	+	−	+	−	+	+	−	+
+	+	−	+	−	+	+	−	−	+
−	−	+	+	+	−	+	+	−	+
+	−	+	−	+	+	−	+	+	−
−	+	−	+	−	+	+	−	+	+
+	+	+	−	−	+	−	+	−	+
−	+	+	+	−	−	+	+	+	−
+	−	−	+	+	+	−	−	+	+

In the plot at the top of the page, we only showed the critical temperature if you put $h = 0$ and $J = 1$. There is an analytic formula for the critical temperature for any value of J, but you still have to put $h = 0$ since you need to be able to solve the Ising model analytically. For any value of J and for $h = 0$, the critical temperature is given by[3]

$$\boxed{T_c = \frac{2J}{\ln\left(1+\sqrt{2}\right)}} \tag{10.5}$$

You might be curious as to what happens if you manage to tune the temperature to exactly the critical temperature. If you do this, something very strange will occur, which does not happen in the phase transition of water from solid to liquid. Your spins will form zones of up-spins and zones of down-spins, and these zones will have all possible sizes. There will be very small zones and very large zones. The special thing is that, if you zoom out and look at a bigger portion of the lattice, it will look exactly the same, since zones of all sizes exist. This property is called **scale invariance** or **conformality**, and is something very special in physics. We will not discuss it more here, but if you are interested you can watch the following very interesting video's on YouTube:

- In this video, you can experience the scale invariance of the critical Ising model yourself: `https://www.youtube.com/watch?v=fi-g2ET97W8`
- Something like this does also happen for water, but not at normal circumstances you will experience on Earth. If you put water in a very high pressure and a certain temperature, you will be at a critical point like the one in the Ising model. If you go beyond that, the water will be in a weird state which is neither liquid nor gas, but a mixture between the two. It is called a supercritical fluid. You can learn more about this phenomenon in this video:
  `https://www.youtube.com/watch?v=zv4sE7R8QO4`

---

[3] To get a feeling for how large the critical temperature of the Ising model is, we need to rewrite it in the proper units. Note that earlier we chose to set $k_B = 1$ for simplicity. This messed up our units, so let's put it back. The critical temperature is really given by $k_B T_c = 2J/\ln\left(1+\sqrt{2}\right)$. J has units of energy (Joules), and k_B has units Joules per Kelvin, so that if you know J for your lattice, in the correct units, you get a temperature in Kelvins.

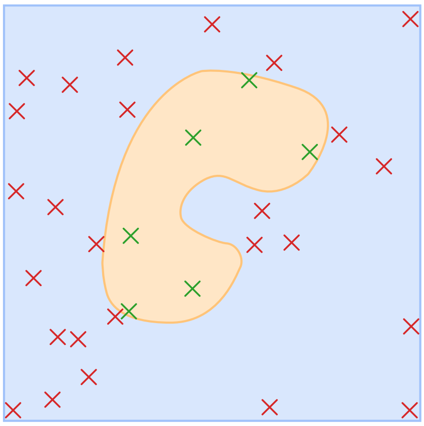
11 Metropolis Monte Carlo simulations

11.1 Simple Monte Carlo methods

Very often in physics we want to do computations that cannot be performed analytically, meaning that we cannot find an exact solution to the problem. This can either be because calculating exactly might take a very long time, or because there does not exist an exact solution to the problem. In these cases we want to find an **approximate solution**, a value that is very close to the exact solution. We also call this a **numerical solution** to the problem, and we often use a computer to calculate it.

There are many different methods to do numerical computations, depending on the kind of problem you are dealing with. One collection of numerical methods is called **Monte Carlo methods**[4]. The most important thing about these methods is that they all use random numbers to calculate the result, and the more random numbers you use in your calculation, the better the result will approximate the exact value. Let us look at two very simple Monte Carlo methods for two different problems.

Example 1: *Area of a blob with uniform sampling*
Suppose you draw a random shape (a blob) inside a square of known size, and you want to know the area of this blob. Since your blob does not have a mathematical equation that describes it, there is no way to calculate its area exactly. The only thing you can do is try to approximate the result numerically. We will describe a very simple Monte Carlo method to calculate the area of the blob.



Generate a bunch of random points inside your square, uniformly distributed in the whole interior. For every point, determine whether it lies inside the blob or outside. You can then approximate the area of the blob as follows:

$$\frac{\#\text{ points inside the blob}}{\text{total }\#\text{ points}} \approx \frac{\text{area of blob}}{\text{area of square}}. \tag{11.1}$$

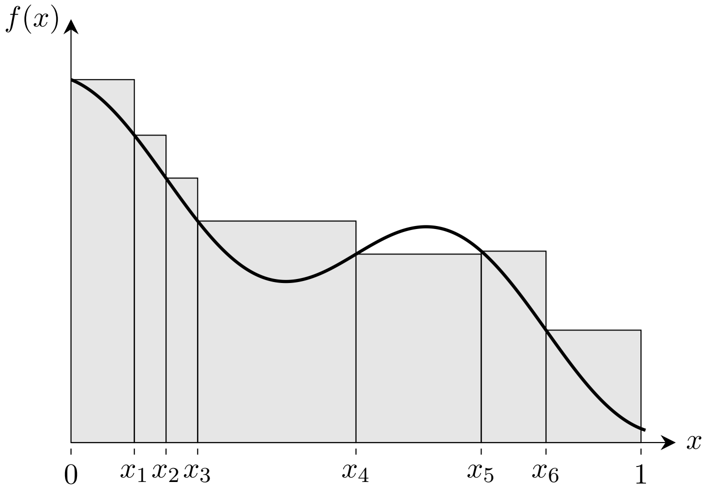
Of course, if you take only a few points this will be a very bad approximation. If you generate only three points, it could be that all three of them lie outside the blob and the result for the area of the blob would be zero. This method only works if you take many, many points. The more points you generate, the closer the result will be to the real value.

Example 2: *Area under the curve of a function (integral) using uniform sampling*
Another example of a very simple Monte Carlo method is the uniform sampling method of

---

[4] Named after a neighborhood in Monaco that is renowned for its casino's, where random numbers play a very important role...

calculating the area under a curve (for those who know what this is, the integral of a function). If the function has a nice analytic formula, you can calculate this exactly. However, if you just draw a random curve or if the function is known but has no exact formula (like the erf(x) function), you will need to do a numerical computation. A uniform sampling Monte Carlo method is one way to calculate the area under the curve numerically.



Suppose that we want to calculate the area under the curve $f(x)$ between $x = 0$ and $x = 1$. Generate a number of points x_i between 0 and 1, uniformly distributed. For each point calculate the area of the rectangle that fits under the curve, as illustrated in the figure above. If you sum up all these rectangles, you will get an approximation for the area under the curve.
As in the previous example, this method only works well if you generate many points. If you only have two points, you will only use three rectangles and the approximation will be very bad. If you generate many points, the rectangles will become very slim, and they will fit a lot better under the curve, giving you a closer approximation to the real area.

The examples above are two very simple examples of Monte Carlo methods. You might have noticed that in these two examples it was not strictly necessary to use random numbers. You could also have used a grid or sequence of even-spaced points to calculate the areas. These simple Monte Carlo methods are not better or faster than other methods that do not use randomness. However, keep in mind that these are really the simplest kind of Monte Carlo methods there are. There exists a whole zoo of Monte Carlo methods, and many of them are much more complex than the uniform sampling methods we discussed. These more complex methods heavily rely on the use of random numbers, and are often a lot better and much faster than other methods that do not use randomness. One of these more complex methods are called **Metropolis-Hastings algorithms**[5] (or Metropolis algorithms for short). We will use a Metropolis algorithm to simulate the Ising model.

If you want to learn more about Monte Carlo methods, here is a link to a nice video on YouTube that explains a simple Monte Carlo method to estimate the value of π, and also the basics of rendering images for animated movies.
`https://www.youtube.com/watch?v=7ESK5SaP-bc`

---

[5] Named after Nicholas Metropolis for first version of the algorithm and Wilfried Hastings who generalized it 17 years later. Metropolis was head of the Theoretical Division of Los Alamos after WWII, where Monte Carlo methods were first developed. Under his guidance, the Los Alamos group built the computers MANIAC I and II that were used to test these Monte Carlo methods. There exists some controversy, however, regarding the credit for developing the Metropolis algorithm. Although Metropolis claimed the title, it was actually plasma scientist Marshall Rosenbluth and his wife Arianna Rosenbluth, together with Edward and Augusta Teller who really developed it. Metropolis' only contribution was to provide them with computer time on the MANIAC.

11.2 Metropolis algorithms

With a Metropolis algorithm, you can simulate the behavior of a system as a function of time. Your system will start in a certain state μ, and in every time step you will determine whether your system will jump to another state ν or not. As such, your system will undergo a series of transitions from one state to the next and evolve in time. Which transitions you make is chosen randomly. A sequence of the states of a system during Metropolis simulation might look like this:

$$\mu \to \nu \to \nu \to \rho \to \mu \to \rho \to \sigma \to \rho \to \dots \tag{11.2}$$

where every arrow is one time step.

Every time step of a Metropolis algorithm consists of two parts. Say that you are currently in a state μ, you perform the following actions during this time step:

- First you select a target state ν. Every transition from μ to another state ν has a certain probability of being selected, which we call the **selection probability** $g(\mu \to \nu)$.
- Then you determine whether you will accept the transition to the new state or not. If the transition $\mu \to \nu$ lowers the energy of your system, you always accept the transition. If it raises the energy you will sometimes accept and sometimes decline, depending on the temperature. This means that sometimes, nothing happens during a time step! The full **acceptance ratio** (probability of accepting the transition) is given by

$$\boxed{A(\mu \to \nu) = \begin{cases} e^{-(E_\nu - E_\mu)/T} & E_\mu < E_\nu \\ 1 & E_\mu \geq E_\nu \end{cases}} \tag{11.3}$$

Note how this acceptance ratio is inspired by statistical physics! Remember that the probability to be in a state μ was

$$p_\mu = \frac{1}{\mathcal{Z}} e^{-E_\mu/T}. \tag{11.4}$$

If you want to raise the energy by going from state μ to state ν, you are moving towards a state with a lower probability ($p_\nu < p_\mu$). The acceptance ratio for $E_\mu < E_\nu$ is

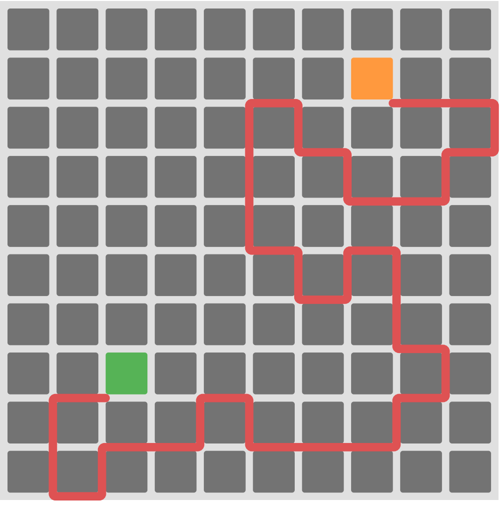
$$A(\mu \to \nu) = \frac{p_\nu}{p_\mu} = \frac{e^{-E_\nu/T}/\mathcal{Z}}{e^{-E_\mu/T}/\mathcal{Z}} = e^{-(E_\nu - E_\mu)/T} \tag{11.5}$$

Altogether, the possibility of going from a state μ to another state ν in one time step is given by

$$\begin{aligned} P(\mu \to \nu) &= g(\mu \to \nu) \cdot A(\mu \to \nu) \\ &= \begin{pmatrix} \text{selection} \\ \text{probability} \end{pmatrix} \cdot \begin{pmatrix} \text{acceptance} \\ \text{ratio} \end{pmatrix}. \end{aligned} \tag{11.6}$$

Example: *Metropolis for a random walker*
Let us apply the Metropolis algorithm to a system called a **random walker**. Imagine a person trying to walk home after a night of drinking at the bar. This person is slightly drunk, and has the tendency to wander around a bit and take wrong turns on his way home. Our system consists of a lattice of streets surrounding the buildings in the city, and one random walker who can sit at any crossroads in the city. The random walker wants to get from the bar (upper right corner of the orange building) to his home (lower left corner of the green building).

Every time step the random walker decides whether or not to walk one street from his current corner to another. If he succeeds in making this transition to the next state, a red line appears an he is now on a new corner. This process is repeated every time step.



To be more precise, during every time step the following process occurs:

- You are currently at a corner μ. The first step was to select a target state ν. Any ν that is one street away from μ can be selected with equal probability. Our selection probability $g(\mu \to \nu)$ is given by

  $$g(\mu \to \nu) = \begin{cases} 1/4 & \mu \text{ and } \nu \text{ are connected by one street,} \\ 0 & \text{else.} \end{cases} \tag{11.7}$$

  This means that in one time step the walker can only walk the length of one block. The direction that he walks in is selected randomly.

- The second step was to determine whether you will accept the transition to the new state or not. This was determined by the energy. For this system, we assume that the energy of the random walker is related to the distance to his home. Say $E(d) = \lambda \cdot d$ for a straight distance d from the walker to his home, and λ is a positive number. You can see that the further away from his home the walker is, the higher his energy will be.

  If the transition to the new state (walking to the new corner) brings him closer to his home, the transition is always accepted. If the transition brings him further away from his home, this would raise the energy. In that case, you will accept the transition with a certain probability (see (11.3)). To summarize, the acceptance ratio of a transition from corner μ to corner ν is

  $$A(\mu \to \nu) = \begin{cases} e^{-\lambda(d_\nu - d_\mu)/T} & \text{if } \nu \text{ is further away than } \mu, \\ 1 & \text{if } \nu \text{ is closer than } \mu. \end{cases} \tag{11.8}$$

You can see that the random walker has a tendency to walk ever closer to his home, and eventually he will end up in the right place. However, every now and then he wanders in the wrong direction, and takes a detour. This randomness is the reason the walker does not take the shortest route from the bar to his home.

12 A Metropolis algorithm for the Ising model

We will now use a Metropolis algorithm to simulate the behavior of the Ising model. Our system will be a 2D lattice[6] of spins of size N. Every configuration of up and down spins makes one state. One time step of the Metropolis algorithm will go as follows:

- You are currently in a state μ, a certain configuration of up and down spins in the lattice. The first step was to select a target state ν. We will only allow transitions to states that differ from μ by exactly one spin. Therefore, the only transition that you can make in one time step is to flip one of the spins in the lattice:

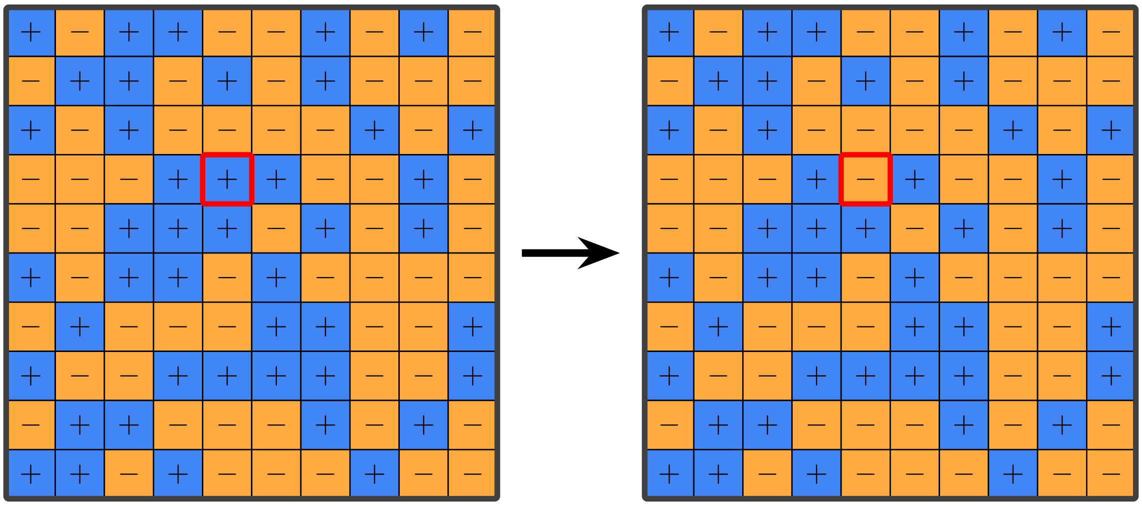


Figure 6: A possible transition from a state μ to a new state ν with only one spin flipped (indicated in red).

  Mathematically, you can write the selection probability as follows:

$$g(\mu \to \nu) = \begin{cases} 1/N^2 & \text{if } \mu \text{ and } \nu \text{ differ by only one spin,} \\ 0 & \text{else.} \end{cases} \tag{12.1}$$

  This means that in one time step the lattice can only change one of its spins. Which spin is flipped, is selected randomly.

- The second step was to determine whether you will accept the transition to the new state or not. This was determined by the energy. The energy of a state in the Ising model was given by (10.1), and consisted of two parts: one part counted every pair of neighboring spins that were equal or opposite, and the second part counted how many spins were aligned with the external magnetic field and how many were opposite.

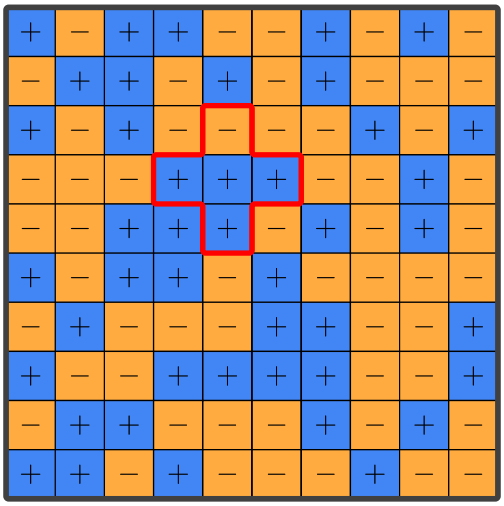
  Whether we accept the transition from state μ to state ν (the flipping of a spin) was determined by the acceptance ratio (11.3):

$$A(\mu \to \nu) = \begin{cases} e^{-(E_\nu - E_\mu)/T} & E_\mu < E_\nu \\ 1 & E_\mu \geq E_\nu \end{cases} \tag{12.2}$$

---

[6] We could also simulate a 3D lattice if we wanted! The algorithm would be very similar, it would only be computationally more expensive since every spin now has 6 neighbors and there are more spins in total.

When calculating the acceptance ratio $A(\mu \to \nu)$ you should in principle calculate the energy of the whole lattice μ, then calculate the energy of the whole lattice ν, and then compare the two. However, since only one spin is flipped in the whole lattice, a very big part of the energy will stay the same. Only the part of the energy which involves the chosen spin (which we will call the **target spin**) will change. Therefore, we will never compute the energy of the whole lattice, but only the part that is relevant for the flip. This relevant part depends on the target spin itself, but also on its neighbors.



+	−	+	+	−	−	+	−	+	−
−	+	+	−	+	−	+	−	−	−
+	−	+	−	−	−	−	+	−	+
−	−	−	+	+	+	−	−	+	−
−	−	+	+	+	−	+	−	+	−
+	−	+	+	−	+	−	−	−	−
−	+	−	−	−	+	+	−	−	+
+	−	−	+	+	+	+	−	−	+
−	+	+	−	−	−	+	−	+	−
+	+	−	+	−	−	−	+	−	−

Figure 7: Spins to consider when calculating the energy for the transition in Figure 3.

The relevant part of the energy will be determined purely by the spins inside the red lines. All other spins will stay the same and are not direct neighbors of the target spin. Let us call the target spin σ^* and the neighboring spins σ_L, σ_R, σ_A and σ_B for the spins to the left, the right, above and below the target spin. In the case of Figure 7 they have the following values:

$$\sigma^* = +1 \qquad \sigma_L = +1 \qquad \sigma_R = +1 \qquad \sigma_A = -1 \qquad \sigma_B = +1. \tag{12.3}$$

The relevant part of the energy is

$$E = -J \cdot \sigma^*(\sigma_L + \sigma_R + \sigma_A + \sigma_B) - h \cdot \sigma^* \tag{12.4}$$

All other parts of the energy will stay the same if you flip the spin.

Chapter IV

Simulating the Ising model in Python

13 A mini introduction to coding in Python

In this part of the course you will use a numerical simulation to study the Ising model in detail. You will use a computer program that is almost finished, but you will have to write a very small piece of code to complete it!

We will be working in a programming language called python. In this section we will briefly discuss the basics of programming in python. This is not a complete course on programming! We just highlight a few simple functions that we are going to use later in the course.

13.1 What is programming?

To most people, a computer is a magic box. It seems like you can use it to do anything! All you need to do is to find the right program, download and install it, and your computer will do exactly what you want it to, and even do things that you might not even be able to do yourself. It seems like a computer has a mind of its own, knows everything and is infinitely smart.

This is, however, not true. A computer is very stupid! The only thing a computer can do is follow a few very simple instructions to the letter. It cannot think for itself, only follow your guidance. However, the magic of computers is that with very simple instructions, you can build complex programs that make your computer seem much smarter than it actually is, even smarter than you! This is called programming. You start with a set of very simple instructions, and keep combining those until you get a big program, like Minecraft or Google Chrome.

In this part of the course, you will do some programming of your own! You will write a few simple instructions for the computer to follow, to complete a program that can simulate the behavior of magnets. The problem is, you don't speak the same language as your computer. For a human, it is very hard (but possible!) to learn the language that the computer actually speaks, it is much too primitive. Also, your computer will never learn English or Dutch like we have. So how will you give your instructions to the computer?

The solution is that you will both learn a language in between! Such an intermediate language is called a programming language. We will use python, since it is relatively simple and user friendly, but still very powerful. You might have heard of other programming languages like Java, C, or even Scratch.

So, your computer does nothing more than follow the instructions you give it to the letter. In a python project, you will create some files of code. If you run (part of) a file, the computer will simply go through this file line per line, and follow your instructions exactly.

13.2 Some very simple features

Let us start with a very easy example: we are going to write a code that consists of only one line. The goal of our code is to make the computer output a few words of text. This is called printing. The command is simply `print`. Your code will look as follows:

```
print('Hello, I am a line of text.')
```

and will print the sentence “Hello, I am a line of text.” in the output window. Note that text in python is called a **string** (like a string of letters), and is always written in between single or double quotation marks.

Now let us do something slightly more complicated. We will define a **variable** x and give it a **value**. We will then calculate the multiple of x with 2 and print the result

```
x = 5
result = x * 2
print(result)
```

We chose 5 as the value of our variable x. We defined a new variable `result`, by using the $*$ symbol for multiplication. We then printed this new variable. Note that there are no quotation marks around result, because we want to print the value of `result` and not the word “result”.

If you want to add comments to your code, you can use the symbol `#`. Anything on the same line after the `#` is not part of the code. It is very useful to use comments to keep a nice overview of what your program does, or as extra information for other people that look at your code.

13.3 Defining functions

A very important notion in programming is that of a function. Very similar to mathematics, you give a function some variable (or sometimes more than one variable), and it returns something. The python language has a lot of different functions built in. We already saw an example of a function, namely the `print` function. another example is the `abs` function. If you have a number `num`, then `abs(num)` will return the absolute value of that number.

Both these examples are built-in functions in python. Sometimes, you want to define new functions that do something specific. Let us see how this works with an example. Suppose we want to define a function that divides a number by 3, and then adds 5 to it. Let us call our function `calculator`. You define it as follows:

```
def calculator(number):
    result = number / 3 + 5
    return result
```

Our function needs one argument. Here we called it `number`, but this name doesn’t matter! Your function defines a variable called `result` according to the formula, and then returns it. Now, whenever you need to calculate $x/3+5$ for some variable x, you can use `calculator(x)` instead.

Some functions have more than one variable. Let us define another function that takes the average of two numbers.

```
    def average(x,y):
        av = (x + y) / 2
        return av
```

Again, the names `x` and `y` do not matter. You could just as easily have written `bibbidy` and `bobbidy` instead.
If you now want to calculate the average of various pairs of numbers, you can use your function `average`:

```
    def average(x,y):
        av = (x + y) / 2
        return av

    print(average(2,5))
    print(average(-4,8))
    print(average(0,6))
```

This code should return 3.5, 2 and 3 in the output window.

13.4 Importing and using packages

When programming, we often heavily rely on the use of **packages**. These are simply extensions to the python language that make your life a lot easier. If a file in your project needs to use a package, you will have to import it at the beginning of your file.

Let us say we need to use a package called `pack`, and it contains three functions called `func1`, `func2` and `func3`. Most packages contain a lot more functions than three, but let us keep it simple in the example. To import the package into your file, write

```
    import pack
```

Sometimes, the package has a very long name, which you would like to abbreviate. In that case, you can write

```
    import pack as pk
```

and now this package is called `pk` in this file.

If you now want to use one of the functions in your code, you should refer to it as follows: You always write the name of the package, a dot (.), and then the function you want. For example, `pk.func1` will give you the function called `func1` that sits in the package `pk`.

Alternatively, if you only need one or a few functions in a package, you can write

```
    from pack import func1
```

in the beginning of your file. Then you can just use `func1` without the `pk.`, but you only have access to `func1` and not `func2` and `func3`.

Example: *Calculating exponentials*
Let us look at an example that you will need later in the lecture. Suppose you want to take the exponential of a number x. You have given x a value previously, and now you want to calculate $y = e^x$. There is a package called `numpy` that has a function called `exp`, which does exactly this. you will see that at the beginning of the file, the package `numpy` is already imported

```
import numpy as np
```

Note that we have renamed **`numpy`** to **`np`** because it is shorter and programmers are lazy.

You can now define your variable y as follows:

```
y = np.exp(x)
```

13.5 Arrays

You often want to store more than one number inside a variable. You can use **arrays** (or **lists**) to do this. An array is simply a collection of numbers. Let us define a very simple array filled with some numbers:

```
ar = [2, 0, 5, 8, 6]
```

We defined a variable called `ar`, which is an array consisting of 5 objects. You always use square brackets to define an array. Now, say we want to extract the first element of `ar` from the fourth and print the result. We can then write the following:

```
ar = [2, 0, 5, 8, 6]
element1 = ar[0]
element4 = ar[3]
result = element4 - element1
print(result)
```

To extract an element from an array, write the name of the array and the position of the element you want to extract in square brackets. You might notice something strange. To extract the fourth element, we wrote `ar[3]` and not `ar[4]`. For the first element we even wrote `ar[0]`! This is because, when it comes to positions in arrays, most programming languages start counting at 0, not at 1! So the first element in an array is actually the zeroth element. Keep this in mind for later.

In the example above, we defined a row of numbers. A one dimensional line. You can also define two dimensional arrays, grids. Instead of inserting one number in the every position of the array, you simply put an array of numbers in every position. We will use the package `numpy` to define higher dimensional arrays. Let us define the following grid of numbers in a variable called `grid`.

```
import numpy as np
grid = np.array([[6, 2, 5], [4, 8, 5], [0, 8, 1]])
```

This will represent the following matrix/grid:

$$\begin{pmatrix} 6 & 2 & 5 \\ 4 & 8 & 5 \\ 0 & 8 & 1 \end{pmatrix} \tag{13.1}$$

Now, suppose I want to extract all the numbers on the second row in the array and print it. Then I write

```
row2 = grid[1]
print(row2)
```

and the result will be a list containing three things: `[4, 8, 5]`.

Suppose I want to extract the number 0 and print it, then I can write

```
nm = grid[2, 0]
print(nm)
```

This means I am looking in column 0 of row number 2, so the first number in the third row, which is 0.

As a final example, we can calculate the sum of all the numbers on the first row:

```
sum = grid[0,0] + grid[0,1] + grid[0,2]
```

The variable `sum` will get the value 13.

13.6 The if statement

To complete the code that simulates the Ising model, we will need one more piece of programming that is slightly more advanced, the `if` statement. This is a conditional statement: it only runs certain pieces of code if some conditions are met. Again, it is easiest to illustrate this with an example. Say we want to define a function that takes the square root of a number. The `numpy` package already has a function that can do this. Only, if your number is negative you will run into problems! This can be solved by using the `if` statement.

We want to define a function called `sq` as follows: if the number is positive or zero, we return the square root using the function `sqrt` from the `numpy` package. If the number is negative, you want to print a message saying “You cannot take the square root of a negative number!” and return zero. Your code would look like this:

```
import numpy as np

def sq(number):
    if number >= 0:
        result = np.sqrt(number)
    else:
        print('You cannot take the square root of a negative number!')
        result = 0

    return result
```

You could also split the function into three different options. Say you want the same function, but if the number is zero you also want to print “Of course the square root of zero is zero!”. You can do this using `elif`:

```
def sq(number):
    if number > 0:
        result = np.sqrt(number)
    elif number == 0:
```

```
            print('Of course the square root of zero is zero!')
            result = 0
        else:
            print('You cannot take the square root of a negative number!')
            result = 0

        return result
```

14 Simulating the Ising model

14.1 Navigating the Toledo folder

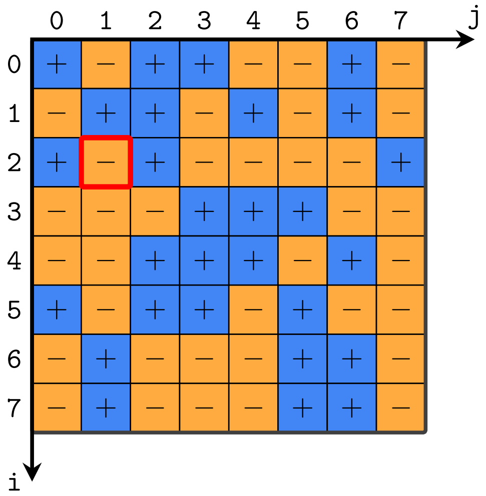
Now that we know the Ising model, Metropolis Monte Carlo simulations and some basic functions in Python, we are ready to make our own simulation of a magnet! We will use the Jupyter notebook file ‘DIY Spin Demo’ on Toledo. Here we will first explain how the code works in general, and in section 14.4 you can find the exercises to complete the program.

When you are done with the exercises, you can open the notebook ‘Full Spin Demo’ on Toledo. Run the full notebook to start the simulation, and follow the instructions in section 14.5.

14.2 The spin lattice as an array in Python

Let us talk about how our model is implemented in Python. We take an $N \times N$ lattice of spins. You will be able to choose N once you run the demo. Our lattice of spins is described by an array of dimensions $N \times N$, filled with $+1$’s and -1’s. Every $+1$ represents a spin up, and a -1 represents a spin down. In the demo, the $+1$’s will be colored blue and the -1’s will be colored orange.

If you want to extract a spin from the lattice, you should do this as follows. First, remember that Python always puts the rows in the first index, and the columns in the second index. Therefore, if you write `lattice[i,j]`, you will get the element of `lattice` that sits in the `i`th row and the `j`th column. To say it differently, the y-component comes first, then the x-component. Also note that you will start counting in the upper left corner (as depicted in the figure below). Finally, remember that Python starts counting at 0, not at 1!



	0	1	2	3	4	5	6	7
0	+	−	+	+	−	−	+	−
1	−	+	+	−	+	−	+	−
2	+	−	+	−	−	−	−	+
3	−	−	−	+	+	+	−	−
4	−	−	+	+	+	−	+	−
5	+	−	+	+	−	+	−	−
6	−	+	−	−	−	+	+	−
7	−	+	−	−	−	+	+	−

Say your lattice that contains the spins is called `lattice`, and it is filled with $+1$’s and -1’s as depicted in the figure above. Say you define

```
target_spin = lattice[2,1]
```

then `target_spin` will get the value -1, since you take the spin in the third row (row number 2) and the second column (column number 1).

14.3 The Metropolis algorithm in action

When the demo starts, your computer will generate an array that is randomly filled with +1's and −1's. It will then use the Metropolis algorithm described in section 12 to update the lattice. It will choose a spin, and decide whether to flip it or not. The demo will then show the new lattice (with one spin flipped or not). This process is called one **time step**. Once the time step is completed, it will use the new lattice as the initial lattice and start again.

The complete python program is fairly long and complicated. However, the most complex part of the program purely has to do with visualizing the demo in a nice window on your screen. The actual Metropolis algorithm is very short and simple! The important part looks like this:

```
def simulation(self):
    for i in range(int(self.params['Steps'])):

        temperature = self.params['Temperature']
        H = self.params['Magnetic field (H)']
        J = self.params['Interaction (J)']

        targetpos = np.random.randint(0,self.gridsize,2)

        energy = func.compute_energy(J, H, targetpos,
                                     self.array,
                                     self.gridsize
                                     )

        self.array = func.spinflip(self.array, targetpos, energy, temperature)

        if self.plot:
            nspins = self.gridsize**2
            self.mag.append(abs(np.sum(self.array) / nspins))

    return self.array
```

Let us go through the different steps that this program makes during one time step:

1. Choose the position of the target spin in the lattice at random, by generating two random numbers describing its x- and y-coordinates.

   ```
       targetpos = np.random.randint(0, self.gridsize, 2)
   ```

   This will be stored as an array called `targetpos` containing first the y-coordinate and then the x-coordinate.

2. Determine whether the target spin will be flipped or not. This step will be divided into two parts. First we compute the relevant part of the energy

   ```
       energy = func.compute_energy(J, H, targetpos,
                                    self.array,
                                    self.gridsize
                                    )
   ```

   Then we determine whether or not to flip the spin and update the lattice

```
    self.array = func.spinflip(self.array,
                               targetpos,
                               energy,
                               temperature
                               )
```

The function **`func.spinflip`** will do the following (schematically):

- Will flipping the spin lower the energy? Then flip the spin and return the updated lattice
- Will flipping the spin raise the energy? Then calculate the Boltzmann factor

$$\texttt{boltzmann\_factor} = e^{-(E_\nu - E_\mu)/T}. \tag{14.1}$$

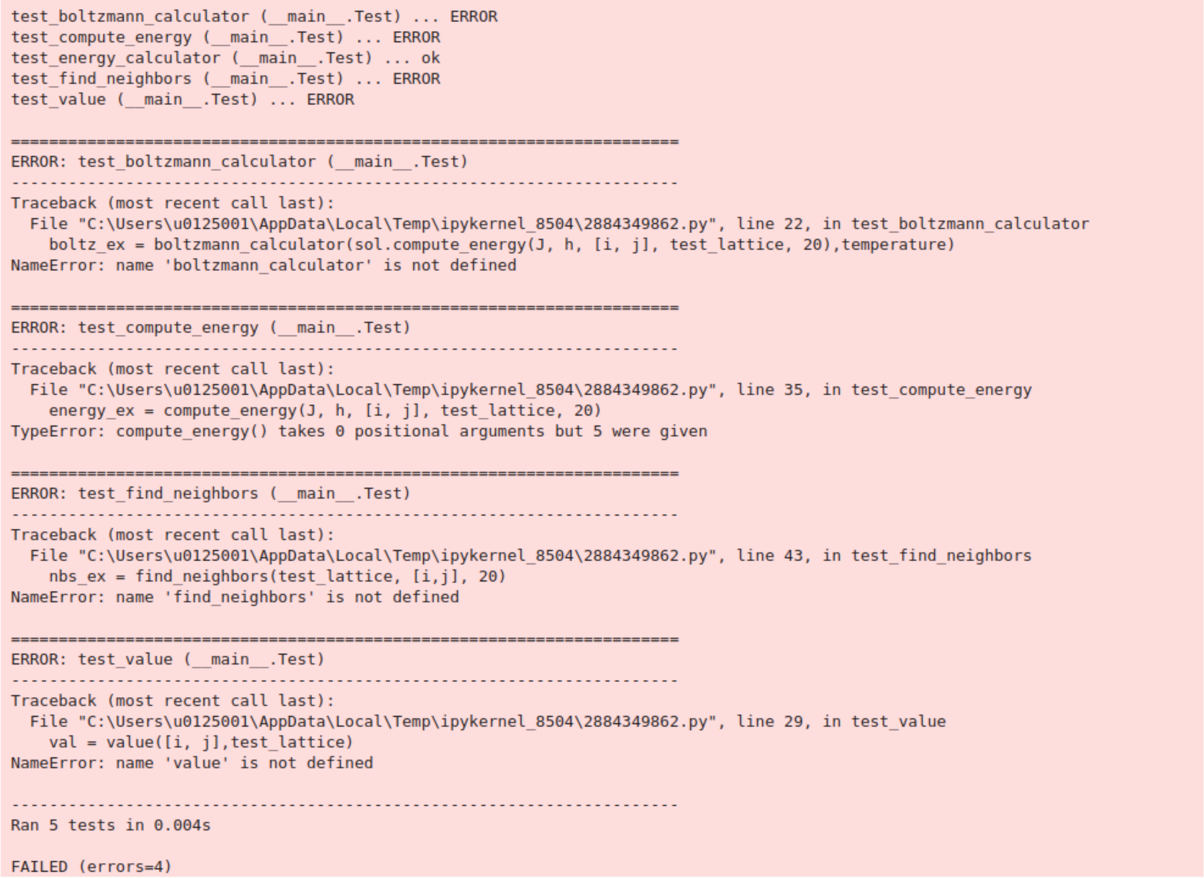
  Since **`boltzmann_factor`** is a positive number between 0 and 1, you can use it as a probability. With this probability, either flip the spin and return the updated lattice or just return the current lattice.

You will write the functions **`compute_energy`** and **`spinflip`** yourself in the exercises below.

14.4 DIY: Completing the Metropolis algorithm!

With all the information above, you are finally ready to start coding the Metropolis algorithm yourself! You will write a few functions that are essential to get the algorithm running. To start, go to 'DIY Ising Demo' on Toledo. You will see that this file is already divided into six different parts, for the six functions you will write. For each function, you will get a description of what it should do in the exercises below.

When you think you have found one of the functions, you can run the cell in section 2. This cell will test if your functions work correctly. Once you run the file, the output will look like this:



```
test_boltzmann_calculator (__main__.Test) ... ERROR
test_compute_energy (__main__.Test) ... ERROR
test_energy_calculator (__main__.Test) ... ok
test_find_neighbors (__main__.Test) ... ERROR
test_value (__main__.Test) ... ERROR

======================================================================
ERROR: test_boltzmann_calculator (__main__.Test)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\u0125001\AppData\Local\Temp\ipykernel_8504\2884349862.py", line 22, in test_boltzmann_calculator
    boltz_ex = boltzmann_calculator(sol.compute_energy(J, h, [i, j], test_lattice, 20),temperature)
NameError: name 'boltzmann_calculator' is not defined

======================================================================
ERROR: test_compute_energy (__main__.Test)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\u0125001\AppData\Local\Temp\ipykernel_8504\2884349862.py", line 35, in test_compute_energy
    energy_ex = compute_energy(J, h, [i, j], test_lattice, 20)
TypeError: compute_energy() takes 0 positional arguments but 5 were given

======================================================================
ERROR: test_find_neighbors (__main__.Test)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\u0125001\AppData\Local\Temp\ipykernel_8504\2884349862.py", line 43, in test_find_neighbors
    nbs_ex = find_neighbors(test_lattice, [i,j], 20)
NameError: name 'find_neighbors' is not defined

======================================================================
ERROR: test_value (__main__.Test)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\u0125001\AppData\Local\Temp\ipykernel_8504\2884349862.py", line 29, in test_value
    val = value([i, j],test_lattice)
NameError: name 'value' is not defined

----------------------------------------------------------------------
Ran 5 tests in 0.004s

FAILED (errors=4)
```

You will see here which of your functions are still unfinished or are not working properly. In the example here, there is one function that works properly, namely **`energy_calculator`** (it does not appear in the list of unfinished functions). You can see that **`boltzmann_calculator`**, **`value`**, **`compute_energy`** and **`find_neighbors`** are not working properly. The function **`spinflip`** is never tested. The description of the error can give you a hint of what you did wrong in your code.

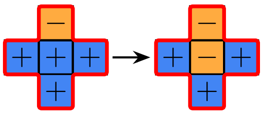
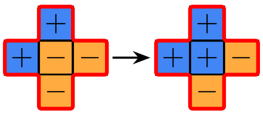
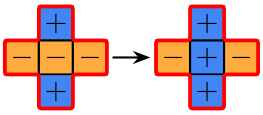
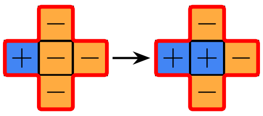
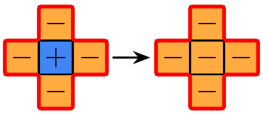
Exercise 0: preparations

Before you start writing the functions, first do the following little exercise:
In the table below you can see a few possible transitions from a state μ to a state ν (spin flips). We only show the targeted spin with its direct neighbors. For each of the transitions, write down the relevant parts of the energy E_μ and E_ν. Do you notice anything interesting when you compare the two energies?

. . . .

Next, write down the difference in energy and determine whether the spin will automatically be flipped, or you need to use the Boltzmann factor. Here you can assume that $h = 0$. Do you notice a pattern? Can you see the condition for automatically accepting?

. . . .

Transition from μ to ν	E_μ	E_ν	$E_\mu - E_\nu$	Auto-accepted? ($H = 0$)
				
				
				
				
				

Exercise 1: energy_calculator

Define a function, called `energy_calculator`, that takes the following input:

- The parameters of your model (J and h)
- The value of the targeted spin
- The values of the neighboring spins

The function should return the relevant part of the energy of this configuration (only the part that changes if you flip the spin).

Exercise 2: boltzmann_calculator

Define a function, called `boltzmann_calculator`, that takes the following input:

- The initial energy of the targeted spin (with its neighbors)
- The temperature

The function should return the Boltzmann factor for flipping the spin.

Hint: Use what you wrote down in the first part of Exercise 0. Be smart about your code and try to calculate as little as possible!
Hint: you will need to use the `numpy` package.

Exercise 3: value

Define a function, called `value`, that takes the following input:

- The position of the targeted spin (two numbers between 0 and the size of your lattice minus 1)
- The full spin lattice (as an array)

The function should return the value of the targeted spin.

Exercise 4: compute_energy

Define a function, called `compute_energy`, that takes the following input:

- The parameters of your model (J and h)
- The position of the targeted spin
- The full spin lattice (as an array)
- The dimension of the lattice

The function should return the energy of the spin and its neighbors.

You can use a function called `find_neighbors` that takes the following input:

- The full spin lattice (as an array)
- The position of the targeted spin
- The dimension of the lattice

It gives you the spin values of the neighbors of the targeted spin as a list with 4 entries. You will define this function in a later problem.

Hint: you might want to use the functions `energy_calculator`, `boltzmann_calculator` and `value` from Exercises 1, 2 and 3.

Note that the order of the input parameters is important here, because this function is used in the main program.

Exercise 5: spinflip

Define a function, called `spinflip`, that takes the following input:

- The full spin lattice (as an array)

- The position of the targeted spin
- The energy of the spin and its neighbors
- The temperature

The function should determine whether or not to flip the targeted spin.

Using the Metropolis algorithm, return a new array that is the inputted lattice, but with the spin flipped if necessary.

Hint: Use what you wrote down in the second part of Exercise 0. Be smart about your code and try to calculate as little as possible!
Hint: you might want to use the functions `boltzmann_calculator` and `value` from Exercise 3.
Hint: you should use the `if` statement to differentiate between the different cases in the Metropolis algorithm.

Note that the order of the input parameters is important here, because this function is used in the main program.

Exercise 6: find_neighbors

Define a function, called `find_neighbors`, that takes the following input:

- The full spin lattice (as an array)
- The position of the targeted spin
- The dimension of the lattice

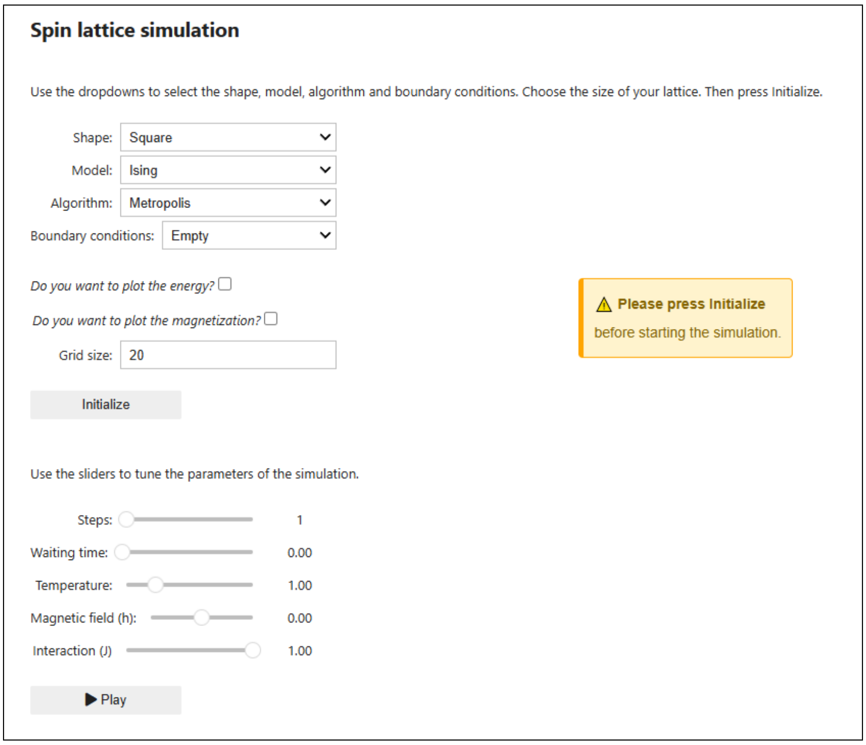
The function should return the values of the neighboring spins.

Note that you need to be careful in the special case when the targeted spin is on the edge of the lattice. In that case, it has only 3 neighbors (or 2 if it is in a corner) since we are working with empty boundary conditions. The missing neighbor should have a spin value of 0 instead of 1 or -1.

Hint: use the `if` statement to differentiate between the special cases and the usual case.

14.5 Exploring the Ising model with your simulation code

If you have completed all the exercises above, your code is finished and you are now ready to start the demo! Do not worry if you did not manage to find all the functions, the program will take the unfinished functions from the solutions. To start the demo, go to Toledo and open the jupyter notebook under 'Full Spin Demo'. Run all the cells. The following applet should appear:



Spin lattice simulation

Use the dropdowns to select the shape, model, algorithm and boundary conditions. Choose the size of your lattice. Then press Initialize.

Shape: Square
Model: Ising
Algorithm: Metropolis
Boundary conditions: Empty

Do you want to plot the energy? ☐
Do you want to plot the magnetization? ☐
Grid size: 20

⚠ **Please press Initialize** before starting the simulation.

Use the sliders to tune the parameters of the simulation.

Steps: 1
Waiting time: 0.00
Temperature: 1.00
Magnetic field (h): 0.00
Interaction (J) 1.00

The dropdown menus and checkboxes at the top of the demo will be used later. For now, do not change these. Choose how big you want to make your lattice. This is entirely up to you, although your grid size should be a positive integer. Once you have entered your grid size, press 'Initialize'. The demo will now generate a random lattice of up and down spins with your chosen size. You are now ready to start your simulation. You can use the sliders to change the following parameters:

- **Steps:** Choose how many times you want to go through the Metropolis algorithm before you update the lattice in the window. The default value is 1, meaning that every time you try to flip a spin, you immediately show the new lattice. The parameter Steps can go up to 100, where you will try to flip 100 spins before you update the picture.
- **Waiting time:** This parameter controls the speed of the demo. If Waiting time is put to zero, the program will update the lattice as fast as it can. If you want to study the simulation in detail, this might be a bit too fast. You can therefore increase Waiting time to add a pause in between each update. If you increase Waiting time to 1.5, your computer will wait for 1.5 seconds after each update, allowing you some time to study the lattice before it updates again.
- **Temperature:** Choose the temperature of your system. You can choose values between 0.1 and 4.0. The upper bound is of course arbitrary, in theory you could choose any (positive) value for the temperature.
- **Magnetic field (h):** This parameter controls the value of h in your model, which describes the external magnetic field. It can take values between -1.0 and 1.0, allowing you to turn on either an "up spin magnetic field" or a "down spin magnetic field".
- **Interaction (J):** This parameter controls the value of J in your model, describing the strength of the interaction between neighboring spins. It can go between 0.0 and 1.0. In theory you could also go higher, but we limit J to 1.0 for now.

You can move the sliders before the demo starts or when it is paused, but also while it is running. You can use the Play/Pause button to start and pause the demo. To restart with a new random lattice, press ‘Initialize’ again.

You can now start exploring the Ising model! You are free to try out things as you please, but here are a few suggestions:

Exercise 1: Influence of the temperature

You can start by keeping the original values of J and h:

$$J = 1 \qquad \text{and} \qquad h = 0$$

and only varying the temperature. What do you expect to happen at very high temperatures? And at very low temperatures?

....

Try this out with the demo. Do the results follow your expectations? There should be a turning point at around $T = 2.269\ldots$. This turning point is called the critical temperature. It distinguishes between the two phases of the Ising model. When the temperature is below the **critical temperature**, the Ising model describes a **ferromagnet**: the whole lattice should eventually be filled with only one color. This might take some time! Try to put the temperature very low, and wait until one color takes over. You should have an equal chance of becoming entirely orange or entirely blue.
Hint: The bigger your lattice, the longer you will have to wait. Start with a small lattice.

At temperatures above the critical temperature, you go into the **paramagnetic phase**. The temperature is high enough so that every spin has the possibility to flip, regardless of its neighbors. This results in a lattice that looks completely random.

Exercise 2: Adding an external magnetic field

Next, you can start playing with the slider for the external magnetic field. What do you expect to happen when you put the slider on 1.0? And on -1.0?

....

Choose a very low temperature and turn on the magnetic field. Does the simulation follow your expectations? Now put the temperature at a higher value. Can you still see the effect of the magnetic field at high temperatures? Why is the effect less visible then?

....

Exercise 3: Changing the interaction strength

Now we will keep h at its original value $h = 0$, but we will vary J. When you change J, you will change how much the spins interact with their neighbors. If you make J smaller than 1, the spins will find it less important to align with their neighbors. This will influence the critical temperature. If you lower J, what do you expect to happen to the critical temperature? Will it now be lower or higher than $T = 2.269\ldots$?

. . . .

Try it out! can you find the new critical temperature when you put J at a chosen value? Write down the value of J that you picked, and your guess for the critical temperature.

. . . .

For $h = 0$, you can prove a formula for the critical temperature:

$$T_c = \frac{2J}{\ln\left(1+\sqrt{2}\right)} \tag{14.2}$$

Put your chosen value of J into this formula and calculate T_c. Was your guess for the critical temperature close?

. . . .

Exercise 4: Equilibration and total magnetization

Finally, we will study how long it takes for the system to reach its equilibrium. When you press 'Initialize' and load a random lattice of spins, this is not the preferred state for the system at any value of T. The system will need some time before it reaches its equilibrium.

To study this, we will look at the magnetization per spin of the lattice. This was defined in section 10 as

$$m = \frac{M}{N^2} = \frac{N_+ - N_-}{N^2}$$

(see equation (10.3)). You sum up all the spins and divide by the number of spins. This gives you the "average value of a spin", or the magnetization of the lattice per spin (also called the relative magnetization).

When the lattice is completely filled with up-spins, what do you expect the value of m to be?

. . . .

When the lattice is completely random (like when you just pressed 'Initialize'), what do you expect the value of m to be?

. . . .

Using methods from statistical physics, you can prove that for the Ising model the magnetization per spin is given by

$$|m| = \left[1 - (\sinh(2J/T))^{-4}\right]^{1/8}$$

but only for $h = 0$ (see equation (10.4)). So, for any value of the parameters J and h, you know what the relative magnetization should be at a given temperature.

Suppose you choose $h = 0$, $J = 1$ and $T = 0.5$. The relative magnetization should then be equal to ± 0.99999977462, so very close to ± 1. However, when you set up your simulation and press 'Initialize', the demo generates a random lattice and the relative magnetization will probably be close to zero. You then start the demo, and it will need some time to evolve towards the equilibrium state. This might take a lot of time steps! The amount of time steps necessary to reach equilibrium is called the **equilibration time** τ_{eq}.

To test this, you will need to check the magnetization checkbox at the top of the demo before starting. You will now plot the relative magnetization as a function of the time step after you close the demo. For low temperatures, it will look something like this when you let it run for a while:

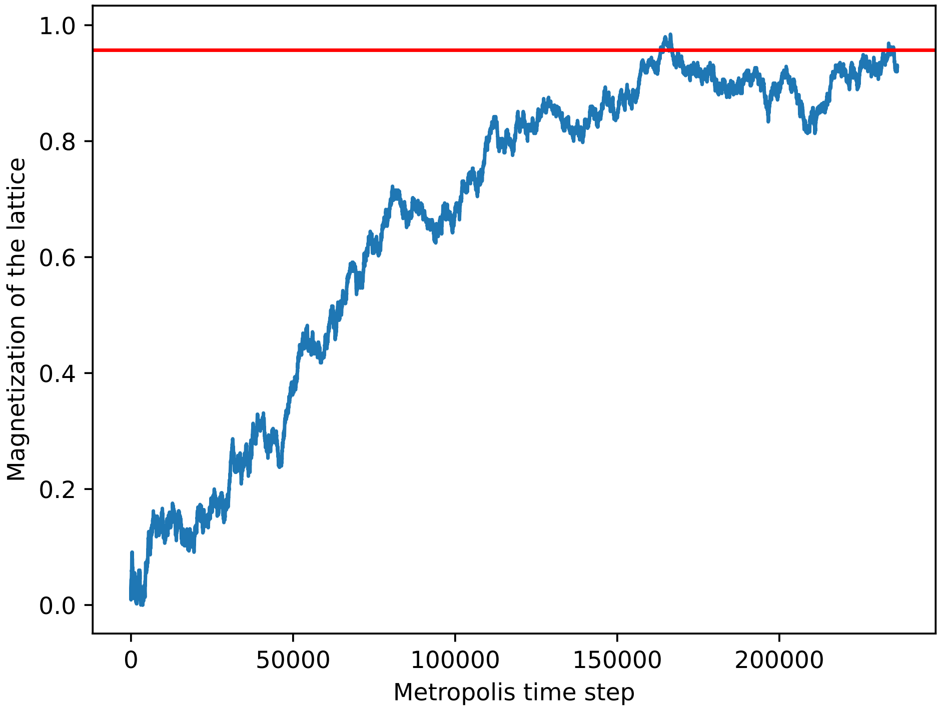


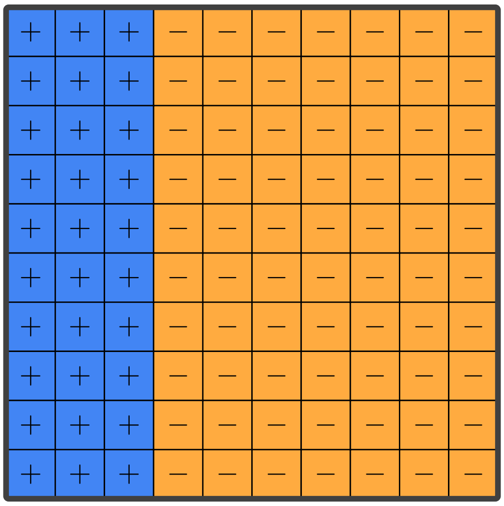
Figure 8: Relative magnetization as a function of the time step for a simulation of a 30×30 lattice with $J = 1$, $h = 0$ and $T = 1.80$. After around $\tau_{eq} = 160\,000$ time steps the system has reached equilibrium. The horizontal line shows the analytic value of $|m|$

You can see that the relative magnetization starts out around zero since the lattice is random initially. It then takes about 160 000 time steps for the simulation to reach equilibrium.

Try this out for yourself! How long does it take for your simulation to reach equilibrium? A few things to take into account:

- The bigger you make your lattice, the longer it will take. It is best to start out with a lattice size of 20 or smaller. Later, you can also try larger lattices if you want.
- It is best to put the slider 'Steps' to 100. Otherwise, you will have to wait for a very long time.

- The lower the temperature, the longer it will take before you reach equilibrium. This phenomenon is called **critical slowing down**. Also, at very low temperatures the simulation can also get stuck in a situation like this:



+	+	+	−	−	−	−	−	−	−
+	+	+	−	−	−	−	−	−	−
+	+	+	−	−	−	−	−	−	−
+	+	+	−	−	−	−	−	−	−
+	+	+	−	−	−	−	−	−	−
+	+	+	−	−	−	−	−	−	−
+	+	+	−	−	−	−	−	−	−
+	+	+	−	−	−	−	−	−	−
+	+	+	−	−	−	−	−	−	−
+	+	+	−	−	−	−	−	−	−

A straight interface forms, and at very low temperatures this will never move. Can you explain why it will get stuck? Do you have any idea how to get rid of this problem?

....

Chapter V

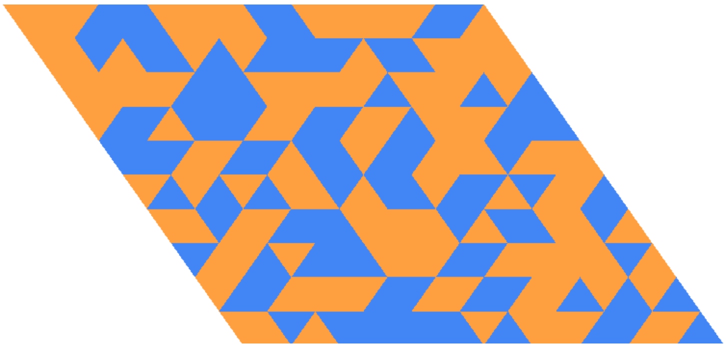
Extensions to the Ising model

You have now fully explored and understood how the standard 2D Ising model works if you simulate it using a basic Metropolis Monte Carlo method. You have already explored the demo on its initial settings, but it is capable of much more than only the Ising model and the Metropolis algorithm! You can simulate a few variations on the Ising model, which have interesting new behavior. Below you will find a description of all the variations you can choose from, and you can even combine all of them!

15 Varying the shape of the lattice

15.1 The triangular 2D Ising model

Until now, your spin lattice was always a square lattice, meaning that each spin is represented by a square in the lattice and has 4 different neighbors. Using the first dropdown menu in the demo, you can also choose to simulate a triangular lattice. Now, each spin is a triangle and therefore has only 3 neighbors.



Funnily enough, changing the shape of the lattice does not have much influence on the model according to the theory. The Ising model will still undergo a phase transition from ferromagnetic to paramagnetic. The critical temperature is slightly different. For the triangular Ising model, the critical temperature is

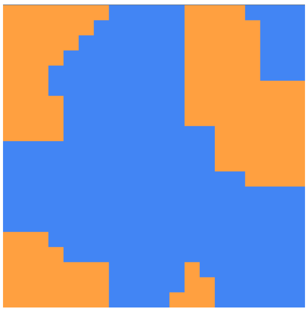
$$T_c = \frac{2J}{\ln\sqrt{3}} \quad = 3.64\ldots\,(\text{for } J = 1) \tag{15.1}$$

So in theory, a triangular lattice should behave very similarly to a square lattice. However, the simulation does suffer from some computational problems. If you run the demo, you will see that something funny happens at low temperatures. If you made Exercise 4 of section 14.5, you know that at low temperatures the Ising model should find an equilibrium where one of the two colors has almost completely taken over. This might take some time, as you saw in the exercise, but eventually this will always happen.

Now try to do the same for the triangular lattice. Take $h = 0$ and $J = 1$ and put the temperature very low. Wait to see what happens. What do you observe?

....

To explain what happens here, we will compare an interface between blue and orange spins for both types of lattice. The figures below are examples of lattices where the demo has started with a random lattice and run for some time at a low temperature.



Take a look at an interface of the square lattice. There are always corners to these interfaces (unless one half of the grid is blue and the other half orange, but that almost never happens). At such a corner, the spin has two blue neighbors and two orange ones. Therefore, if it flips its energy will stay the same (for $h = 0$ only). Such a flip will then always be accepted, creating even more corners which can then flip as well. The interface will keep moving like that, until one color has completely taken over.

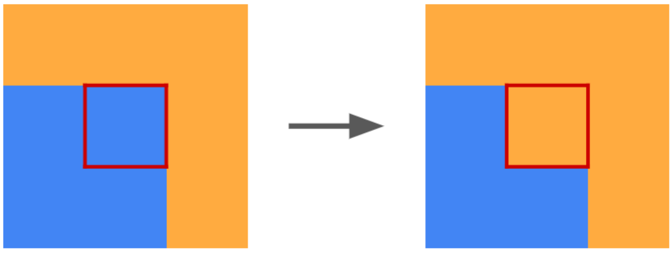
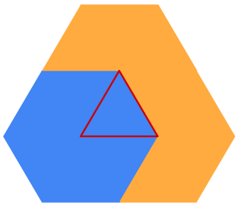
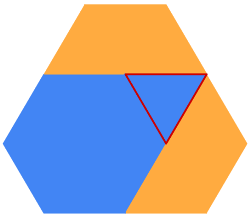


Figure 10: For a square lattice, a spin flip at a corner of an interface will always be accepted, allowing the interface to evolve and the lattice to reach equilibrium.

For a triangular lattice, all spins at an interface are stuck at low temperatures. Take a spin at an interface, even in a corner. You can see that it will always have two neighbors with the same color and one with another color (see left figure below). At very low temperatures it will therefore never flip, and the interface is stuck. It cannot move.





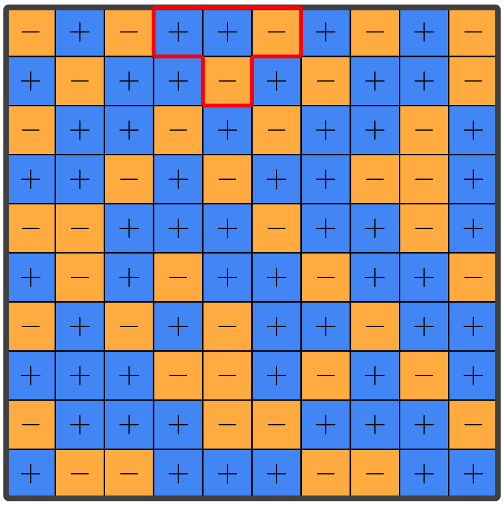
The only spins that will be flipped are like the ones in the right figure, but once these are all gone, your interfaces will never move.
You might have noticed that the spins at the edge of the lattice do move sometimes. This is because they have only two neighbors instead of three. Therefore, if they are at an edge between blue and orange, they have one blue neighbor and one orange one, so they can be flipped at any time.

We can conclude that, for the triangular Ising model, the equilibrium behaviour is exactly the same as for the square Ising model. The big difference is that, when you start from a random lattice, the simulation will never reach equilibrium at low temperatures. The phenomenon of critical slowing down is extra strong here. In section 18.2 you will learn a different simulation algorithm that solves the problem of critical slowing down.

16 Varying the boundary conditions

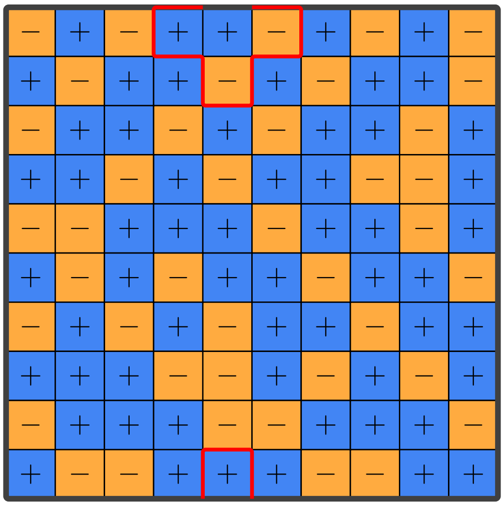
So far, our lattice always had empty boundary conditions, as explained earlier. We now include the option to choose three different types of boundary conditions. We list them here, and provide an example lattice with a target spin at the edge as an illustration.

- **Empty:** These were the boundary conditions that we used before. A spin at the edge of a square lattice only has three neighbors, not four, and a spin in the corner of a square lattice only has two neighbors.



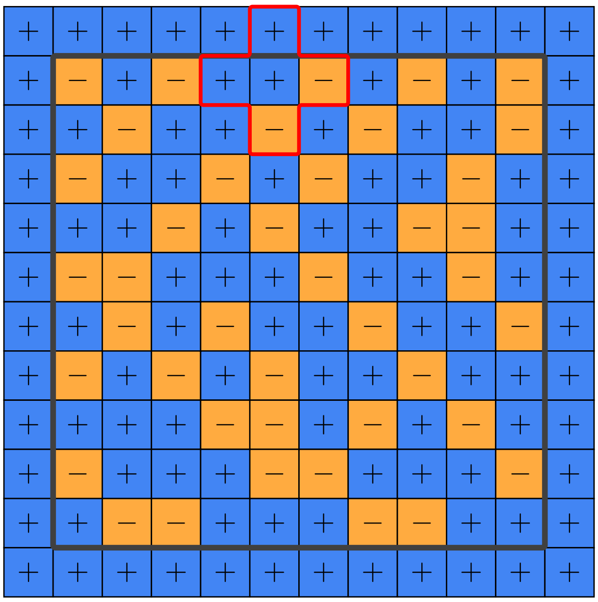
−	+	−	+	+	−	+	−	+	−
+	−	+	+	−	+	−	+	+	−
−	+	+	−	+	−	+	+	−	+
+	+	−	+	−	+	+	−	−	+
−	−	+	+	+	−	+	+	−	+
+	−	+	−	+	+	−	+	+	−
−	+	−	+	−	+	+	−	+	+
+	+	+	−	−	+	−	+	−	+
−	+	+	+	−	−	+	+	+	−
+	−	−	+	+	+	−	−	+	+

- **Periodic:** These boundary conditions essentially get rid of the boundary completely by closing the lattice on itself. The neighbor above a spin at the top of the lattice is the one in the same column, all the way at the bottom, etc.



−	+	−	+	+	−	+	−	+	−
+	−	+	+	−	+	−	+	+	−
−	+	+	−	+	−	+	+	−	+
+	+	−	+	−	+	+	−	−	+
−	−	+	+	+	−	+	+	−	+
+	−	+	−	+	+	−	+	+	−
−	+	−	+	−	+	+	−	+	+
+	+	+	−	−	+	−	+	−	+
−	+	+	+	−	−	+	+	+	−
+	−	−	+	+	+	−	−	+	+

- **Fixed:** With fixed boundary conditions, the spins at the edge of the lattice each have an "imaginary neighbor" that has a fixed value. As you can see in the figure below, the lattice is surrounded by an extra ring of spins. They are not part of the simulation, but only act as neighbors for the spins at the edge. Here we choose the same fixed value for all the "imaginary neighbors": they are all up spins (+1).



+	+	+	+	+	+	+	+	+	+	+	+
+	−	+	−	+	+	−	+	−	+	−	+
+	+	−	+	+	−	+	−	+	+	−	+
+	−	+	+	−	+	−	+	+	−	+	+
+	+	+	−	+	−	+	+	−	−	+	+
+	−	−	+	+	+	−	+	+	−	+	+
+	+	−	+	−	+	+	−	+	+	−	+
+	−	+	−	+	−	+	+	−	+	+	+
+	+	+	+	−	−	+	−	+	−	+	+
+	−	+	+	+	−	−	+	+	+	−	+
+	+	−	−	+	+	+	−	−	+	+	+
+	+	+	+	+	+	+	+	+	+	+	+

Of course, it makes a difference what values you choose for the fixed boundary conditions. What do you expect to happen to the lattice at low temperatures for these fixed boundary conditions? Why is this different than for the other two types of boundary conditions?

. . . .

17 Varying the model

The Ising model describes a lattice filled with spins which have two possibilities: up or down, $+1$ or -1. This model can describe certain types of magnets, but can also be used to simulate other phenomena! For example, you can use it to describe the evolution of language use in countries with two languages. It can be applied to many different things like voting behavior, IT network anomalies and economy.

Below we will discuss some variations of the Ising model. Although they are not all suited to describe magnets, they have many other applications similar to the ones above.

17.1 The Potts model

A very simple extension of the Ising model is called the Potts model. Every spin now has more than two choices. In a Potts model with 4 possible spins, every spin can now choose between the values 1, 2, 3 and 4. Every value gets its own color.

The energy of a spin lattice now has a similar but slightly different formula[1]:

$$E = -J \sum_{<ij>} \delta(\sigma_i, \sigma_j) \tag{17.1}$$

You can see that the only difference is the δ. You should interpret this as follows: As usual, we take a spin and look at its neighbors. If the neighbor has the same color, the delta function will spit out 1, and a $-J$ is added to the energy (so the energy is lowered). If the neighbor has a different color, the delta function gives 0, and nothing is added to the energy.

You can see that the Potts model behaves very similarly to the Ising model: Spins want to have the same color as their neighbors, but if the temperature is high enough they also get the tendency to flip randomly to another color. At low temperatures, one of the colors will take over. At high temperatures, the lattice will be a random mixture of different colors.

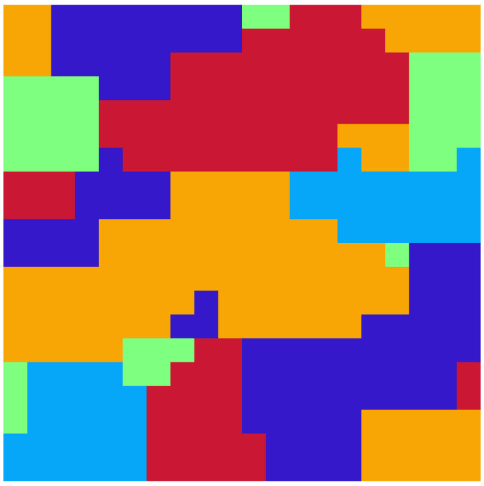


Figure 12: Simulation of the Potts model with 5 different colors. Paused after a while, for $T = 0.1$.

When you open the demo, choose a square grid and choose 'Potts' for the model. An input bar

---

[1] For the Potts model we will not add an external magnetic field, although in principle you could make it so that one of the colors is preferred over the others. The slider for the magnetic field does not do anything if you select the Potts model.

will appear where you can choose the number of different colors. You can play around a bit with the demo. Do you see the same behavior as for the Ising model?

……

Run the demo for different values for the number of colors (call q the number of colors). Can you find the critical temperature for the different values of q (at $J = 1$)?

……

You can compare your findings to the analytic value for the critical temperature:

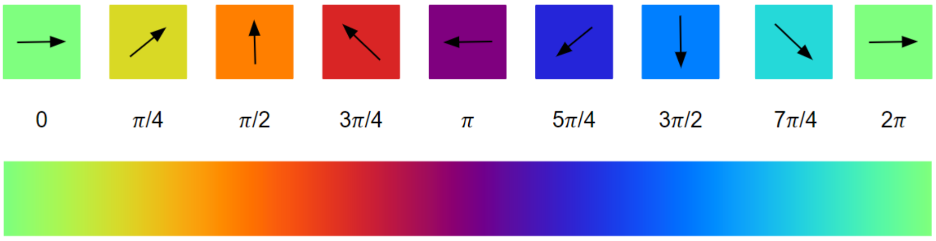
$$T_c = \frac{J}{\ln(1+\sqrt{q})} \tag{17.2}$$

where q is the number of different colors. Do your critical temperatures agree with this formula?

……

17.2 The XY-model

We can go even further than the Potts model, and allow not just a number of discrete values for the spins, but now allow for continuous spins! Every spin can now point in a random direction, not just in a few possible directions. You can think of every spin as an arrow, which can be held under any angle. The value of the spin can go from 0 (to the right) all the way around to 2π which is again to the right. See the figure below as an illustration.



Each direction is given a color, as you can see above. In the demo, only the colors will appear and not the arrows. Every spin is now characterized by an angle θ_i, which can go from 0 to 2π. The energy of a lattice is given by

$$E = -J \sum_{<ij>} \cos(\theta_i - \theta_j) - H \sum_i \cos(\theta_i) \tag{17.3}$$

The more two neighboring spins point in the same direction, the lower their interaction energy will be. If you decide to add a positive external magnetic field, the spins will want to point to the $\theta = 0$ direction (green in the demo). If you dial h to -1, the spins will prefer to become purple.

Unlike the Ising and Potts models, the XY-model does not have a phase transition from ferromagnetism to paramagnetism. There will be no temperature where the behaviour will suddenly change from ordered spins (all trying to point in the same direction) to disordered spins (pointing in random directions). Try this out for yourself: what do you see happening at very low temperatures?

. . . .

Instead, there is another, very special phase transition. It is called the Kosterlitz-Thouless transition after the people who discovered it. They even got a Nobel prize for this discovery in 2016! This transition is much more complicated than the phase transition of the Ising model. at high enough temperatures, the spins in the XY-model will automatically form something called **vortices** and **anti-vortices**. These are like little whirlwinds of spins pointing in different directions. They look something like this:

Figure 13: A vortex (left) and an antivortex (right).

If you plot the colors of the corresponding spins, they would look something like this:

Figure 14: A vortex (left) and an antivortex (right) with colors instead of arrows. As you can see, it is very hard to spot the difference between the two when you only look at the colors.

At high temperatures, the vortices and antivortices can exist on their own, and move freely around the lattice. If you then lower the temperature, you will pass a critical temperature. Below that, the vortices and antivortices will start moving ever closer towards each other. Once they meet, a vortex and an antivortex will annihilate, disappear together. The critical temperature is around $0,89$ for $J = 1$ and $h = 0$.

Since the demo only plots colors and not arrows, it is very hard to spot vortices and antivortices. Therefore, when you pause the demo, it will always show the current state of the lattice with arrows. Note that this does not work for a triangular grid. A first thing you can do is to let the simulation run for a while on a very low temperature and then pause the demo. Can you spot (anti)vortices? How many did you find?

. . . .

If you want to try and observe the phase transition yourself, please choose periodic boundary conditions and take a large enough lattice size (> 15). Try to let the simulation run at temperatures just above the critical temperature, and then pause the program and count the vortices and antivortices. Do not put the temperature too high, because then you won't be able to see them. How many (anti)vortices do you see, and are they close together?

....

Now start up the demo again with the same options, and put the temperature just below the critical temperature. Don't put the temperature too low, otherwise the spins will get stuck and nothing will move. Let the demo run a while, pause it and count the number of vortices and antivortices. How many do you see, and are they close together now?

....

Do not be discouraged if you cannot see the phase transition happening. It is very hard to spot with this demo!

If you want to know more about the Kosterlitz-Thouless transition and the 2016 Nobel prize, here is a nice video you can watch on YouTube:
`https://www.youtube.com/watch?v=aGNGl-sQzNA&t=159s&ab_channel=simpleshowfoundation`

18 Varying the simulation algorithm

18.1 The Heat bath algorithm

In the basic Metropolis algorithm that we used before, the decision whether to flip a spin was made as follows:

- Will flipping the spin lower the energy? Then flip the spin.
- Will it raise the energy? Then flip it with the following probability:

$$P(\text{flip}) = e^{-\Delta E/T} \tag{18.1}$$

The Heat bath algorithm is very similar, but uses a slightly different flipping probability: it does not matter whether the energy is raised or lowered, you always flip with probability

$$P(\text{flip}) = \frac{e^{-\Delta E/T}}{1 + e^{-\Delta E/T}} \tag{18.2}$$

If you compare this probability with the one for basic Metropolis, it looks like this as a function of ΔE:

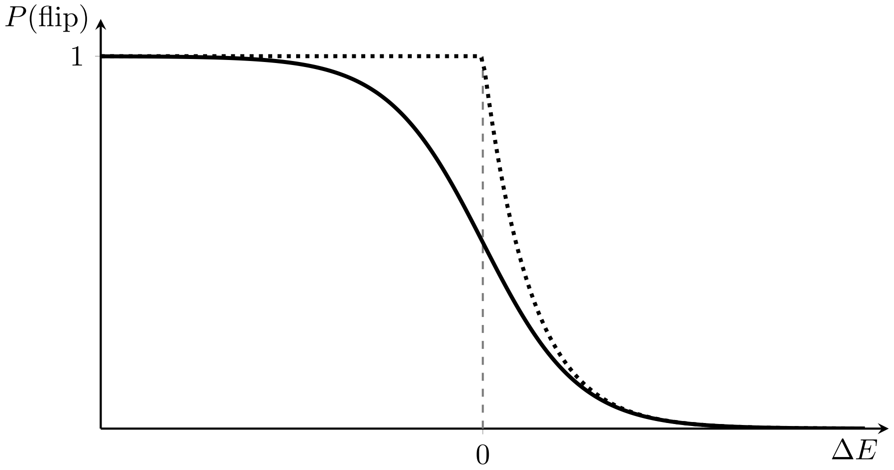


Figure 15: comparison of the flipping probability (acceptance ratio) for basic Metropolis (dotted line) vs. Heat bath algorithm (full line)

If you run the demo with this algorithm, you will see that not much has changed compared to the basic Metropolis algorithm. Only when simulating the Potts model with a large number of colors will you notice a difference: The Heat bath algorithm is much more efficient in that case.

18.2 The Wolff algorithm

Where the Heat bath algorithm was very similar to the basic Metropolis one, the Wolff algorithm is very different. In this algorithm, you will not flip one spin every time step, but a whole cluster of spins! The average size of the cluster that flips depends on the temperature: at low temperatures you will flip very large clusters, at high temperatures the clusters will be very small.

The algorithm works as follows:

- Select one spin at random
- Look at the neighbors of this spin. If they have the same color, add them to the cluster with probability
$$P(\text{add}) = 1 - e^{-2J/T} \tag{18.3}$$
- Now look at all the neighbors of the cluster. Repeat the same process until you have added no more spins to the cluster.
- Always flip the whole cluster.

The Wolff algorithm solves the issue of critical slowing down: your system will reach equilibrium much faster than with the other two methods. You can try this yourself: how many time steps does it take for the Ising model at $T = 0,1$ to reach equilibrium with this method. (Tip: keep the slider 'Steps' on 1 and put the slider 'Waiting time' on $1,0$ so that the demo always waits one second to show the next state. Otherwise you won't be able to follow.)

……

Note that this Wolff algorithm only works for $h = 0$, so put the slider for the external magnetic field to zero.

Chapter VI

Magnetic Storage and Computer Memory

19 Introduction

Humans have been storing information on magnetic mediums for more than 200 years. In fact, in 1899 magnetic audio recording was achieved by magnetising a thin steel wire with an electrical wave that was generated by the incoming sound wave. The real developments of magnetic recording and computer memory occurred in the late 1930s and primarily after the end of the Second World War with the development of magnetic tape storage and core memory. Over the years, these technologies evolved, new magnetic technologies were developed and some were discontinued. We do not intend to discuss all magnetic memory technologies that ever existed, but we will focus on four of them.

We can make a distinction of two primarily categories of magnetic memory:

1. Magnetic storage
    - Made for long term storage of information that we don't need to access frequently
    - Data is stored on a tape or disk
    - Data location is found by moving the tape or disk head
    - Very cheap to store data, but slow to read it
    - *Examples: magnetic tape and hard disk drive (HDD)*
2. Magnetic computer memory
    - Made for storage of small amount of data that the computer needs to access very frequently
    - Data is magnetic ring or thin film
    - Every bit can be accessed randomly by only applying currents (No moving parts!)
    - More expensive to store data, but very fast
    - *Examples: magnetic core memory and magnetic random-access memory (MRAM)*

What is the reason that we want to store information on magnetic media in the first place? There are primarily two reasons:

1. It is cheap and mostly easy to fabricate.
2. Permanent magnets are non-volatile. This means that we don't need energy to refresh data. The magnetic media maintains its magnetisation for many, many years when left untouched.

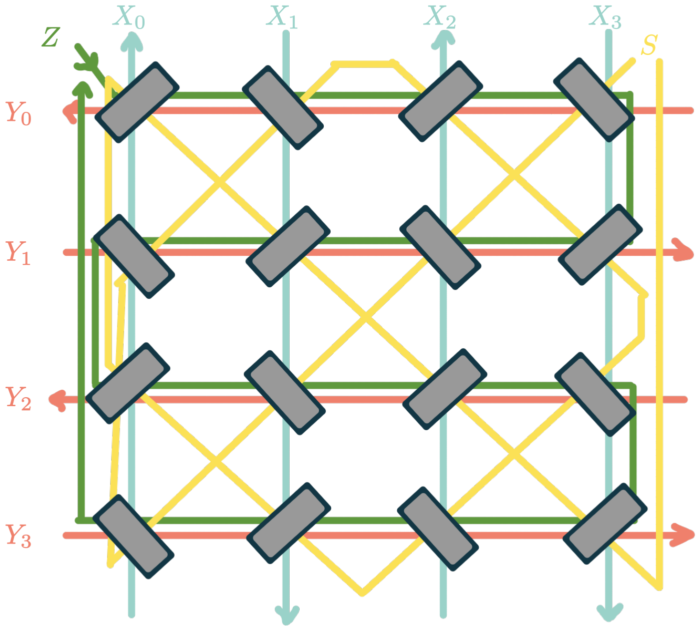
These reasons is also why the first real computer was magnetic (core memory was used in the computer of the Apollo missions!) and why long time storage of data is still done on magnetic media.

It was only in the 1960s, that magnetic computer memories were replaced by semiconductor memories. These store their data in the presence or lack of electric charge. Since charge easily leaks away, these technologies are volatile and require more energy. However, these semiconductor memories are based on transistors[1]. Due to very rapid development, transistors were shrinking half in size every two years. This made these technologies exponentially increase their storage density and exponentially decrease their fabrication cost. A trend that kept going for 60 years and magnetic computer memory could not compete with. Only in recent years a new magnetic computer memory was commercialized, named magnetic random-access memory (MRAM).

20 Core memory: how magnets brought us to the moon

Magnetic core RAM

Magnetic-core memory was developed between 1955 and 1975. Core memory uses small rings of hard ferromagnetic material. Each core is one bit. A clockwise magnetised ring is a “1” and a counter-clockwise a “0”. Two or more wires pass through each core, forming an X–Y array of cores. When an electrical current above a certain threshold is applied to the wires, the core will become magnetized. The core to be assigned a value –or written– is selected by powering one X and one Y wire to half of the required power, such that only the single core at the intersection is written.



---

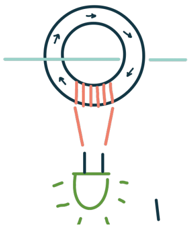
[1] Tiny electronic switches that we can control by locally applying voltages

When not being read or written, the cores maintain the last value they had, even if the power is turned off. Therefore, they are a type of non-volatile memory. Depending on how it was wired, core memory could be exceptionally reliable.

Core rope ROM

There is also a read-only variant of this memory. The bits in read-only memory (ROM) are pre-programmed in the hardware, so the user cannot store new data, only read the pre-programmed one. The read-only variant is called **core rope memory**. Unlike magnetic-core memory, the cores themselves are not used to store the data; the way a core is wired controls whether that core represents a '0' or a '1'.

The core rope memory works as follows: a wire is woven through many paramagnetic rings (cores) in a very specific pattern, where each ring represents one bit. If the wire passes through the ring, the bit is assigned a "1", and when it skips the ring it is assigned a "0". Around each ring, we also wind a coil, that is connected to a current detector. The memory can then be read out by passing a pulse through the woven wire.



If the wire passes through the ring (1), the ring will get magnetized for a very short time, as seen in the picture on the left. This change in the magnetic field will be picked up by the coil, and a current will be detected (here represented by an LED lighting up).
If the wire has skipped this ring (0), the ring will not be magnetized, and no current will be detected in the coil.

You can weave this wire through 16 rings (16 bits) consecutively, to get one byte per wire. If you then weave as many wires through the rings as possible, you can maximize the amount of information you are able to store in the rings.

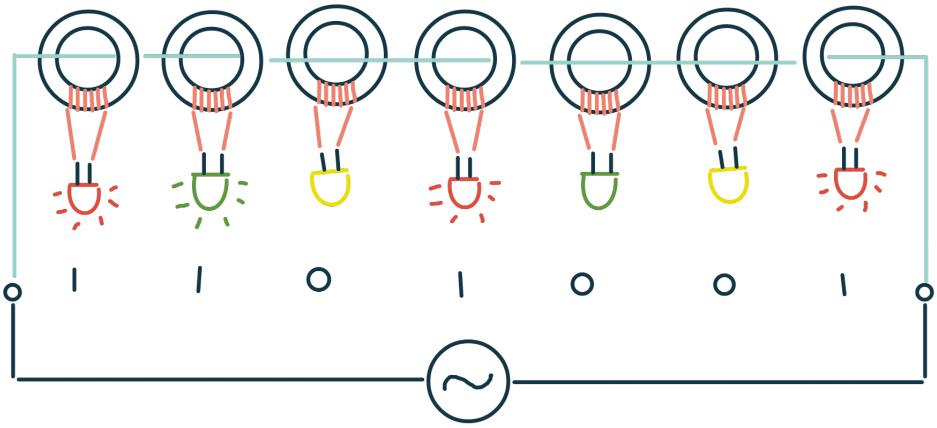


Figure 16: An example of a 7-bit wire, woven to create 1101001.

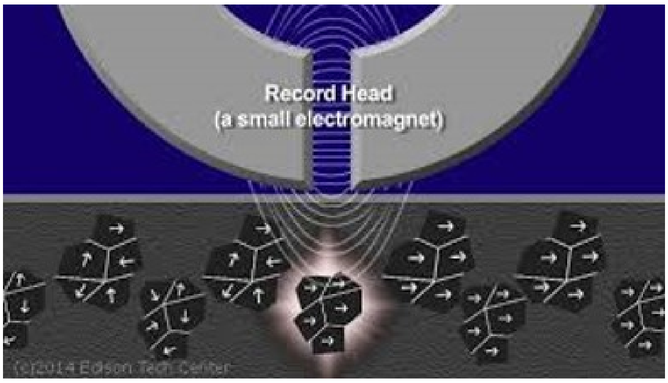
This ROM was used on the mission-critical Apollo Guidance Computer essential to NASA's successful Moon landings. Thousands and thousands of wires were woven through many many rings, to write the programs that were necessary to launch a spaceship into the sky. Together with the RAM memory[2], they only needed to 152kB of information into the computer of the Apollo spacecraft. Which was a good thing, because core rope memory is nowhere near as compact as the memory of, say, your smartphone. A core rope memory can store about 2.5MB per cubic meter!

If one byte consists of 16 bits (i.e. one wire is woven through 16 rings), and the total storage memory was about 72kB big, then 72.000 wires needed to be woven, with in total 1.152.000 stitches through the rings! That is quite a challenging task, which is why NASA recruited the help of women working in the rocket factories.

Figure 17: NASA women weaving the core rope ROM and the core RAM for the Apollo mission.

21 Tape recording

Magnetic tape stores its data in **elongated magnetic particles that are coated on along tape**. In elongated particles, the magnetisation wants to align along the long axis and this results in two magnetization directions opposed to each other. By sensing the direction of the magnetisation of the particles, we read out our bits.



---

[2] For the RAM they used the X-Y core memory.

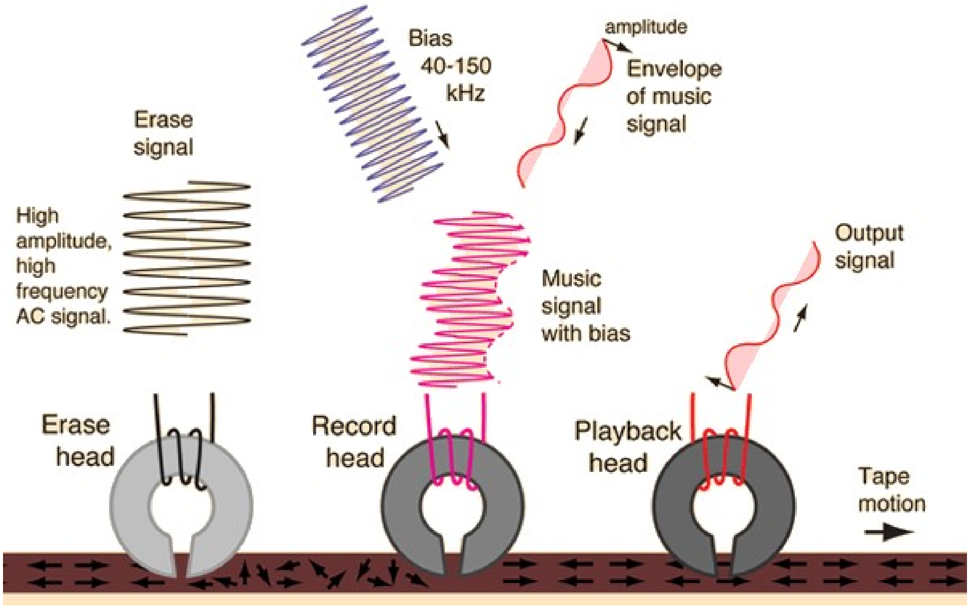
There are three uses for the magnetic tape as a storage medium: **audio, video and computer data**.

Magnetic tape revolutionized sound recording, reproduction and broadcasting. It allowed radio, which had always been broadcast live, to be recorded for later or repeated airing.

Even more revolutionizing were video tapes, such as VHS. These allowed video and sound storage on the same medium. This made it possible to enjoy your favourite movie at home, or to record video yourself. It was the main video storage until the invention of the DVD.

Since the early 1950s, magnetic tape has been used with computers to store large quantities of data and is still used for backup purposes. In fact, currently, this is still by far the cheapest way to store vast amounts of backup data.

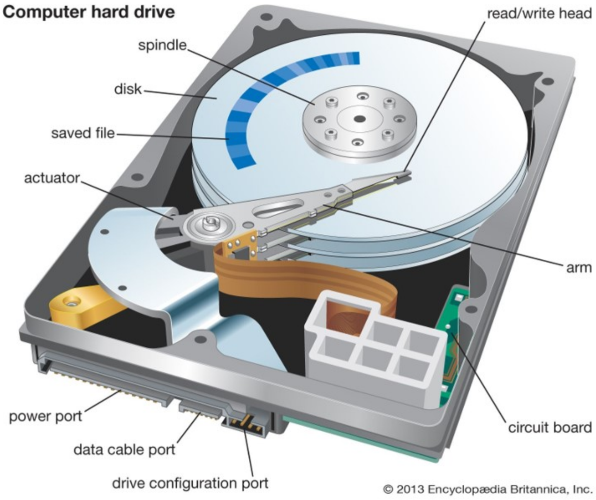
For all these purposes the recording and reading of magnetic tape happens according to the following working principle:



Over the years the heads evolved to smaller and more functional designs. But the biggest challenges were in the magnetic particles. We want to fit as many bits per area of tape that we possibility can. This means that the magnetic particle need to be smaller and sit together much closer. However, this results in an easier demagnetisation of the particle, which equals loss of our data. So scientists had to look for better and better magnetic materials with higher coercivity that didn't demagnetise as quickly.

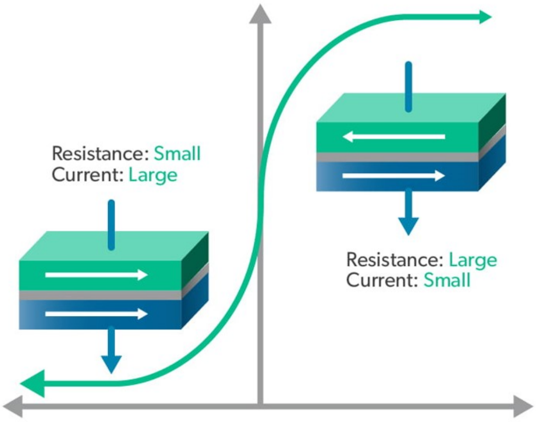
22 Hard disk drives (HDD)

In addition to conventional tape recording described above, in the early days of personal computers there was a need to have a device to hold software and store files due to the lack of capacity of random access memory at that time. To this end a floppy disk system was developed which was basically a circular sheet of tape contained within a cardboard or plastic sleeve with a window on it which was either internal or external to early personal computers. Using the disk format had obvious advantages over a tape format in that access times were reduced significantly and the device was capable of storing data at a higher density than conventional tapes available at that time. So-called floppy disks again went through a number of generations with increasing coercivity in the particles similar to that used in tape recording. However, the requirement for ever larger coercivity and the obvious advantages of having a mechanically rigid system, rapidly led to the development of hard disk drives (HDD). Also, on a rigid disk it is possible to have a much higher track density and rotation speed. So in summary, the working principle of the HDD is very similar to that of magnetic tape, but it is not flexible as tape and the magnetic material is on a disk, hence hard disk drive.



22.1 Magnetoresistance read-write heads

As the size of the magnetic particle decreased, the read and write heads needed to be also smaller, otherwise one would read out (detect) the magnetisation of multiple particles at the same time. Furhtermore, they needed to become more sensitive, since smaller particles have smaller magnetisation to detect.



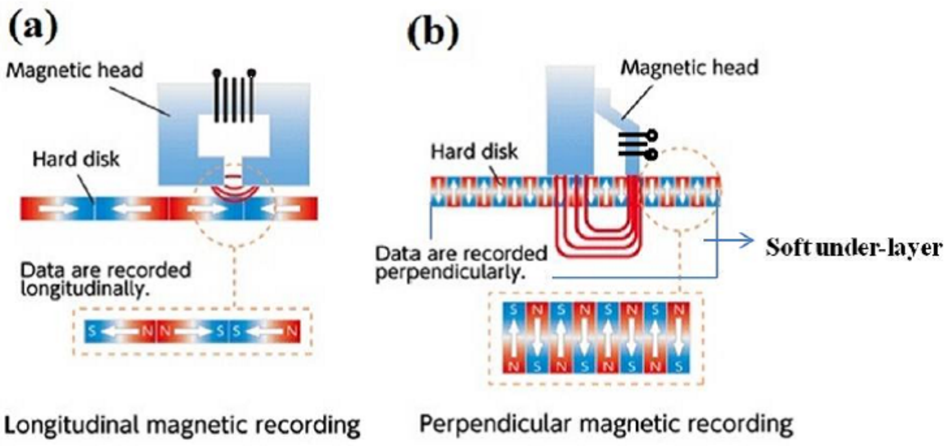
In the 1970s, the pick-up coil heads, that made use of magnetic induction, were replaced by read/write heads that detect or use **magnetoresistance**. Scientist found that the electrical resistance of magnets can be dependent on its magnetisation direction. The 2007 Nobel Prize in physics was awarded to Albert Fert and Peter Grünberg for the discovery of giant magnetoresistance. Eventually, it was found that even better detection could be reached by using tunnel magnetoresistance. Tunneling is a quantum mechanical effect of electrons passing a very thin insulating barrier (which normally it shouldn't pass). It turned out that electrons tunnel easily through the insulator between two ferromagnets that are magnetised in the same direction and find it very difficult to tunnel when they are anti-parallel aligned. This means that if you send a current through this so-called tunnel junction, it will have a high resistance when they are anti-parallel and low when they are parallel. Now, if we hover this tunnel junction above a magnetised particle from a HDD, the bottom layer will align itself with the particle. Then we detect if the bottom layer is now parallel or anti-parallel with the top, and thus we indirectly read the direction of our magnetised particle. Writing data can be done in the same fashion except now the head needs to be of a higher coercivity so that the magnetisation of the particle follows the one of the tunnel junction.

The magnetoresistance read/write heads made an exponential increase of data density in HDD possible over time possible. Between 1990 and 2020, a more than 100,000x density increase was achieved. This means that we can store vastly more data on the same area, which also make it a lot cheaper to produce.

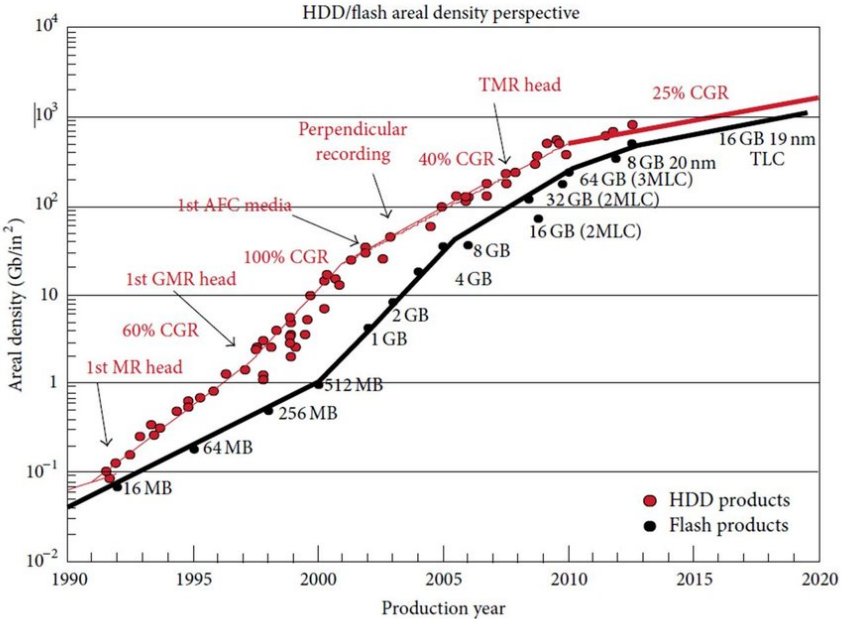
22.2 Transition from longitudinal to perpendicular recording

As demand for more and more data grew, HDD needed to be able to store its data more and more densely. After a while, the magnetic particles that we saw in magnetic tape were replaced by a continuous magnetic film where the data was stored in the individual domains of the ferromagnet. Also here, scientist and engineers looked for ways to make the domains as small as possible. However, the same problem remained, as the domains got smaller and less elongated, it became harder to prevent spontaneous demagnetisation. It looked like they were reaching an unavoidable wall in the scaling trend of the domains.

However by the 2000s, they solved that issue. What if you magnetise the domains perpendicular to the plane of the disk? That way the domains are still elongated (in the perpendicular directions) and can sit much closer together in the longitudinal direction. Additionally, this configuration also solved the main problem they had before, since the demagnetising fields of the neighbours is naturally stabilising the magnetisation of the domain.



Of course, it looks very simple and straightforward now to transition to perpendicular magnetised systems. However, at the time many scientist believed this was downright impossible. This is because of the **magnetic anisotropy**[3] of thin films. Like a metallic needle magnetises along it longitudinal direction and consequently directs itself to the magnetic north pole, so does a thin film magnetise along the film plane. The film thickness is only several nanometres as the radius can be micro- to decimetres. To “force” the magnetisation perpendicular to the plane of the film/disk, there should be a stronger anisotropy in that direction. This was achieved by the combination of very neat material research and development of extremely accurate material deposition tools. It was found that if the thickness of magnetic films got ultra thin (around ¡ 3-4 atomic layers), perpendicular magnetic anisotropy was achieved in some materials (with the right crystal structure).



23 Magnetic Random Access Memory (MRAM)

After the magnetic core memories were replaced by electric RAM, the latter exponentially developed together with development of faster, smaller and cheaper transistors. This meant that, until recently, there was no magnetic material based RAM that could compete with electronic RAM. However, all the new developments for HDDs, like the magnetoresistance and perpendicular magnetised domains, looked very promising to be used in a magnetic RAM (MRAM).

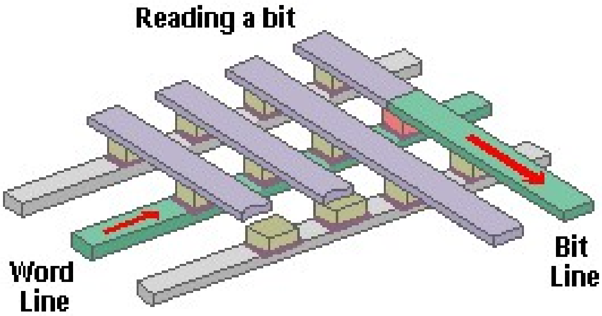
Over the years, different designs and effects were used to development the MRAM technology. But very importantly in recent years, MRAM is being integrated in technologies. Of course, MRAM still cannot compete with DRAM in performance, but it is very energy efficient. So it’s being used when power efficiency is important, such as smart watches like Fitbit. If your RAM would consume a lot of energy to perform, you would have to charge your smartwatch every hour or so.

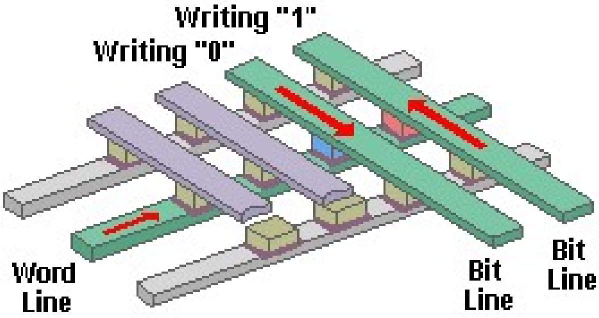
---

[3] An-isotropy means that it is not isotropic. In an isotropic system, there is no difference in directions. For example, the distance from centre to the edge of a circle are all the same, doesn’t matter in which direction you look/walk. However, in a ellipse, it is obvious that some distances to the edge are shorter and some are longer. If you want to preserve your energy as much as possible, you of course want to take the shortest distance. An ellipse is anistropic in shape.

MRAM makes use of the magnetic tunnel junctions that were developed for the read heads for HDD. However, now the bit itself is stored in the tunnel junction. Two parallel magnetised ferromagnets means a low resistance junction and is a "0", anti-parallel is high resistance and a "1".

Random-access also means that we need to access every tunnel junction individually. This is done by placing the junctions on the crossing of metallic bit and world lines:





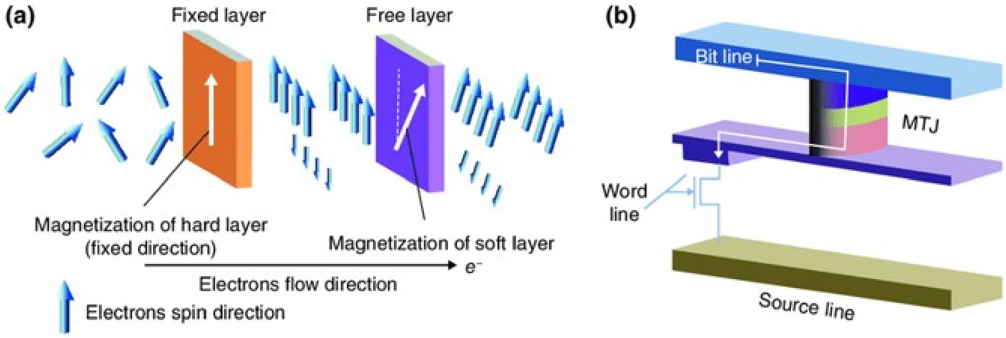
You can address the right junction by applying a small voltage on the world line and a small voltage on the bit line. That way only the junction on the crossing of both lines will conduct the right current.

Writing data in MRAM means that we deterministically want to change the magnetisation direction of one layer in the tunnel junction. Over the years, different writing mechanism were developed. The earliest and simplest design is depicted in the figure above: writing by induced magnetic fields. Since the world and bit lines are made out of conducting material, a current through them will induce a magnetic field. By applying the right current in both lines simultaneously, the added magnetic field is large enough to flip the direction of one of the ferromagnets. If the ferromagnet was already in that direction, nothing happened, but the results is always the same: magnetisation in a by us determined direction.

Writing by induced magnetic fields turned out to be too power consuming, since relatively large currents were needed to induce the right magnetic field. In recent years, writing by spin polarisation is used. The electrons of a charge current all have a spin, but since they all point in random directions, the electron current doesn't have net spin direction. However, there are physical phenomena that can filter or polarise[4] these spins. Injection of spin polarised electrons in a ferromagnet influences the direction of the spins of the electrons of the material itself. Since electron spins are the magnetisation of the material, it is possible to change the direction of the magnetisation in this way. This mechanism does not require high currents, this is why currently most MRAMs are produced with this writing mechanism.

---

[4] Make them have all the same direction.



At the moment, many other quantum mechanical effects are being explored to develop new kinds of MRAM, which will be important in a world were energy efficient applications become more and more useful.

Chapter VII

Introduction to Superconductivity

In the previous chapters, we have learned a lot about how magnets work. We saw that all materials have magnetic properties, and that these properties depend on their electron configuration. In fact, all of a material's properties (apart from its mass density) are determined by how the electrons behave inside it. For example:

- **Color**: which wavelengths an object absorbs and reflects depends on the energy levels of the electrons.
- **Strength**: determined by the bond strength (quantum interactions between electrons) and crystal structure (depends on the number of electrons available for bonding)
- **Conductivity**: whether or not a material conducts electricity depends on whether there are any electrons free to be moved around.

One of the most spectacular properties a material can have 'out-of-the-box' is **superconductivity**. The discovery of superconductivity happened a little over 100 years ago, and it is a fun story, so please allow me to tell it, briefly.

24 The discovery of superconductivity

The protagonist of this story is a Dutchman named **Heike Kamerlingh-Onnes**. Born in 1853, he was thrown into an interesting world in physics. At least, that we know *now.* At the time many scientists were growing more and more confident that they had discovered *all* the laws of physics, and that 'the rest are details'[1]. Little did they know that the small inconsistencies would build up and lead to a true revolution in physics with the discovery of quantum mechanics and the theory of relativity.

In 1870, Onnes started studying physics, his mentors included Robert Bunsen[2] and Gustav Kirchhoff[3]. He started pursuing a doctorate and during this time met Johannes Diedrich Van der Waals[4], who was trying to improve upon the gas laws known at that time (the ideal gas law). Crucially, gases deviate more from ideal gases when cooled down – a lot.

In this way, Onnes got in the business of making stuff really cold. He was not the only one trying, though, in fact, there was somewhat of a race to 0K going on at the time. Already in 1877, French and Swiss scientists Lewis Coilletet and Raoul Pictet succeeded in liquefying

---

[1] Here I have somewhat appropriated a quote by Albert Einstein, which he used in a different context, but conveying the same meaning: "*I want to know God's thoughts, the rest are details*".

[2] A well-known name to any high school student who has used a burner.

[3] Famous for Kirchhoff's circuit laws.

[4] Famous for the *Van der Waals gas model*, also called *real gas model.*

oxygen and nitrogen in small amounts[5]. In 1898, a Scottish scientist called James Dewar was the first to liquefy hydrogen.

The ‘race’ was maybe not as much a race as I’m making it out to be, since Onnes’ interest was not to get to low temperatures just for the sake of it. Rather, his goal was to observe the gases in this cooled state, and for this you need large amounts of them. For this reason, Onnes’ lab was more like a factory, producing many times more liquid hydrogen than Dewar, which would play to his advantage later.

With the discovery of helium in 1895[6], liquid helium became the Holy Grail of the cooling business. Here, Onnes succeeded first, largely thanks to his large supply of liquid hydrogen, which was crucial in the cooling process. In 1908, he had liquefied helium at a temperature of $4.2K$. A few years later, in 1911, Onnes’ lab had a helium cryostat[7] ready for use.

Another debate at that time was concerned with the resistivity of metals as a function of temperature. It had been observed that a current flows slightly better when metals are cooled (R descreases). On the other hand, it was believed that, as you near absolute zero, the electrons would come to a halt, blowing R up to infinity.

Onnes, with his helium cryostat, was in a good position to put this conjecture to the test. He chose to perform the experiment using mercury (Hg)[8] because it was easy to purify. As the temperature was lowered, the resistance also decreased steadily, until at liquid helium temperature **the resistance dropped to zero**. The precise opposite of what has been conjectured!

Of course, the lab team thought something must be wrong, but eventually they decided the drop in resistance must be real. Onnes called this phenomenon **superconductivity**. It would take another 44 years and a quantum revolution for superconductivity to be understood theoretically.

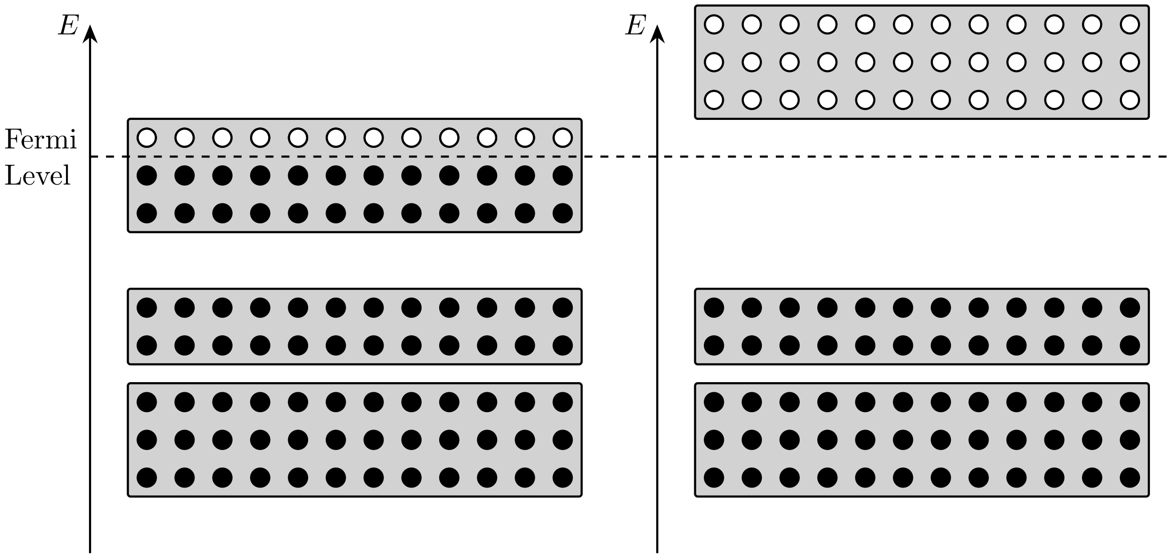


Figure 18: Two examples of band structures, one where the Fermi level lies inside the band, the other where it lies between two bands.

---

[5] This was a big achievement – at the time many people thought that these gases were ‘beyond liquefaction’, along with H_2 and He_2.

[6] On Earth. Helium had already been observed on the sun – hence the name, meaning *sun-stuff* – but not yet on Earth.

[7] A *cryostat* is a vacuum insulated sample environment that uses liquid cryogens for performing experiments at low temperatures.

[8] The symbol for mercury, Hg, comes from its Greek name *hydrargyrum* coming from the greek words for *water* and *silver*, i.e. *liquid silver*. In English, a synonym for mercury is *quicksilver* which has a similar vibe to it.

25 Insulators, conductors and semiconductors

Before we tackle superconductivity, we first need to understand normal conductivity, and the difference between conductors and insulators.

Remember the theory and rules for a single atom: electrons were assigned orbitals or shells in a certain order. Some of the shells would be entirely filled, and the last shell would be partially full or empty depending on the atom (on the number of electrons). The theory for solids is similar, but slightly different.

The electrons will also fill up certain levels, but now they are called **bands**, not shells. You start filling up the lowest band, and once it is completely full, you go to the next. At some point you are out of electrons, and you have filled up the bands up to a certain level called the **Fermi level**. If the Fermi level lies inside a band, then this band will be only partially filled. If it lies between two bands, then the band below will be completely filled, and the one above will be entirely empty.

The band structure of a material is very important for its conductivity properties. Let's apply a voltage to the material. This will alter the band picture as follows:

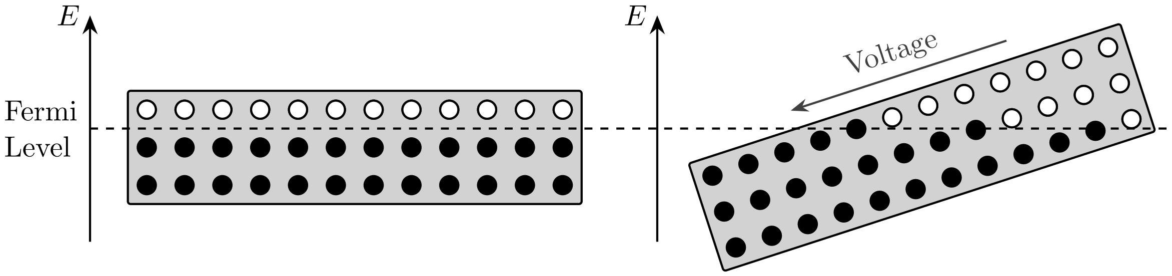


Figure 19: Example of what happens when a voltage is applied. When the last band is partially empty, the marbles/electrons can move down, so a current will flow through the material.

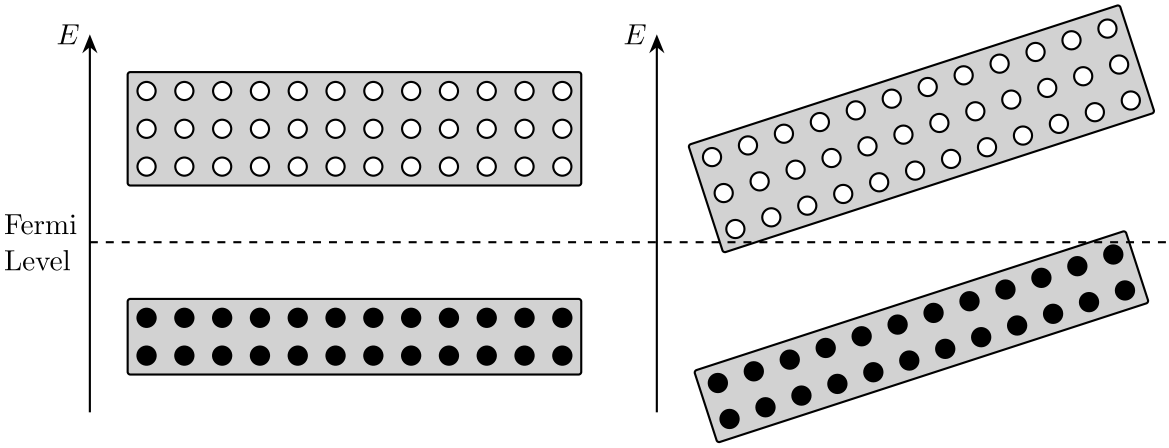
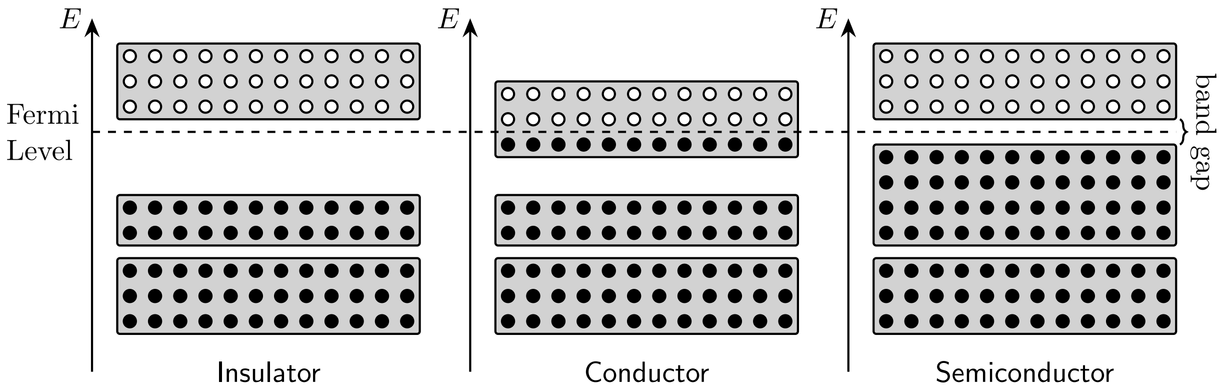


Figure 20: Example of what happens when a voltage is applied. When The last band is completely full, there is no room for the marbles to move, so no current can flow.

If you view the bands as boxes full of marbles, you can imagine that they would want to flow to the bottom left corner of their box (because gravity pulls them down). For a partially filled band/box of marbles, the marbles have the room to flow down when the box is turned by the voltage. However, if the box is already filled to the brim, the marbles cannot flow down because there is no space to move!

This marks the difference between a conductor and an insulator. For a **conductor**, the highest band is only half-filled with marbles/electrons. When a voltage is applied, the electrons are free to move, and will conduct electricity. For an **insulator**, the highest band is completely filled, the electrons cannot move, even when a voltage is applied. No electricity can be conducted by insulators.

A special case is the **semiconductor**, where the highest band is completely filled, but it lies very close to the next, empty band. Here, in principle, the electrons cannot move, and you have an insulator. However, if you give the electrons some extra energy, they might be able to jump from the filled band to the empty band[9], allowing for movement there and creating an empty spot below. This will allow a small amount of conductivity. So, a semiconductor is an insulator by default, but can turn into a conductor if you give it a little bit more energy. This empty space between bands is called a **band gap**.



We now know the difference between conductors and insulators, but what about **resistance**? The short answer is: electrons bump into stuff. This slows them down, and therefore limits the current that can flow through a conductor. More scientifically, the electrons 'scatter' off of 'scattering centers'. What sort of things do they bump into? Usually these are the atoms making up the crystal, or impurities in the lattice.

When an electron gets scattered, it will be put in a different state (maybe with different energy, momentum, ...). For this to be possible, there must of course be such a state available.

---

[9] Note that this is not possible for insulators, because there the filled and empty bands are very far apart. You would need a LOT of energy to make an electron jump over this gap.

A note on Fermions and Bosons You already know that two electrons cannot occupy the same quantum state (much like marbles cannot overlap). This fact is called the *Pauli exclusion principle*, which we already talked about when discussing Hund's rules. This principle would not be called a principle if it applied only to electrons. Indeed, the exclusion principle is more general than that, and applies to all particles that are *fermions.* The definition of a fermion is that it obeys the Pauli exclusion principle, and that it has half-integer spin. The most famous examples are electrons and protons, but there are many more. Fun fact: all matter is made up of Fermions!
There are also particles that do not follow the Pauli exclusion principle. These particles have integer spin and are called *bosons.* Since they don't obey the exclusion principle, they can be in the same state! This makes them have the tendency to bunch up in the lowest-energy state and form a *Bose-Einstein condensate*, a sort of weird (super)fluid. The most famous bosons are probably photons and the Brout-Englert-Higgs boson. Fun fact: all forces are *mediated* by bosons!

26 Superconductors

Now that we know how conductors conduct and resistors resist, we have all the tools to understand how superconductivity arises. So let's dive into it.

To tell the truth, superconductivity is in fact a **loophole**. Given the band structure and the Pauli exclusion principle, there really is no way to get a material whose electrical resistance is **literally zero**. Unless... you find a way *around* the rules.

The loophole of superconductivity goes as follows: imagine that electrons (with spin-1/2, obeying the exclusion principle) would not travel alone, but as couples or **pairs**. This would give them, as a unit, either spin-1 or spin-0. In either case, these pairs are bosons, who are not subject to the exclusion principle!

This means that these **Cooper pairs**[10] can (and happily do) occupy the same quantum state. The pairs of marbles, if you will, can all lie together at the bottom of the box[11]. When a voltage is applied, the marbles have room to move and current flows, regardless of the band structure. And here is the punchline: Cooper pairs *do not experience resistance*!

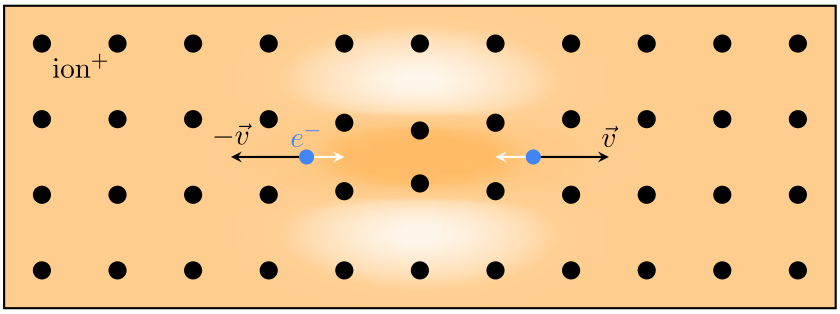
Let us be a little more precise. For a start, how can electrons form a pair, since the still repel each other? There are multiple mechanisms to form Cooper pairs, but we will not go into mathematical detail here. Rather, we will try to convey the right intuition.

Imagine a (negatively charged) electron moving through a crystal lattice. Since the lattice is made up of positively charged ions, those will be attracted to the electron, and move a tiny bit when the electron passes. When the ions move closer together, they form a region where there is a stronger positive charge density than elsewhere, attracting other electrons as they pass by. Though this positively charged region, a second electron is (indirectly) attracted to the first.

---

[10] After Leon Cooper, who discovered that this was possible, and together with John Bardeen and John Schrieffer developed the successful *BCS theory* of superconductivity.

[11] In the case of marbles, though, this is clearly impossible.



This example might seem a bit contrived, and indeed it is. But it serves as an example of how to think of these couplings of electrons. More generally (and more accurately) the attraction between electrons is **mediated** by vibrations in the crystal lattice[12].

Why do these pair not feel any resistance when they form a current? Cooper pairs cannot scatter, because there are no states available to scatter to! Cooper pairs form a Bose-Einstein condensate in the ground state. The nearest state a pair could scatter to is denoted as being some energy Δ above the ground state (a energy gap, as before for insulators and semiconductors).

If the temperature is low, the fluctuations on the system are very small. When $k_B T$ is below the energy gap Δ, Cooper pairs will not have enough energy to bridge the gap and break the pair. They will stay in their Bose-Einstein condensate. We conclude that, at low temperatures, Cooper pairs are **protected** from scattering by the energy gap Δ.

The condensate of Cooper pairs is an interesting system because its description is quite simple. It is well described by a single wavefunction, instead of $\sim 10^{23}$ electron wavefunctions, making superconducting states **macroscopic quantum systems**.

High-T_c Superconductors

What I have told you until now completely explains superconductivity *qualitatively*. The BCS theory, developed by Bardeen, Cooper and Schrieffer is also quantitatively very successful at describing *some* superconductors.

The standard BCS theory assumes that the spins in a Cooper pair pair up to form a *spin-0* boson. And this seems to give accurate predictions of the properties of *ordinary superconductors* (most pure metals when cooled down are ordinary superconductors).

However, there is a different class of superconductors which is of high practical interest: these are the so-called *High-T_c Superconductors*[14].

These materials are often quite complex mixtures, often ceramics. One of the simplest ones to make is $YBa_2Cu_3O_7$, dubbed ‘YBCO’. The Cooper pairs in such high-temperature superconductors often have exotic pairing symmetries, making calculation of their properties much harder.

Much like in the case of ferromagnetism, one of the most useful models in gaining some understanding is the Hubbard model, which is still a far cry from a perfect, and exact description.

---

[12] These vibrations behave in many ways as a quantum particle would, and are called *phonons*, coming from the Greek word for *sound*, which itself consists of long-wavelength phonons[13].

[13] Just like light, sound behaves as a wave, but its energy comes in packets, in the case of light these packets are called *photons*.

[14] Don’t be fooled by the name: none of the superconductors currently known have a T_c anywhere near room temperature. High-T_c refers to anything above the boiling point of nitrogen, which is 77K or -196°C.