

Stern-Gerlach Simulations

Scott N. Walck

October 18, 2016

1 The Stern-Gerlach Experiment

The Stern-Gerlach experiment was performed by Otto Stern and Walther Gerlach in 1922, a time when the theory of quantum mechanics was being developed. A schematic of the Stern-Gerlach experiment is shown in Figure 1. Silver (Ag) atoms are heated in an oven and made into a beam by passing through small holes. Stern and Gerlach chose silver because, with a single electron in its outer shell, it mimics the magnetic behavior of an electron. The electron and the silver atom each possess a *magnetic dipole moment*, that is, they behave like tiny bar magnets in the presence of a magnetic field. Such a magnetic dipole moment will feel a torque, an urge to rotate, in the presence of a magnetic field, and will moreover feel a force if the magnetic field is *inhomogeneous*, changing from one position in space to another.

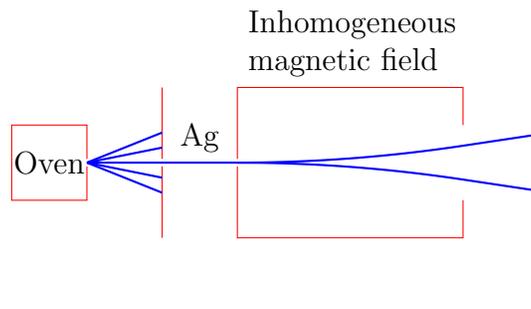


Figure 1: The Stern-Gerlach experiment.

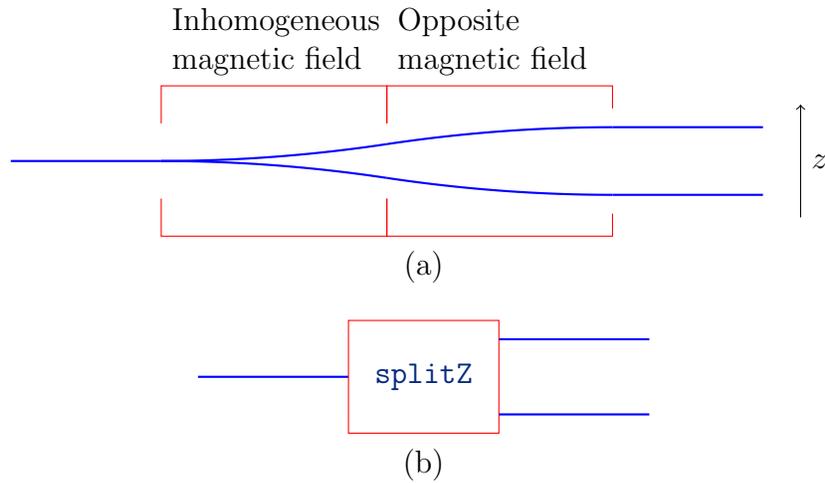
The Stern-Gerlach experiment aims to produce a force on the silver atoms with an inhomogeneous magnetic field oriented in a particular direction, say the z direction. The silver atoms in the beam are then expected to deflect upward or downward in the z direction, depending on the extent to which their magnetic dipole moments (vectors oriented along the imagined tiny bar magnets) point in the negative or positive z direction. Classical (pre-quantum) physics predicts that the random distribution of magnetic dipole moments coming from the oven should produce a continuous spectrum of deflection of the silver atom beam.

Instead, what is seen in the experiment is that the beam splits into two beams, and produces two spots on a detecting screen. Gerlach called this “directional quantization” [Tow12], and, since magnetic dipole moment is proportional to angular momentum, we now think of this as quantization of angular momentum. The electron and the silver atom are called spin-1/2 particles because the z -component of angular momentum for particles in one of the two beams is $(1/2)\hbar$, and that in other beam is $-(1/2)\hbar$. Spin-1/2 particles are particles that have two outcomes in a Stern-Gerlach experiment (as such, they are examples of quantum bits, or qubits). Spin-1 particles have three outcomes in a Stern-Gerlach type experiment, spin-3/2 particles have four outcomes, and so on.

2 Splitting Beams

We can learn quite a bit more from the Stern-Gerlach experiment if we can do sequential Stern-Gerlach measurements, that is if we can take one of the outgoing beams from the inhomogeneous magnetic field and send it into another Stern-Gerlach device. For this purpose it is helpful to have a Stern-Gerlach splitter that creates two parallel beams. Such a device is shown in Figure 2(a). The key to making the beams parallel is to put an oppositely directed inhomogeneous magnetic field immediately after the first field to deflect the beams back to parallel.

A schematic view of the SG (Stern-Gerlach) beam splitter is shown in Figure 2(b), which is labeled with the laboratory language function `splitZ`. The `splitZ` function takes a `BeamStack` as input, pops the top beam off of the stack, and replaces it with two new beams. The lower beam on the right side of the splitter (the *spin-down* beam) is placed on the top of the stack. Figure 2(c) lists functions for beam splitters in various directions. These



```

splitX :: BeamStack -> BeamStack
splitY :: BeamStack -> BeamStack
splitZ :: BeamStack -> BeamStack
split  :: Radians -> Radians -> BeamStack -> BeamStack
  
```

Figure 2: The Stern-Gerlach beam splitter. (a) A Stern-Gerlach splitter oriented in the z direction. (b) Schematic representation of the splitter in the z direction, using the `splitZ` function from part (c) of the figure. (c) Haskell functions for Stern-Gerlach beam splitters oriented in the x , y , and z directions. The `split` function takes two spherical coordinates in radians as arguments so that the splitting can be done in an arbitrary direction. (`Radians` is a type synonym for `Double`.) These functions act on the top (most recent) beam of the stack, remove that beam from the stack, and replace it with two new beams.

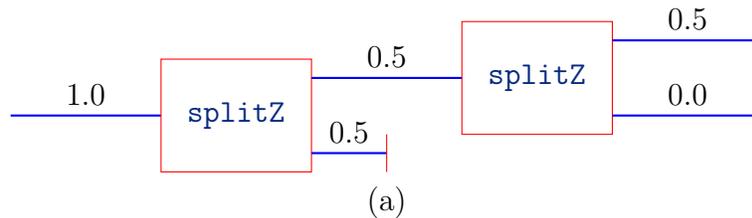
functions and the rest of the laboratory language are available in the module `Physics.Learn.BeamStack` in the *learn-physics* package[Wal16].

The quantum mechanics book by Townsend[Tow12] gives a sequence of SG experiments that help to show what a theory of quantum mechanics needs to explain, or at least predict. Townsend’s Experiment 1 is designed to show that although there is randomness in the measurement of spin-1/2 particles, there is not complete randomness in every measurement. In Experiment 1, shown in Figure 3, the z -spin-up beam of the first SG splitter goes into a second SG splitter oriented in the same direction. The results show that when a beam of z -spin-up particles enter a z -splitter, the entire beam comes out with z -spin-up. The intensity of 0.0 in the z -spin-down beam coming out of the second splitter represents a beam of no particles, or no beam at all. Part (b) of Figure 3 shows the use of the laboratory language in GHCi to carry out Experiment 1. The stack is shown so that the top beam on the stack is printed last, or at the bottom of the printed list. We use the `dropBeam` function because we have no further use for the z -spin-down beam exiting the first splitter. It is not necessary to drop the beam; we could have flipped the beams instead to act with the second splitter on the beam we want while continuing to include all beams in the stack.

In Experiment 1, a combination of splitting and dropping occurs that is called *filtering*. The first splitter is used to filter the beam for particles that have spin-up in the z direction. The filtering operation happens often enough that it is useful to name it. Figure 4 shows several filtering functions that we will use in upcoming experiments.

In Townsend’s Experiment 2, shown in Figure 5, the second z -splitter of Experiment 1 is replaced by an x -splitter. We see that an incoming beam of z -spin-up particles experiences a 50/50 split at the x -splitter. Although it is not shown in the figure, an incoming beam of z -spin-down particles will also experience a 50/50 split at an x -splitter.

Townsend’s Experiment 3 extends Experiment 2 with a third splitter, so that the orientations of the splitters are z then x then z . At the output of the last splitter, we now have equal intensities of z -spin-up and z -spin-down beams. This result may be surprising or puzzling when compared with Experiment 1, in which a second z -splitter sends all of the particles in the direction in which they split at a previous z -splitter. In Experiment 3, all of the particles entering the last z -splitter had previously split upward at the first z -splitter, yet now half of those entering the last z -splitter are splitting downward. Experiment 3 can also be viewed as inserting a filter for x -spin-up



```

GHCi, version 7.10.2: http://www.haskell.org/ghc/  :? for help
Prelude> :m Physics.Learn.BeamStack
Prelude Physics.Learn.BeamStack> randomBeam
Beam of intensity 1.0
Prelude Physics.Learn.BeamStack> splitZ it
Beam of intensity 0.5
Beam of intensity 0.5
Prelude Physics.Learn.BeamStack> dropBeam it
Beam of intensity 0.5
Prelude Physics.Learn.BeamStack> splitZ it
Beam of intensity 0.5
Beam of intensity 0.0

```

(b)

Figure 3: Townsend’s[Tow12] Experiment 1. (a) Measuring the same thing twice in succession gives the same results. Every particle that is found to deflect in the positive z direction at the first splitter also deflects in the positive z direction at the second splitter. We see this from the intensities. The entire intensity of 0.5 that enters the second splitter comes out with positive deflection. The intensity of 0.0 in the negatively deflected beam means that no particles are deflected in the negative z direction at the second splitter. (b) A GHCi transcript showing use of the laboratory language to obtain the results of Experiment 1. The beam at the top of the stack is the last beam printed and hence appears at the bottom of the list.

```

xpFilter :: BeamStack -> BeamStack
xpFilter = dropBeam . splitX

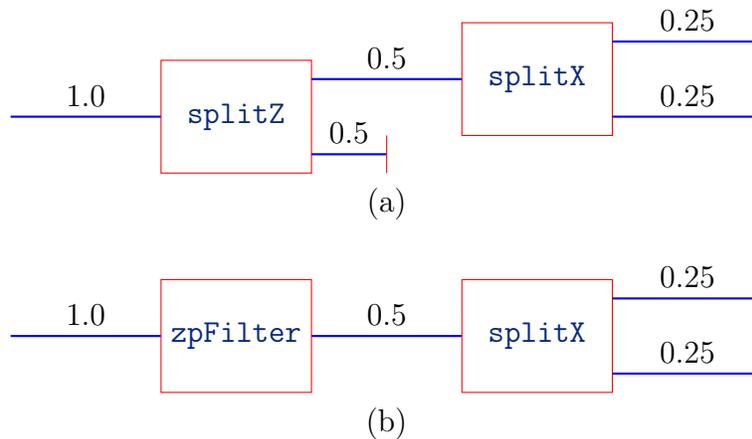
xmFilter :: BeamStack -> BeamStack
xmFilter = dropBeam . flipBeams . splitX

zpFilter :: BeamStack -> BeamStack
zpFilter = dropBeam . splitZ

zmFilter :: BeamStack -> BeamStack
zmFilter = dropBeam . flipBeams . splitZ

```

Figure 4: Filtering is the composition of splitting and dropping. The function `xpFilter` keeps only the beam that deflected in the positive x direction. Since the beam that deflected in the negative x direction is placed on the top of the stack in the `split` function, we merely have to drop it. The function `xmFilter` keeps only the beam that deflected in the negative x direction. Since the beam that deflected in the negative x direction is placed on the top of the stack in the `split` function, we need to flip the beams on the stack so that we drop the positive x beam. The functions `ypFilter` and `ymFilter` could, of course, also be defined.



```

GHCi, version 7.10.2: http://www.haskell.org/ghc/  :? for help
Prelude> :m Physics.Learn.BeamStack
Prelude Physics.Learn.BeamStack> zpFilter randomBeam
Beam of intensity 0.5
Prelude Physics.Learn.BeamStack> splitX it
Beam of intensity 0.25000000000000006
Beam of intensity 0.24999999999999994

```

(c)

Figure 5: Townsend's Experiment 2. (a) Schematic diagram with splitters. (b) Alternate diagram of the same experiment using a filter. (c) GHCi transcript.

particles between the splitters of Experiment 1. Clearly this filter is having a significant and strange effect on the final splitting. It seems as though the particles have “forgotten” that they had previously split upwards at a z -splitter. This is a crucial observation that will need to be reflected in the theory.

Experiment 3 shows that the upper beam exiting the x -splitter will undergo a 50/50 split at the subsequent z -splitter. It is also the case, as can be tested with the laboratory language, that the lower beam exiting the x -splitter will also undergo a 50/50 split if sent into a z -splitter.

We can recombine two beams with the same kind of inhomogeneous magnetic fields that we used to split a beam. Figure 7 shows an SG recombiner. Recombining is not as intuitive as it might seem. If the two beams that enter a recombiner did not come from a splitter in the same direction, there is no guarantee that they will bend the right way to merge them into a single beam. For example, flipping two beams before recombining will generally give different results (and often a beam intensity of zero) from simply recombining the two beams.¹

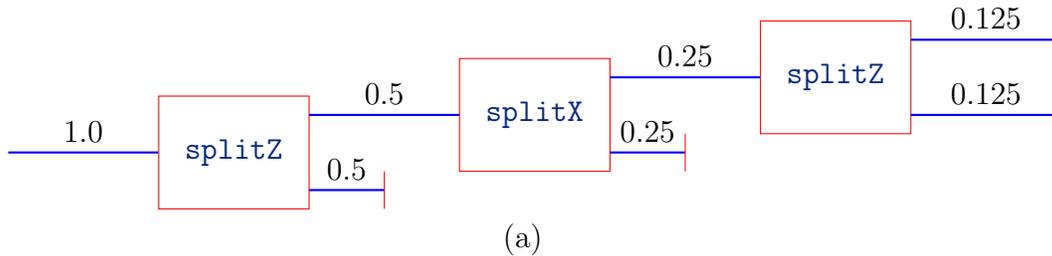
Townsend’s Experiment 4, shown in Figure 8, builds on Experiment 3 by adding an x -recombiner after the x -splitter. From Experiment 3, we know that each of the beams with intensity 0.25 in Figure 8 would experience a 50/50 split at a z -splitter. Recombining the two beams before the z -splitter produces a 100/0 split! The final z -splitter produces no z -spin-down particles, just as the final z -splitter in Experiment 1 produced no z -spin-down particles. Whereas the x -splitter in Experiment 3 disturbed the repeatability of the z -splitter results in Experiment 1, the x -splitter-recombiner combination in Experiment 4 does not disturb the repeatability. This is another result that the theory will need to deal with.

Of course we can use the laboratory language to go beyond Townsend’s experiments. Students can invent other configurations of splitters and recombiners and see if the results match their expectations.

The last basic building block of the laboratory language is application of a uniform magnetic field to a beam. A magnetic field can be applied in a particular direction with a certain strength for a certain amount of time. The functions to do this are as follows.

```
applyBFieldX :: Radians -> BeamStack -> BeamStack
applyBFieldY :: Radians -> BeamStack -> BeamStack
```

¹I thank my student Justin Cammarota for noticing this and bringing it to my attention.



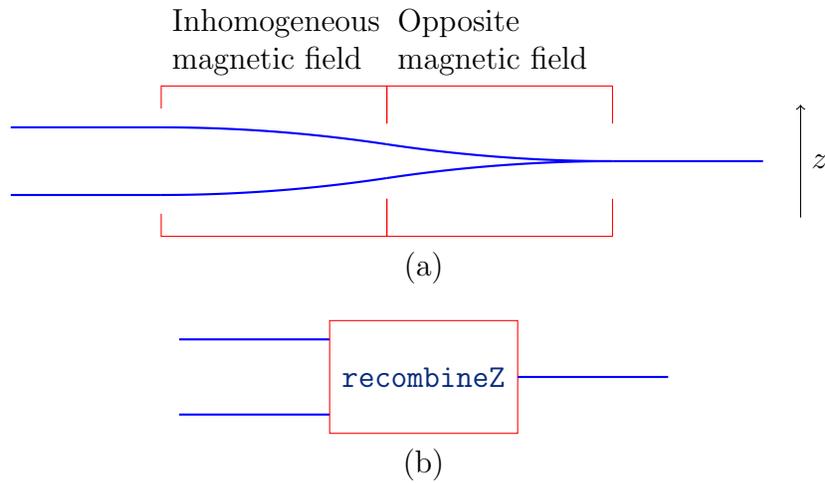
```

Prelude Physics.Learn.BeamStack> randomBeam
Beam of intensity 1.0
Prelude Physics.Learn.BeamStack> splitZ it
Beam of intensity 0.5
Beam of intensity 0.5
Prelude Physics.Learn.BeamStack> dropBeam it
Beam of intensity 0.5
Prelude Physics.Learn.BeamStack> splitX it
Beam of intensity 0.25000000000000006
Beam of intensity 0.24999999999999994
Prelude Physics.Learn.BeamStack> dropBeam it
Beam of intensity 0.25000000000000006
Prelude Physics.Learn.BeamStack> splitZ it
Beam of intensity 0.12500000000000006
Beam of intensity 0.125

```

(b)

Figure 6: Townsend’s Experiment 3. Some particles split downward at the last z -splitter, even though all particles entering the last z -splitter have previously split upward at the first z -splitter. This tempers the results of Experiment 1, which showed that no particles would split downward at the second z -splitter after they had split upward at the first z -splitter. Reconciling Experiments 1 and 3 is an important job for the theory. (a) Schematic diagram with splitters. (b) GHCi transcript. There are many alternate ways to do this, including using filters.



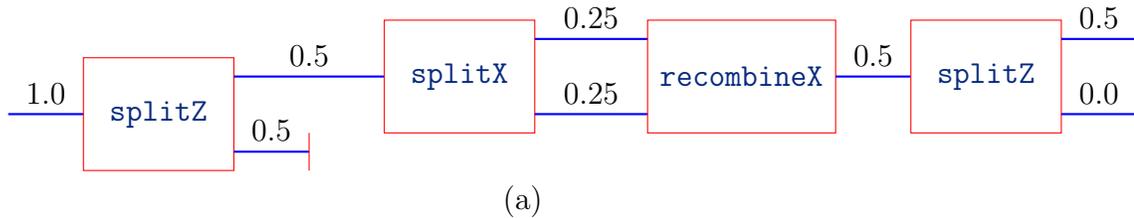
```

recombineX :: BeamStack -> BeamStack
recombineY :: BeamStack -> BeamStack
recombineZ :: BeamStack -> BeamStack
recombine  :: Radians -> Radians -> BeamStack -> BeamStack

```

(c)

Figure 7: The Stern-Gerlach beam recombiner. (a) A Stern-Gerlach recombining oriented in the z direction. (b) Schematic representation of the recombining in the z direction, using the `recombineZ` function from part (c) of the figure. (c) Stern-Gerlach beam recombiners oriented in the x , y , and z directions. The `recombine` function takes two spherical coordinates in radians as arguments so that the recombining can be done in an arbitrary direction. These functions act on the top two beams of the stack, remove those beams from the stack, and replace them with a single new beam.



```

Prelude Physics.Learn.BeamStack> randomBeam
Beam of intensity 1.0
Prelude Physics.Learn.BeamStack> splitZ it
Beam of intensity 0.5
Beam of intensity 0.5
Prelude Physics.Learn.BeamStack> dropBeam it
Beam of intensity 0.5
Prelude Physics.Learn.BeamStack> splitX it
Beam of intensity 0.25000000000000006
Beam of intensity 0.24999999999999994
Prelude Physics.Learn.BeamStack> recombineX it
Beam of intensity 0.5
Prelude Physics.Learn.BeamStack> splitZ it
Beam of intensity 0.5
Beam of intensity 0.0

```

(b)

Figure 8: Townsend’s Experiment 4. One way to view this experiment is that an x -splitter-recombiner pair has been inserted into the apparatus of Experiment 1, and the same results are obtained as in Experiment 1. A second way to view this experiment is that two beams, each of which would produce a 50/50 split in a `splitZ`, are being recombined into a beam that produces a 100/0 split in a `splitZ`.

```
applyBFieldZ :: Radians -> BeamStack -> BeamStack
applyBField  :: Radians -> Radians -> Radians -> BeamStack -> BeamStack
```

These functions apply a uniform magnetic field to the top beam of the stack. In the function `applyBFieldX`, the `Radians` argument is an angle in radians that represents a combination of the strength of the applied magnetic field and the duration over which it is applied. The `applyBField` function takes two spherical coordinates as arguments to represent the direction of the applied magnetic field, and a third numerical argument to represent the combination of magnetic field strength and time over which the field is applied.

3 Puzzles

1. First, find a sequence of two filters such that no particles exit the second filter (no particles is the same as a beam of zero intensity). Now, is it possible to find a third filter to place between the first two, such that particles now flow from the last filter? If this is possible, we may need to adjust our intuition about what a filter is.
2. Can you find a direction and duration for a uniform magnetic field to act on a beam exiting a `zpFilter` so that the entire beam intensity will make it through an `xpFilter`? Does this suggest a way to think about what a uniform magnetic field does to a spin-1/2 particle?
3. In Townsend's Experiment 4, suppose we apply a uniform magnetic field in the x direction to the lower beam between the x -splitter and x -recombiner. If the duration of application of the magnetic field is zero, the results will match that of Experiment 4. What is the next shortest duration when the results match again? Is the answer surprising?
4. Play around with the `BeamStack` language, putting splitters, recombiners, and magnetic fields in various configurations until you come up with something interesting or surprising. Write down something you found that was interesting or surprising. (Ok, this isn't really a puzzle; it's an open-ended creative endeavor.)

References

- [FLS65] Richard P. Feynman, Robert B. Leighton, and Matthew Sands. *The Feynman Lectures on Physics, Quantum Mechanics*. Addison-Wesley, 1965.
- [Kar03] Jerzy Karczmarczuk. Structure and interpretation of quantum mechanics: A functional framework. In *Proceedings of the 2003 ACM SIGPLAN Workshop on Haskell*, Haskell '03, pages 50–61, New York, NY, USA, 2003. ACM.
- [PFTV89] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 1989.
- [SN11] J. J. Sakurai and Jim Napolitano. *Modern Quantum Mechanics*. Addison-Wesley, 2nd edition, 2011.
- [SW10] Benjamin Schumacher and Michael Westmoreland. *Quantum Processes, Systems, and Information*. Cambridge University Press, 2010.
- [Tow12] John S. Townsend. *A Modern Approach to Quantum Mechanics*. University Science Books, 2nd edition, 2012.
- [Wal16] Scott N. Walck. The learn-physics package. <http://hackage.haskell.org/package/learn-physics>, 2012–2016.