Lecture Notes on Symmetry Optics

Lecture 10:
Information and Uncertainty

To accompany https://youtu.be/cr-I0iK18C0
Version 01, released September 28, 2022

Paul Mirsky
paulmirsky633@gmail.com

1 Introduction

1.1 Opening

Hello, and welcome to Lectures on Symmetry Optics. I'm Paul Mirsky. This is lecture 10, and the topic is: information and uncertainty.

It seems ironic that there have already been nine lectures on symmetry optics, and we have not yet addressed the topic of symmetry. In this lecture we finally address it. Symmetry is described in mathematics by the subject of group theory. That's a very unfortunate name, because the word 'group' really doesn't give you any feel for what a mathematical group is. Perhaps a better name might have been 'symmetry theory'.

The concepts developed in this lecture have been profoundly influenced by a remarkable book called *Asymmetry: the Foundation of Information*, by Scott J. Muller. The core idea of that book is among the deepest, most powerful insights I've ever encountered. And yet, Muller's work is virtually unknown. Note that even though Muller uses the term 'asymmetry' to mean something distinct from ordinary symmetry, we consider them both to be two aspects of the single concept of symmetry, which is more-or-less a synonym for 'group'.

1.2 Definition of a group

In any case, group theory deals with all the ramifications of a very basic concept. To call it basic, though, is not to say that it is easy to grasp. Compared to most of optics, group theory is almost unbearably abstract. To show you what I mean, here is the definition of a group:

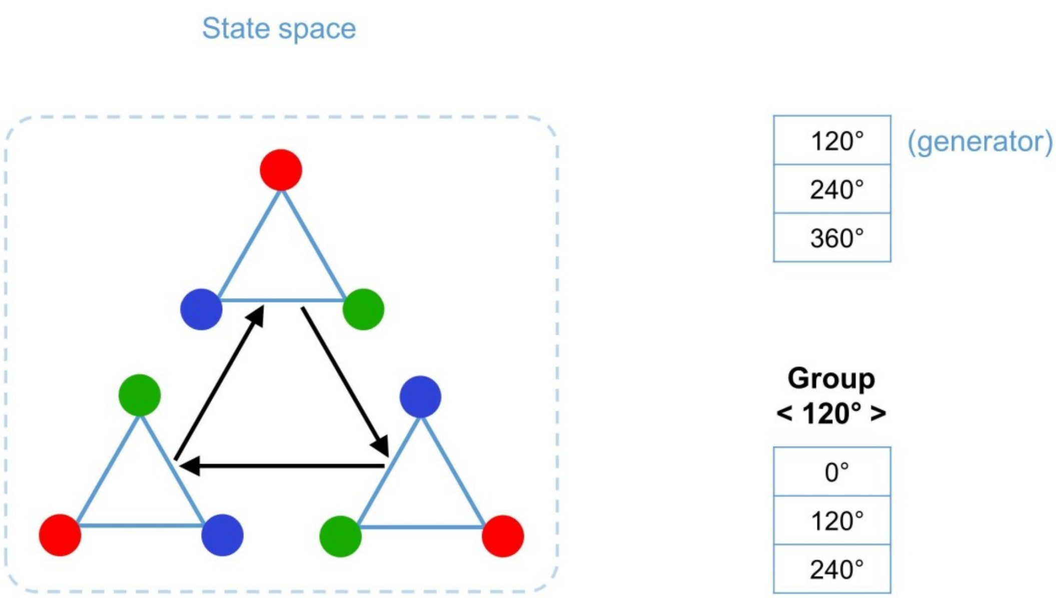
A group is set of elements with a binary operation and:

1. Closure
2. Identity element
3. Inverse
4. Associativity

Most people find this definition pretty incomprehensible, but it's pretty typical for definitions in higher mathematics. That discipline uses very strict, formal reasoning and it comes to conclusions that are truly airtight. But, we will do something much looser, which is to apply group theory as a mathematical tool to model physical objects. This will give us an interpretation of the math, which hopefully will be more intuitive than the formal axioms given here. We will address these formal axioms only if they arise in the discussion.

2 Information and uncertainty subgroups

2.1 Colored triangle



We'll start discussing symmetry in terms of a simple example. Consider an equilateral triangle with its 3 vertices colored red, green, and blue. It can be in any one of these 3 possible orientations, or in other words 3 possible states. For now, we're assuming that the triangle cannot rotate to any other angle, and so these 3 states are the entire state space of the triangle. We'll call the states by the orientation of the red vertex. Red up, red right, red left.

Now we'll change our perspective a little. Rather than thinking in terms of the state space, we'll think in terms of transformations from one state to another. For example, suppose that we start with red up. If we rotate the triangle around its own center clockwise by 120°, or 1/3 of a full rotation, we get red-right. This 120° rotation is a basic degree of freedom for this object, and we will imagine that we can apply that rotation by the click of a button. In group-theoretical terms, this is the *generator* of the group.

We click a second time, and the net effect is to transform the state by 240° degrees relative to the starting orientation. We will keep a list of all possible net transformations, and we will continue clicking as many times as we need until we see what is called *closure*, which is one of the formal axioms of group theory. Closure occurs when eventually, the transformation gets back to where it started. In the case of rotation, it occurs at 360°. After closure, no matter how many times you click, the list of transformations can never get any longer. You simply keep treading over old ground and you never get anywhere new.

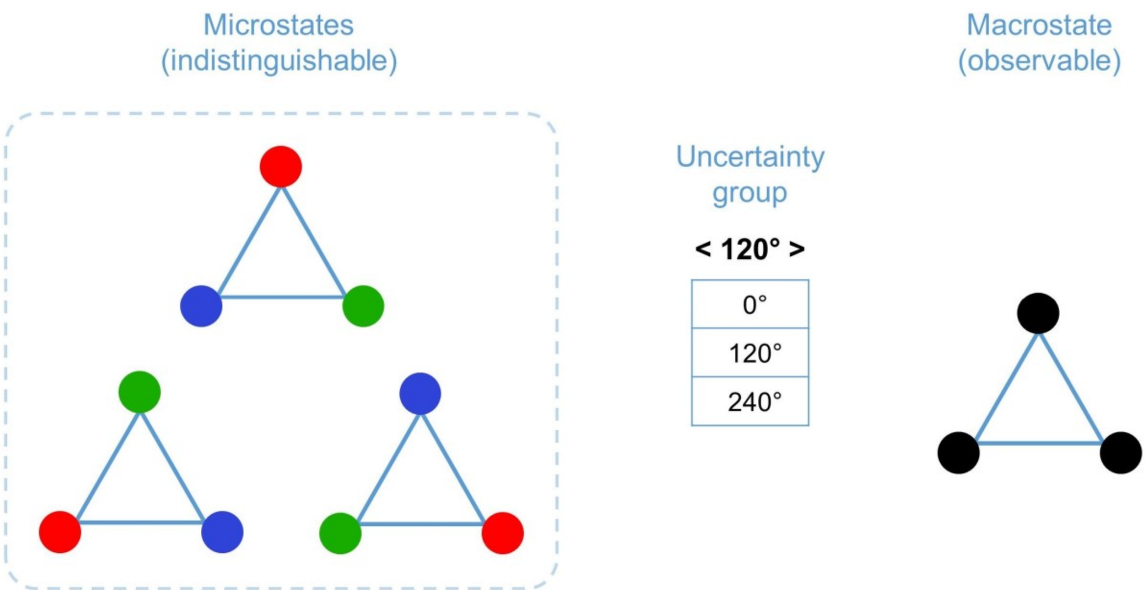
The three transformations on this list together constitute a group. Each transformation is an *element* of the group. The 120° rotation is the transformation that generates the group. One convention for notating groups is to call the group by its generators, written inside chevrons; so, this is the group <120°>.

Note that rotating by 360° is the same as rotating by 0°, and so we can re-write the list with 0° at the top. A 0° rotation is the *identity transformation*, which is the 'transformation' of not doing anything at all. Fundamentally, every group must include the identity.

Also: we took red-up as the starting state, so that red-up corresponds to the identity. But it's not essential to start with red-up. If we take red-right as the starting state instead, it means that the states now correspond differently to the group elements. But the group is the same in either case.

We will be very interested in what can and cannot be observed. In this example, we are able to distinguish the 3 states from one another by the different colors, and so an observation tells us precisely the state of the triangle.

2.2 Colored triangle, observed in black and white

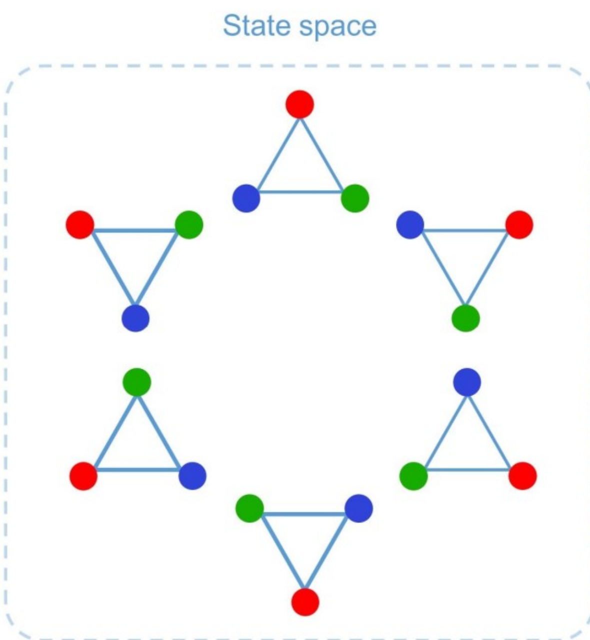


But next, let's consider how the state would appear to an observer who sees only in black and white. In this case, the observer can't distinguish the different orientations, and perceives only a single *macrostate*, which actually contains the 3 different *microstates*.

We don't think of an object as being in just one state. Rather, we think of a probability distribution over some range of possible states. At a minimum, all physical objects have some uncertainty due to thermal motion, unless they are at absolute zero.

The transformations in the group change the triangle from one microstate to another. The observer knows that it's in 1 of these 3 possible states, but doesn't know which one. The group describes that uncertainty. An equivalent way of thinking about the group it that the generators could act arbitrarily to randomize the state space, without having any effect on what the observer perceives.

2.3 Group of 6 rotations



Group

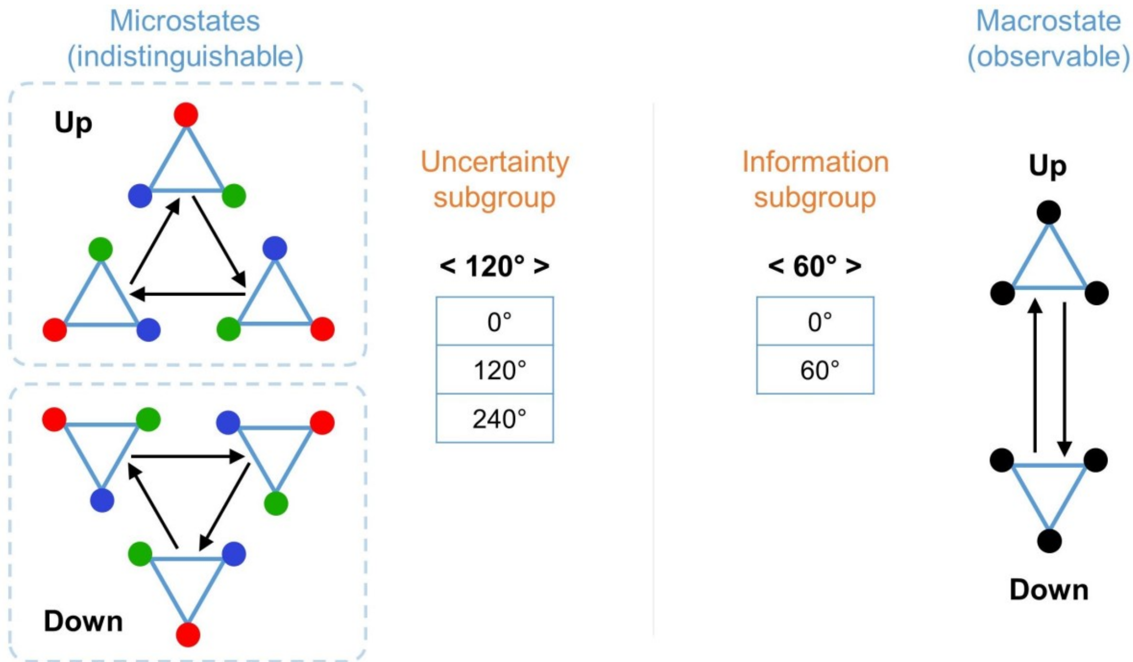
< 60° >

0°
60°
120°
180°
240°
300°

Next, consider a rotation by 60° – 1/6 of a full rotation. This expands the state space to 6 different orientations, and it expands the group to 6 different transformations.

What happens when this new state space is observed in black and white?

2.4 Group of 6 rotations, observed in black and white



In this case, the observer can *not* distinguish colors, but *can* distinguish vertex-up from vertex-down. In other words, the state space is partitioned into two subsets, and each subset constitutes a different macrostate, or a different observable state. An observer is able to perceive which macrostate the object is in currently.

The uncertainty from the previous case, <120°>, is now the *uncertainty subgroup* because it changes any vertex-up state into another vertex-up state, and it also changes any vertex-down state into another vertex-down state. However, it *preserves* the vertex direction. In other words, even if some degrees of freedom are allowed to act randomly, the randomness is bounded by the closure of the group transformations. Even though the state of an object will move around within a subspace, it will not leave the subspace.

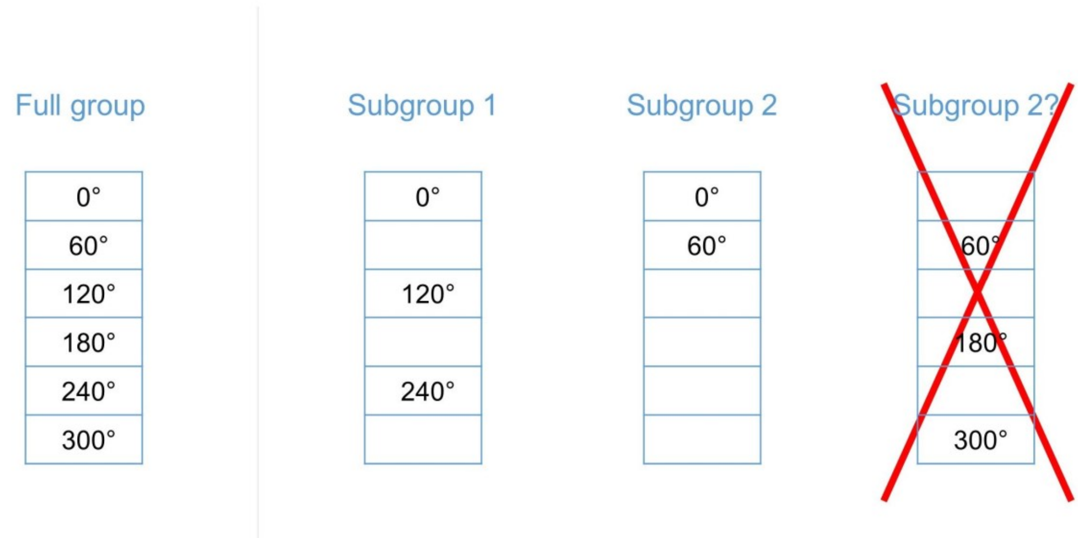
What about a 60° rotation? It would turn any vertex-up state into a vertex-down state, and vice-versa. In other words it also generates a group, namely <60°>, this one with two elements. This is called the *information subgroup*. It describes transformations that are *not* occurring.

The object is remaining at stasis in one macrostate, and thus it is *not* undergoing transformations into a different macrostate. Stasis by itself doesn't necessarily constitute information. But stasis is a precondition for having any 'information' about an object, because any change of macrostate would mean that the information no longer matches the object.

Uncertainty and information are two subgroups of the full group. In fact, the same boundary determines them both. All the generators of the group – in this case, rotations by 60° and 120° –

can be partitioned into two non-overlapping subsets. One set generates the uncertainty subgroup, the other one generates the information subgroup. This implies that if a generator of uncertainty is eliminated – for instance, if a color camera is used – that the same generator then becomes a generator of information instead.

2.5 Subgroups



Let's define the term *subgroup*, which we have used to describe information and uncertainty. A subgroup is a smaller group that is a subset of the full group. These 6 elements describe the full group. What subgroups does it have?

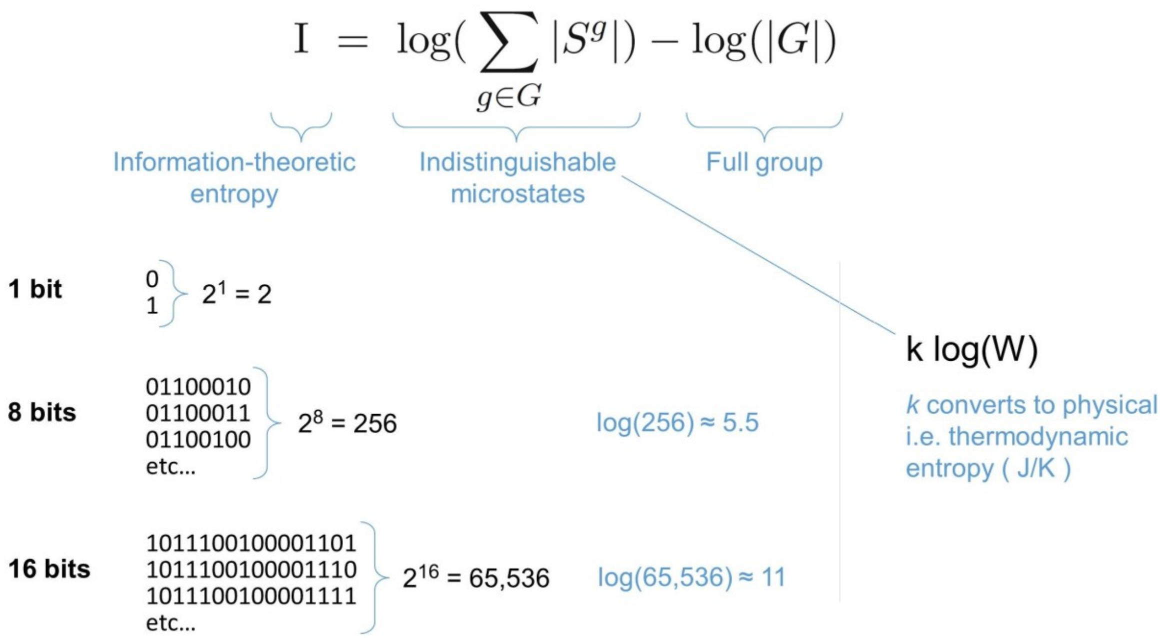
These 3 elements form <120°>, so that is a subgroup of the full group.

There is one more important subgroup. Try to guess what it is. Did you guess *this*? Wrong. This is *not* a subgroup. For one thing, it doesn't contain the identity element, therefore it's not a group, therefore it's not a subgroup. Rather, this is a *coset* of the subgroup <120°>. This lecture won't discuss cosets in any more detail; we are introducing this term only to show what a subgroup is *not* and help you to avoid a very natural mistake.

Here is the correct second subgroup. Its generator is a rotation by 60°. But when it acts twice, it returns the triangle to the starting state. Therefore it contains only the identity, and 60° rotation.

In going from 120° to 60°, we didn't just enlarge one degree of freedom. Rather, we added a second degree of freedom – another generator. By the way, there also technically exist a few other subgroups which are not relevant to our discussion. 0° and 180° constitute a subgroup. Also, the full group is a subgroup of itself. Finally, the identity by itself is a subgroup.

2.6 Muller's formula for entropy



In Muller's thesis, the unifying idea is the concept of *entropy*. And, we do need to discuss it. But in symmetry optics we avoid this word, because it is so confusing.

The word *entropy* means:

- Nothing to most people
- One thing to educated laypeople
- Two other things to scientists (and those two are exact opposites)
- A 1990's punk-rock band

This is Muller's formula for entropy. We won't study it in detail, but we will make a few important points:

First of all, it relates three terms. The one on the left is the information-theoretic entropy, which corresponds to the information subgroup. The second term corresponds to the number of indistinguishable microstates, and the last term is the entropy of the full group.

We'll start with the first term. Within mathematics and engineering, the field of *information theory* deals with fundamental questions about data and communication. The key concept is entropy, which in this case essentially comes down to counting the number of distinguishable states that an object can be in. For example, a digital bit can be in one of two states: on or off. 8 digital bits considered together as a single object can be in 2^8 or 256 different states.

If you double the number of bits to 16, you get 65,536 states, which is a big increase over 256. A more natural thing to count is not the number of *states* directly, but rather the number of *bits*. In this view, the second set of 8 bits carries the same amount of information as the first set of 8. Apart from a scale factor, counting bits is equivalent to taking the logarithm of the number of states. 8 bits has about 5.5 units of information entropy, and 16 bits has around 11 units. Generators of a group are similar to bits, in the sense that as more generators are added linearly, the size of the group increases exponentially.

The state of the object carries a particular message. For example, suppose that a digital picture of a flower is represented by a sequence of a million 0s and 1s, and a picture of a bird is represented by a different sequence of a million 0s and 1s. We can think of a million bits of computer memory collectively as a single object, and the two different pictures are two different states of that one object. The more states an object it has, the more different messages it can carry.

The second term corresponds to the uncertainty, which is also a type of entropy. Again, it's the logarithm of the number of states. But here it has the opposite meaning -- it's the number of indistinguishable microstates corresponding to each macrostate. But actually the main application for this concept of entropy isn't information theory at all, but rather thermodynamics. Considerations of entropy govern many processes in physics and chemistry. To convert entropy from information units into physical units of Joules / Kelvin, just multiply by Boltzmann's constant k.

This is the end of the first part of the lecture. Building on Muller's work, we have shown how the symmetry of an object can be divided into two subgroups – information and uncertainty.

3 Slide, Tilt, and Phase

3.1 Overview

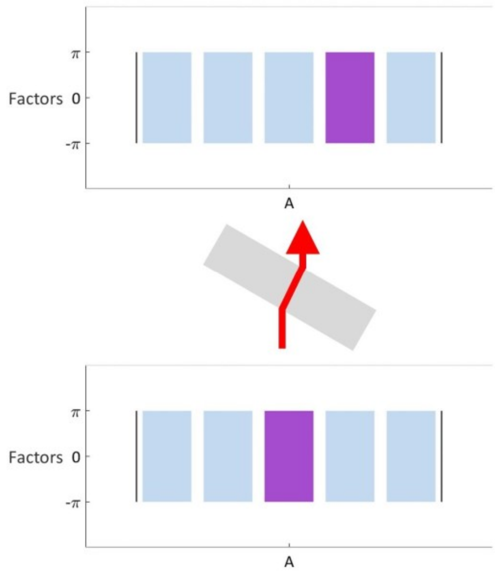
Now we'll begin the next part of the lecture, and we'll apply the new concepts to optical factors.

We will consider 3 basic transformations:

- Slide
- Tilt
- Phase

Each one can be physically implemented by a simple optical device. Each one is also represented by a matrix which acts on state vectors. These go by various other names – clock and shift, Sylvester matrices, and others – but in this context, the terms slide, tilt, and phase are the most meaningful ones. Let's go through each of these in turn.

3.2 Slide

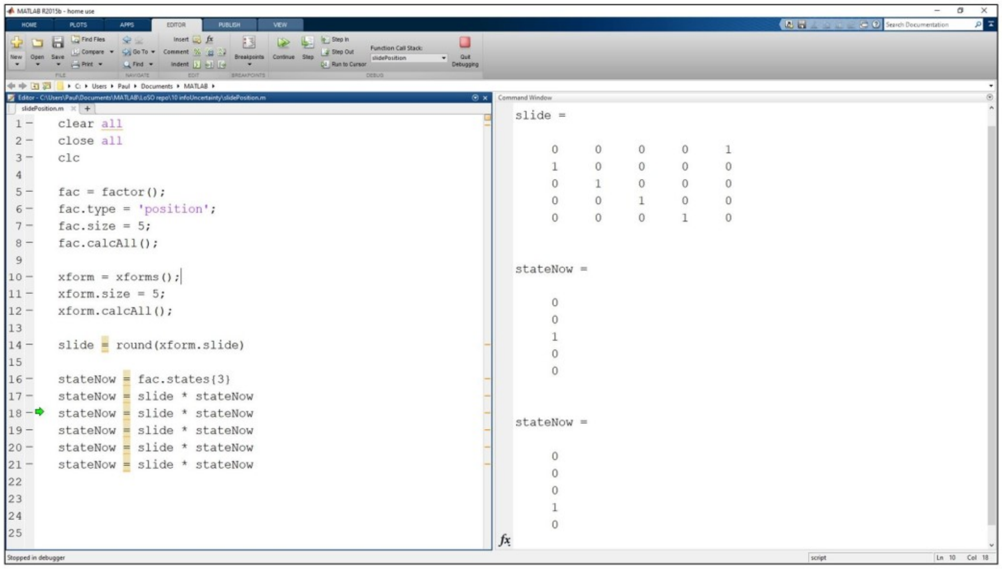


$$S = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The *slide* transformation is physically implemented by an optical slab. This is a thick piece of glass with parallel faces, and it's oriented at an angle. For a moment, we'll briefly think in terms of geometrical optics, which means that we model light as a ray, rather than as a wave. When a ray reaches the surface of the slab, the difference in index of refraction causes the ray to be refracted according to Snell's law, and it begins propagating at a different angle inside the glass. After it has propagated some distance, the ray reaches the other face of the slab and refracts again. It emerges parallel to the original ray, but shifted over laterally to be at a different position.

Mathematically, the starting and ending positions can be represented by two different states of a position factor. The effect of the slab is represented by the slide matrix *S*. The slide matrix looks a little bit like the identity matrix, which would have 1s along the diagonal and 0s everywhere else. The only difference is, every column is shifted over one to the left, and the leftmost column circles around to become the rightmost column. To see how it works, let's look at the companion code.

3.3 Companion code

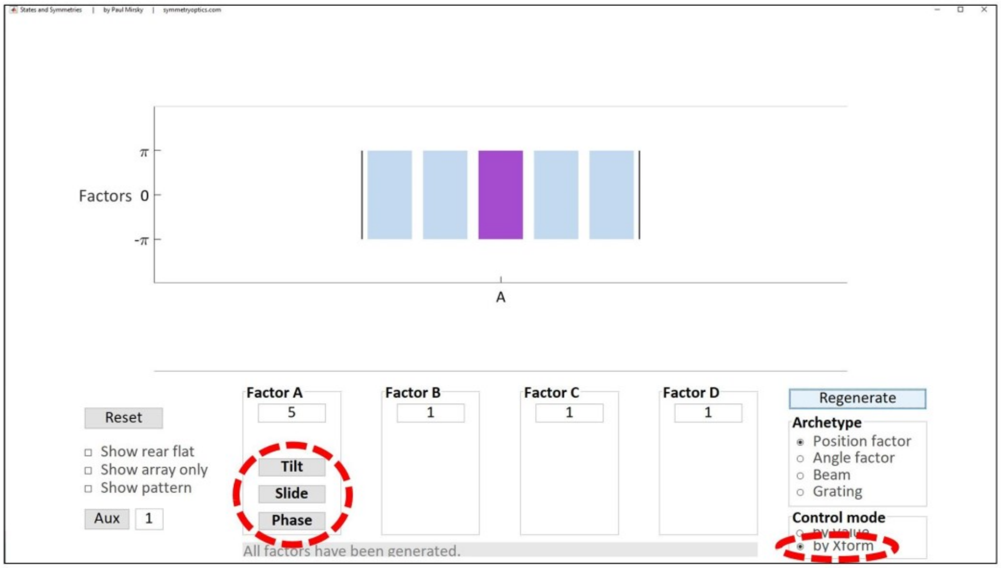


To create the transformations in the companion code, open *slidePosition.m*. First, we create an instance of the factor class, and create a position state. Then, we create an instance of the xforms class. We assign 5 to the size, and then calcAll(). The xform object contains the slide matrix as one of its properties. We're rounding here just to make it more legible; in this case we're simply trimming off a bunch of decimal zeros that don't matter.

The matrix acts on state vectors. To see this, we'll assign the zero-eigenvalue state, which has the 1 in the center, to a variable called *stateNow*. Next, we'll multiply stateNow times the slide matrix, and assign the result back to the variable stateNow. The new stateNow is shifted over one place from the previous; it's the +1 eigenvalue state. We can continue this.

Note that when we slide the +2 vector, it circles back around to become the -2. This is unphysical, but for this idealized model it's OK.

3.4 States and Symmetries app (SnS)

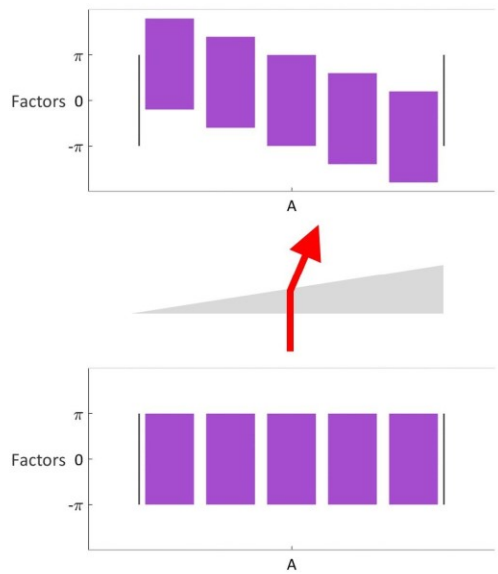


The last lecture introduced the app called States and Symmetries, which is a very useful tool for computing states and patterns, and for visualizing them too.

In the lower-right corner of the app screen, there is a box labeled 'control mode'. For the entire last lecture, we had selected the option 'by value'. That let us choose the value for each factor from a drop-down menu, like this. Now we're going to check 'by transformation' and click regenerate. The menu of values disappears, and instead we now see three buttons, which are labeled tilt, slide, and phase.

When we click slide, it multiplies the current state vector by the slide matrix. The reset button returns the state to the zero-eigenvalue state.

3.5 Tilt



$$T = \begin{bmatrix} u^{-2} & 0 & 0 & 0 & 0 \\ 0 & u^{-1} & 0 & 0 & 0 \\ 0 & 0 & u^{0} & 0 & 0 \\ 0 & 0 & 0 & u^{+1} & 0 \\ 0 & 0 & 0 & 0 & u^{+2} \end{bmatrix}$$

Unit phase

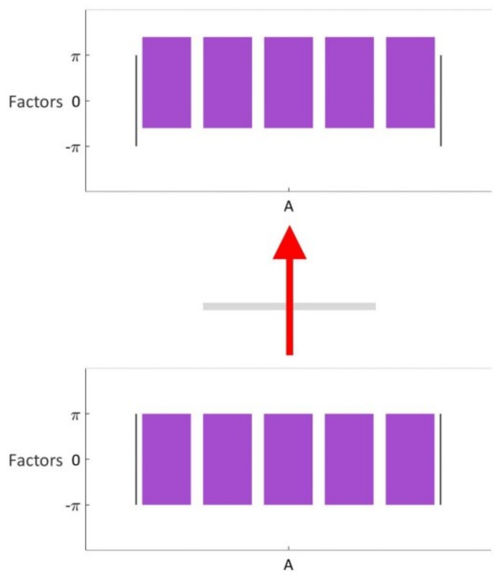
$$u = e^{i \cdot 2\pi/5}$$

We have been discussing the slide transformation. Now we'll move on to the *tilt* transformation. Tilt is physically implemented by an optical wedge. This is a piece of glass with two faces. The first face is perpendicular to the optical axis, the second face is at some angle to it. In terms of geometric optics, a ray passes through the first face without changing angle. But at the second face, the ray is refracted and it continues on, propagating off into space at some new angle.

Mathematically, the starting and ending angles can be represented by two different states of an angle factor. The effect of the wedge is represented by the tilt matrix T. The tilt matrix is diagonal, and each diagonal entry is a different power of the unit phase u, which we discussed in the last lecture in the context of state vectors.

The companion code file *tiltAngle.m* shows how the tilt matrix works on an angle factor. But the States-and-symmetries app is the easiest way to view it. It's perfectly analogous to the way the slide matrix works on a position factor. Each click of the tilt button applies the tilt matrix, and advances the angle state to the next angle. Actually, in this case it increments the angle in the negative direction. It eventually circles back around to the start, which once again is unphysical but works best for this idealization.

3.6 Phase



$$\begin{bmatrix} u^1 & 0 & 0 & 0 & 0 \\ 0 & u^1 & 0 & 0 & 0 \\ 0 & 0 & u^1 & 0 & 0 \\ 0 & 0 & 0 & u^1 & 0 \\ 0 & 0 & 0 & 0 & u^1 \end{bmatrix} \cdot \begin{bmatrix} u^{-2} \\ u^{-1} \\ u^{0} \\ u^{+1} \\ u^{+2} \end{bmatrix} = \begin{bmatrix} u^{-1} \\ u^{0} \\ u^{+1} \\ u^{+2} \\ u^{-2} \end{bmatrix}$$

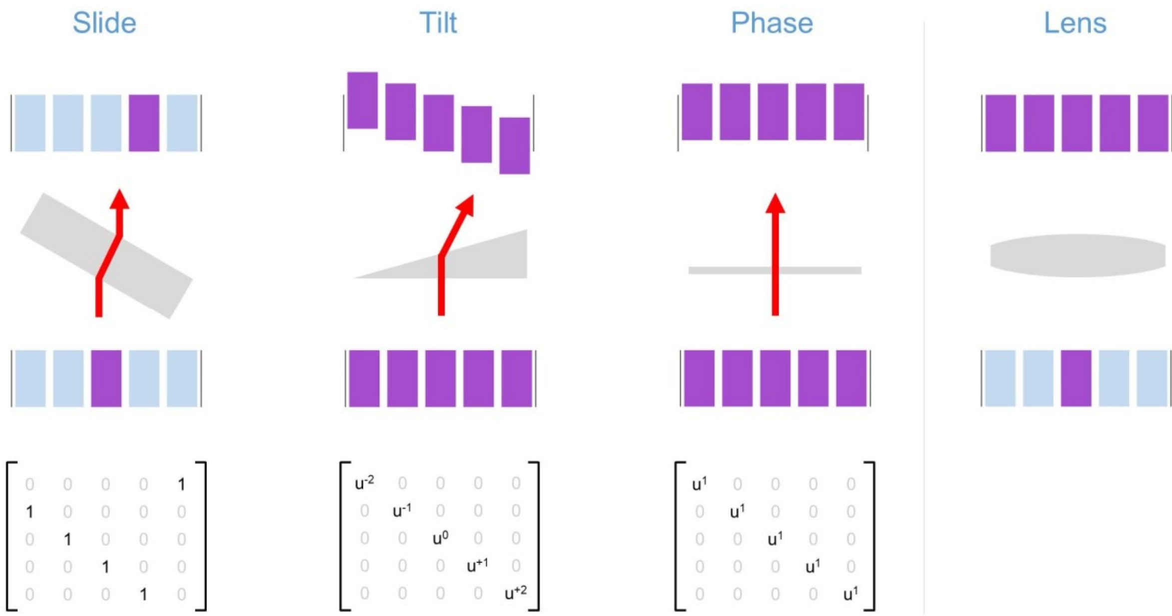
The final basic transformation is the *phase* transformation. Phase is physically implemented by an optical window. This is a thin piece of glass with both faces perpendicular to the optical axis. The glass has a higher index of refraction than the air or vacuum around it, and so the window shifts the phase by some amount, compared to the case with no window.

Mathematically, the effect of the window is represented by the phase matrix P. The phase matrix is diagonal, and all the diagonal entries are the same, and equal to the unit phase. Equivalently, we can say that it's a scalar, times the identity matrix, or even just a scalar. When the phase matrix acts on a state vector, it multiplies all of the components of the vector by a common phase.

For instance, the components of the -1-angle vector have the exponents -2, -1, 0, 1, and 2. When we multiply by the phase matrix, we add 1 to each exponent. This type of state vector was not discussed in the previous lecture, but it is another form of the -1 angle vector, not a different angle.

The SnS app shows the effect visually and intuitively. Each application of the phase matrix causes the boxes to move vertically by 1/5 of a cycle (for a size-5 factor), and after 5 applications it's back where it began. It's the same for any of the angle states. It's also the same for any of the position states.

3.7 Ideal vs actual devices



These are the 3 basic transformations slide, tilt and phase, and each one corresponds to an optical device. But more precisely, these transformations describe *ideal* devices. In practice, actual devices behave differently, and the picture becomes more complex.

Firstly, we described the window a moment ago as a thin piece of glass. The point is, wavelengths are very small and only a very short travel through glass is necessary. For instance, to shift the phase by 0.2 cycles might require a glass optic that is a fraction of 1 micron thick. This is impractical to fabricate or handle, so in practice a much thicker optic would be used. However, the effect would be the same because for a coherent beam it actually doesn't matter whether the phase is shifted by 0.2 cycles, 1.2 cycles, 5.2 cycles, or 1000.2 cycles because the integer cycles have no effect. The only qualification is that the thickness has to be shorter than the coherence length of the light. Also, note that almost nobody actually uses windows to delay phase this way, because it has no meaningful effect, as we will discuss later.

Secondly, an actual wedge and an actual slab also have the effect of a phase factor, in addition to their primary effect. Unless extreme care is taken to control the thickness of the optic, it will effectively contain a window and will cause an overall phase shift. In practice, this care is never taken because nobody cares.

Thirdly, let's clarify a confusing point: these illustrations appear to show the light propagating some substantial distance, passing through the optic, and then propagating some distance again. But for ideal devices, we neglect this distance. For ideal devices, the input, the device, and the output are all in a single plane or at least they are infinitesimally close.

By the way, note that this is different from a lens. We often illustrate the lens-limited configuration with a picture which looks very similar to the others but is actually completely different. For the lens, these planes are the front and rear flats, and the spatial separation is a necessary part of the principle, even for an ideal device.

At this point, we've introduced all of the basic transformations. Next, we'll connect these transformations to the concepts of information and uncertainty.

4 Eigenvectors, eigenvalues, and matrices

4.1 Chart

Effect of this transformation…

… on this type of factor

	Tilt	Slide	Phase
Position factor	Eigenvalue phase	Increment state (+)	Common overall phase
Angle factor	Increment state (-)	Eigenvalue phase	Common overall phase

In this chart, the columns correspond to the transformations tilt, slide, and phase. The rows correspond to position and angle factors. Each cell tells the effect of one transformation on one type of factor.

As we've explained, tilt increments an angle factor from one angle to the next, in the negative direction. Slide works analogously and increments a position factor from one position to the next, in the positive direction.

Also phase multiplies anything by a common overall phase. It's a number on the unit circle. *Overall* phase means, the same number is applied to all entries of the vector. *Common* overall phase means, this same number is applied to all of the different state vectors.

Now we'll discuss the two remaining cells and see that for both, the effect is a state-specific *eigenvalue* phase. It's also a number on the unit circle. It's also an overall phase, meaning that the same number is applied to all entries of the vector. But in this case it's *not* common; instead, each state vector gets a *different* overall phase.

The term *eigenvalue* is German for 'characteristic value'. The idea is, each state corresponds to a different number.

4.2 Hermitian matrices, Observables

Orthogonal vectors	Real Eigenvalue
(1, 0, 0, 0, 0)	-2
(0, 1, 0, 0, 0)	-1
(0, 0, 1, 0, 0)	0
(0, 0, 0, 1, 0)	+1
(0, 0, 0, 0, 1)	+2

Hermitian matrix

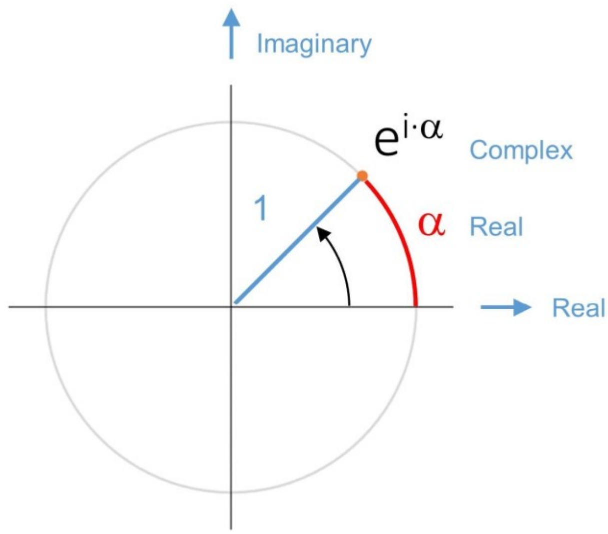
$$\begin{bmatrix} -2 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & 0 \\ 0 & 0 & 0 & 0 & +2 \end{bmatrix}$$

See calcHermitianAndUnitary.m

As an example, here are the position state vectors. Each one corresponds to a position value. Mathematically, a basis of orthogonal vectors and a corresponding set of real eigenvalues are the parameters needed to specify a Hermitian matrix. In other words, if you have these you can compute a Hermitian matrix, and also vice-versa. If you're interested in the details, check out *calcHermitianAndUnitary.m* in the companion code.

In quantum mechanics, a Hermitian matrix represents an observable. If you measure or observe an object, you find it in one of the eigenstates of the observable matrix, and its value is the corresponding real eigenvalue. But our main interest is actually not observable matrices per se, but rather something slightly different.

4.3 Real and imaginary phase



Recall that any number on the unit circle has a complex value, which contains a real and an imaginary part. But because that number is constrained to lie on the unit circle, you can also parameterize it with a real number α which is the length of the red arc. The real number and the complex number correspond 1:1; if you know one, you can easily calculate the other. The only qualification is, if the real number exceeds 2π, it becomes 0 again. So, it's a real number modulo 2π.

4.4 Unitary matrices, Symmetry, time evolution

Orthogonal vectors	Real Eigenvalue	Complex Eigenvalue
(1, 0, 0, 0, 0)	-2	u^{-2}
(0, 1, 0, 0, 0)	-1	u^{-1}
(0, 0, 1, 0, 0)	0	u^{0}
(0, 0, 0, 1, 0)	+1	u^{+1}
(0, 0, 0, 0, 1)	+2	u^{+2}

Unitary matrix

$$\begin{bmatrix} u^{-2} & 0 & 0 & 0 & 0 \\ 0 & u^{-1} & 0 & 0 & 0 \\ 0 & 0 & u^{0} & 0 & 0 \\ 0 & 0 & 0 & u^{+1} & 0 \\ 0 & 0 & 0 & 0 & u^{+2} \end{bmatrix}$$

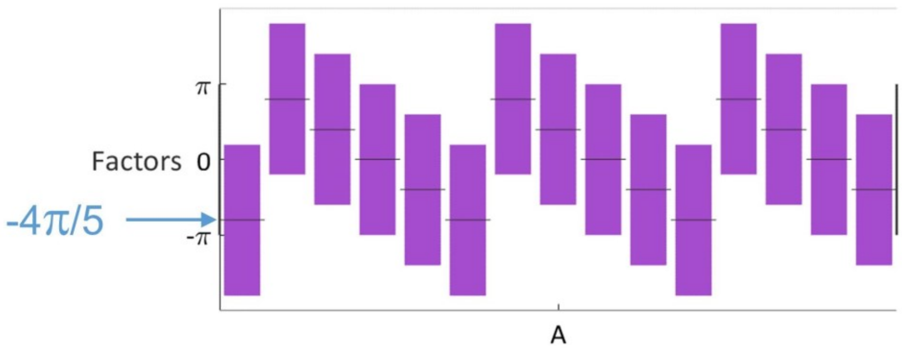
See calcHermitianAndUnitary.m

If we apply this to the real eigenvalues of an observable matrix, we can compute complex eigenvalues. Mathematically, the orthogonal vectors and a corresponding set of unit-circle eigenvalues specify a unitary matrix, and vice-versa.

Tilt, Slide, and Phase are all unitary matrices.

If you multiply a matrix times an input vector, you get some other vector as an output. But out of all possible vectors, there are a handful of special ones which happen to be eigenvectors. If you happen to choose an eigenvector of the matrix as the input to the matrix, then the output will be the same vector as the input, but multiplied by an overall constant. That constant is the eigenvalue which corresponds to that eigenvector.

4.5 Slit applied to angle states



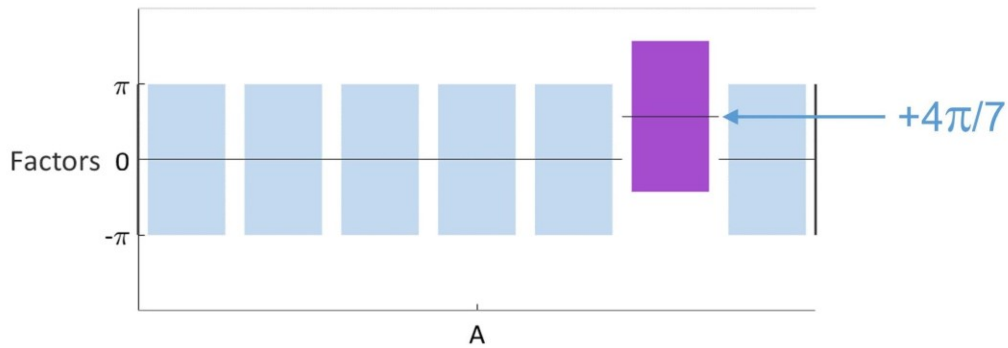
Let's see an example. Here's an angle state. The phase goes through 3 cycles, so the angle is +3. The slide transformation shifts every component to the next position.

The eigenvalue for this angle is u^{+3}. Here's one way to understand that number: If we begin with this state and apply slide exactly once, the very first component moves to the phase -4π/5. To show the phase of each component more clearly, we have the center lines temporarily turned on.

Next we return to the original state, and this time instead of clicking slide, we click phase – and we find that by clicking phase 3 times, we arrive at the same state as when we clicked slide once. Each phase click is a factor of u, so the net effect is u^{+3}, which is the eigenvalue. In other words, slide is equivalent to 3 phase shifts, for this angle state. You can easily generalize this to all the other states and their values.

Angle 0 seems like it might have been a natural example to start with, but actually the opposite is true – it's a terrible example. Because when you apply slide, it appears to have no effect at all. But actually, it is having the trivial effect of multiplying the entire state by u^0, which is 1, and which does nothing.

4.6 Tilt applied to position states



The situation is perfectly analogous when a tilt is applied to a position state. Here is a position state at position +2, or eigenvalue u^{+2}, and at phase 0. Next, we apply the tilt matrix. The phase is increased to 4π/7. If we revert to position +2 at 0 phase, we find that we can reach the same end state by clicking the phase button 2 times.

For position -1, tilting is equivalent to -1 phase shifts, etc. We can extrapolate from these to formulate the general rule, which is the same as the rule for angle factors: the effect of the tilt matrix on any position state is to multiply the state vector by the state's own eigenvalue.

5 Subgroups for optical factors

5.1 Info and uncertainty in the chart

Uncertainty

	Tilt	Slide	Phase
Position factor	Eigenvalue phase	Increment state (+)	Common overall phase
Angle factor	Increment state (-)	Eigenvalue phase	Common overall phase

Information

We've discussed the three basic transformations, and how they work on different kinds of factors. Now let's connect them to the topics of information and uncertainty which we discussed in the first part of the lecture. We need to identify which transformations are distinguishable, and which are indistinguishable.

The key rule is that an *overall phase change cannot be measured* and is an indistinguishable change. Overall phase change includes all the cells highlighted in green – the common overall phases, and the state-specific eigenvalue phases. These together constitute the uncertainty. Incrementing the state to the next state is distinguishable, and therefore the cells highlighted in yellow constitute the information.

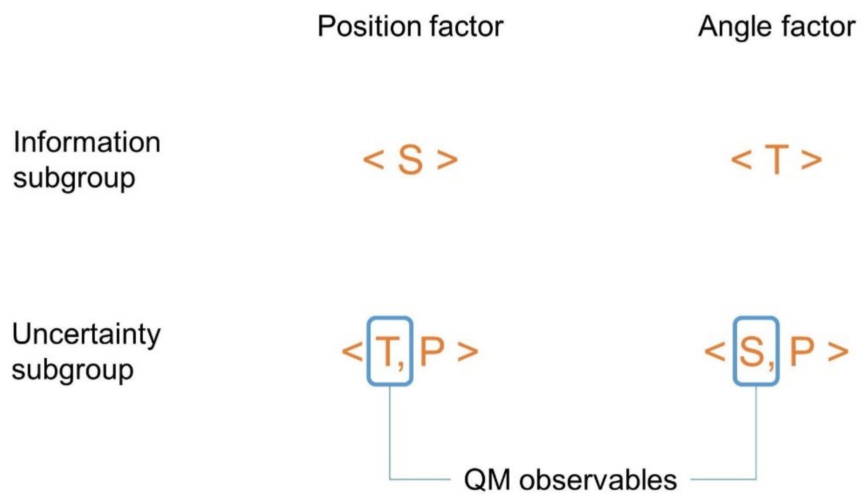
5.2 Overall phase

One way to think about indistinguishable phase is that waves of light are passing through a stationary plane. Looking at the amplitude of the wave in that plane, the phase advances linearly in time. Effectively, it is as if the phase transformation is being applied over and over at every instant. This phase change happens at an extremely high frequency, because the numerator is the speed of light, which is enormous, and the denominator is the wavelength, which is typically tiny. In practice, there is no way to measure electric or magnetic fields so rapidly and so the phase is unobservable.

Even though it is possible to create interference patterns which remain stable, those demonstrate *relative* phase between different waves, rather than overall phase. And, even that stable interference pattern itself has an overall phase that oscillates.

In quantum mechanics, observing phase is not only impossible in practice, but even impossible in theory. It just falls out of the math of quantum mechanics that an overall phase change does not affect observables. It even suggests that ontologically, the overall phase might not even physically exist.

5.3 Factor information and uncertainty



This chart summarizes our analysis of optical factors. For a position factor, slide generates the information subgroup, while tilt and phase generate the uncertainty subgroup. For an angle factor, tilt generates the information subgroup, while slide and phase generate the uncertainty subgroup.

By the way, from the name 'observable', you might assume that quantum-mechanical observable matrices correspond to information generators. But actually, they do not. The observables, or rather their complex-eigenvalue equivalents, are uncertainty generators.

5.4 Example: list of information matrices

Information subgroup

< S >

= S^L for all L = { -1, 0, 1 } for size 3

S^{-1}
S^{0}
S^{+1}

We have notated these groups in terms of their generators, but each of these groups can also be represented as a table of matrices. Here's one example – a position factor, size n = 3. You can easily generalize to angle factors, and to arbitrary sizes.

The information subgroup is generated by slide, so for a size-3 position factor the group is represented by these 3 matrices which are various powers of slide. If we apply slide n times, it gets back to the identity, so there are only n transformations in the group. In this table, the exponents are centered around 0, but effectively it's the same as exponents running from 1 to n.

5.5 Example: list of uncertainty matrices

Uncertainty subgroup

$< T, P >$

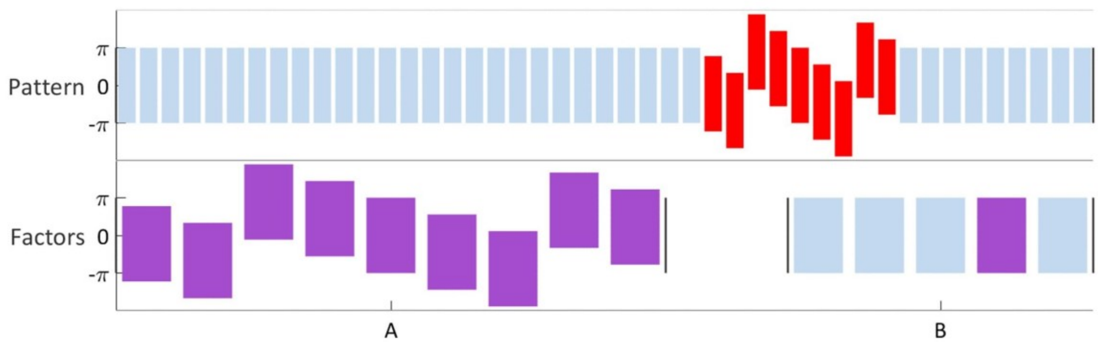
$= T^M \cdot P^Q$ for all M, Q = { -1, 0, 1 } for size 3

$T^{-1} \cdot P^{-1}$
$T^{-1} \cdot P^{0}$
$T^{-1} \cdot P^{+1}$
$T^{0} \cdot P^{-1}$
$T^{0} \cdot P^{0}$
$T^{0} \cdot P^{+1}$
$T^{+1} \cdot P^{-1}$
$T^{+1} \cdot P^{0}$
$T^{+1} \cdot P^{+1}$

The uncertainty subgroup has two generators: tilt and phase. Each of them can be applied individually up to n times, and we count their applications independently. This table lists every possible combination of tilt and phase.

What's very interesting is that in terms of complex numbers, there are only 3 different states. But the size of this uncertainty group is n^2, or 9 possible net transformations. This suggests a tension between symmetry optics and quantum mechanics, which is not currently understood. Indeed, this tension seems likely to deepen before it gets resolved.

5.6 Beam in SnS



Up to this point, we have only discussed groups for a single factor. But of course, multiple factors can combine to make compound systems like beams and gratings.

Here is an example of a beam, which is composed of an angle factor A and a position factor B. Each factor has its own individual tilt, slide, and phase transformations. Each factor transforms entirely independently of all the others. The transformations on A have no effect on B, and vice-versa.

Tilting Factor A, and sliding factor B change the beam to a different states. But sliding factor A makes only a phase difference. Same with tilting factor B. And of course, either of the two phases only make a phase difference.

5.7 Beam subgroups

Information subgroup $< T_A, S_B >$

$= T_A^M \cdot S_B^N$ for all angles M, all positions N

Uncertainty subgroup $< S_A, P_A, T_B, P_B >$

$= S_A^W \cdot P_A^X \cdot T_B^Y \cdot P_B^Z$ for all positions W, all phases X, all angles Y, all phases Z

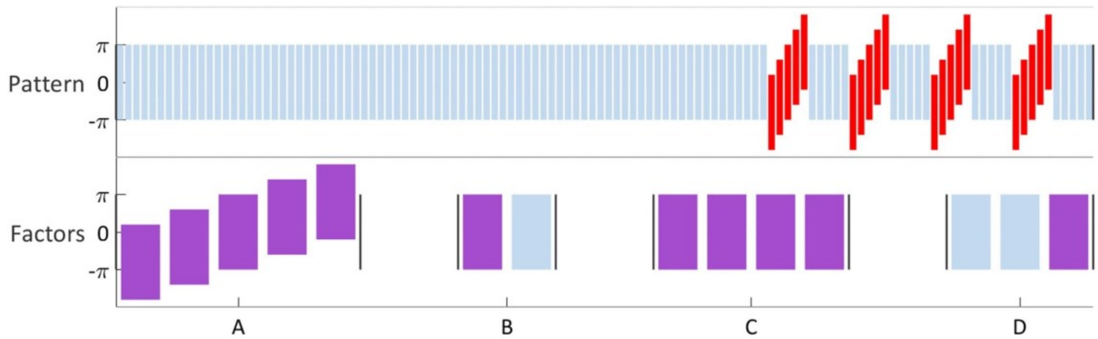
We can write the information and uncertainty subgroups by their generators.

The information subgroup is generated by tilt on factor A and slide on factor B.

The uncertainty is everything else: Slide and Phase on factor A, and Tilt and Phase on Factor B. For the example on the last slide, the beam can be at 9 different angles and 5 different positions. In other words, the information subgroup consists of all possible combinations of some power of tilt on A, times power of slide on B, for a total of 9·5 or 45 different information transformations.

The uncertainty subgroup consists of all combinations of A's and B's uncertainty transformations. This means any of 9 possible slides on A, any of 9 possible phases on A, any of 5 possible tilts on B, and any of 5 possible phases on B. This is a total of 9·9·5·5 or 2025 different transformations.

5.8 Grating in SnS



To analyze the grating, we simply extrapolate further to more factors. A grating is composed of an angle factor A, a position factor B, angle factor C, and position factor D.

There are four information transformations: tilt A, slide B, tilt C, and slide D.

There are 8 uncertainty transformations: slide A, tilt B, slide C, and tilt D – plus phase A, phase B, phase C, and phase D.

5.9 Grating subgroups

Information subgroup $< T_A, S_B, T_C, S_D >$

Uncertainty subgroup $< S_A, P_A, T_B, P_B, S_C, P_C, T_D, P_D >$

The information transformations generate this subgroup

The uncertainty transformations generate this subgroup.

6 Conclusion

6.1 Reviewing key points

That's all for this lecture, so let's review the key points:

- The full group of an object consists of all possible net transformations that can be done to the object.
- The uncertainty subgroup consists of those transformations which change the object to a different unobservable microstate.
- The information subgroup consists of those transformations which would change the object to a different observable macrostate.
- For a position factor, the information is generated by slide, while the uncertainty is generated by tilt and phase.
- For an angle factor, the information is generated by tilt, while the uncertainty is generated by slide and phase.

6.2 Outro

I hope you've found this class informative and interesting. To learn more about symmetry optics, please check out www.symmetryoptics.com. If you have specific questions about this or other lectures, please post them on Reddit at www.reddit.com/r/symmetryOptics/, and I'll try to answer them.

This is a new field, and there's a lot of opportunity to discover new science and develop new applications. I hope you'll take advantage.

I'm Paul Mirsky, thanks for listening.

6.3 References

Miller, Scott J. *Asymmetry: The Foundation of Information.* Springer-Verlag, 2007