**Eigenfunctions/values of**

So going back to:



We ascertained what the eigenfunctions/values were, in principle. But we were particularly interested in writing down simultaneous eigenfunctions of the operators L2, S2, J2, Jz in terms of the simultaneous eigenfunctions of L2, S2, Lz, Sz. We’re calling this |ℓsjmj>. And we want to solve the equations:



We argued in the last file that for a given ℓs, j can run between |ℓ-s| and ℓ+s, and that mj runs between -j and j for each j. So we can take a given j and mj as being specified. So then we’ll suppose we have j and mj specified. Then the question is, what does this eigenstate look like in terms of the uncoupled basis? The generic form would be:



And we want to know what cmℓ,ms are. The coefficients cmℓ,ms are called the Clebsch-Gordon coefficients by the way. There are tables which list the various values for a given j and mj, but we will attempt to determine them ourselves. We’ll set about answering this question via a ‘differential’ approach to solving the eigenvalue equation. Our differential equation will be more like a difference equation rather, but the spirit of this approach is similar to the one with which we’ve written down eigenvalue equations in the past.

**‘Differential equation’ approach to solving for eigenvectors/values of coupled representation**

So the eigenfunctions can be written as:



Let’s start out by applying the Jz operator. The eigenfunctions would obey the equation:



For this equation to be true, we see that whatever mj is, cmℓ,ms must be 0 if mℓ + ms ≠ mj. So the sum over mℓ and ms is restricted to only those values for which mℓ + ms = mj. So presently, we have a condition:



and we can write:



Now let’s fill this into the J2 equation.



Projecting this equation onto the orbital/spin momentum basis (the uncoupled basis in other words) we have:



So we have the eigenvalue/vector equation to solve, switching the ′ and un′ indices for prettiness.



Maybe progress can be made by filling in the **L** and **S** vectors, and then evaluating the action of Sz and Lz at least.



We can evaluate the matrix elements by writing Lx, Ly and Sx, Sy in terms of the L and S raising and lowering operators…



Let’s recall the action of creation/annihilation operators on an L, S eigenket. And for future concision make a parameter definition Cℓ,mj±.



and similarly for S. Before we work this out, let’s observe that the only terms which will survive the sum are the L+S-, L-S+ terms. This is because the sum is over those terms which have the same value of mℓ′ + ms′, and to keep the sum the same we would have to raise and lower mℓ′ and ms′ by opposite amounts. So this simplifies our expression to (factoring out the 1/4):



The δ functions collapse the sums, and leaves us with,



This is the equation for the Clebsch-Gordon coefficients. The equations are a little too difficult to solve in generality I think, so let’s take a special case.

**Solution of the Clebsch Gordon recursion relation for ℓ = 1, s = ½.**

The simplest non-trivial case we could get away with is where ℓ = 1, and s = ½. Then mℓ = -1, 0, 1 and ms = -½, ½. So there are 3×2 = 6 independent basis states. So our Clebsch-Gordon recursion equation will be a 6×6 matrix equation. Let’s write out the equation for each (mℓ, ms) pair and then we’ll write everything as a big matrix equation and solve for the coefficients.

***Case: mℓ = 1, ms = ½.***

Then we have:



We could say something about this, but let’s keep going.

***Case: mℓ = 1, ms = -1/2***

Then we have:



Filling in the creation/annihilation operator coefficients we have:



and so we have:



We’ll leave this alone for now and go on to the next,

***Case: mℓ = 0, ms = 1/2***

Then we have:



Filling in the creation/annihilation operator coefficients we have:



Filling these in we have:



Next,

***Case: mℓ = 0, ms = -1/2.***

Then we have:



Evaluating the creation/annihilation operator coefficients…



Filling these in we get:



Next,

***Case: mℓ = -1, ms = ½***

Then we have:



Filling in the operator coefficients…



Filling these in…



And lastly,

***Case: mℓ = -1, ms = -1/2***

We have:



C’est finit.

***Putting equations into matrix form and solving***

So let’s gather up our equations:



which can be written as:



Let’s put these in matrix form,



Now in order for there to be non-zero solutions for the coefficients, we need the determinant of our matrix to be zero. Since this is a block diagonal matrix, the determinant is just the product of the determinant of the sub-blocks. So:



The solutions to these equations are:



The values of j are consistent with our argument that j could range between |ℓ-s| = 1 – ½ = ½ and ℓ+s = 1 + ½ = 3/2. Okay so let’s examine each state and see what we get:

***States: j = 3/2, mj = whatever***

So if we fill j = 3/2 into our matrix, we’ll get this:



This block diagonal matrix then requires:



Adding up the mℓ and ms subscript values of these coefficients to get mj, we can see clearly which |ℓsjmj> states these coefficients belong to.



So we can write, normalizing things::



The top one is interesting since it says that when the particle is in its maximum Lz and Sz state, then it is also in its maximum Jz and J2 state. Before we were saying that if we know Lz and Sz, then we couldn’t know J2 because the J2 operator doesn’t commute with Lz and Sz. And this is true in general, but we see that it is not true for *every* single state.

***States: j = 1/2, mj = whatever***

Now let’s fill j = 1/2 into our matrix, we’ll get this:



This block diagonal matrix then requires:



Adding up the mℓ and ms values in the cmℓ,ms coefficients, we can see which |ℓsjmj> states are associated with them::



So normalizing, we get:



upon normalization. So we see that mj can take values of -1/2, and ½, consistent with that fact that mj = -j, -j+1, …, j.

**Summary of ℓ = 1, s = ½ results**

Summarizing then, our results are that for the case ℓ = 1, s = ½, j can take on two values, namely j = ℓ + s = 3/2, and j = ℓ-s = ½. And in either case, mj = -j, -j +1, …, j. And in terms of the uncoupled basis, the eigenvectors are:



We can put this in the position/spin basis by projecting the eigenvectors against <**r**|<ms| to get:



So we get our first taste of wavefunctions that have a mixing between spin and spatial components. This might seem weird at first, but it is no different than a wavefunction mixing x and y coordinates. Of course *that* might seem weird to someone who only lives in 1 dimension.

**Appendix: Result for ℓ arbitrary, s = 1/2**

Well reviewing what we did, and observing that our matrix was block diagonal, it occurs to me that we *can* actually solve the equations in general, without too much work, for ℓ arbitrary and s = ½. This would involve diagonalizing a 2×2 matrix. In general we can solve for s arbitrary, but would have to diagonalize a (2s+1)×(2s+1) matrix. So we want,



So for given ℓ and s, j can be either ℓ - s or ℓ + s. mj can range between -j and j of course. And note since ms = ±s = ±1/2, for a given mj, mℓ can take on two values at most, namely mj – s, or mj + s. So we can say,



where again, s = ½. Then let’s go back to our Clebsch-Gordon recursion equation:



where I’ll recall,



and examine the equation for these two coefficients,



and cross out the two terms there because we cannot have any coefficients with spin indices greater than ½ or less than -1/2. And also recognize that the same colored coefficients are the same. We can now write, simplifying some of those subscripts, and the [ ] thing:



and then, using s = ½ intermittently,



And so,



Let’s see what those C terms are:



So they’re the same. Now our equations are:



In matrix form, looks like:



For a non-zero solution to these equations, these rows must be linearly dependent, and must be so for any of our allowed values of j and mj. Or in other words, the determinant of our matrix must be zero. Let’s check,



where we’ve used s = ½ again. Let’s fill in j = ℓ±s, and we’ll liberally use s = ½.



So this checks out. That’s nice. So take note of this identity, the consequence of the bottom line in that block of equations just above the set immediately above:



So going back to our matrix and eliminating the bottom row of our matrix equation, say, via row operations, the solution to our matrix equation is:



where k can be anything. Let’s use our purple identity to simplify this a little bit:



where k´ is just another constant, and we recognize that ℓ(ℓ+1) + s2 = (ℓ+s)2 since s = ½. This better improve fast. Well we should normalize it so we need:



which brings us to:



So can write this as:



This reproduces our earlier results sans an occasional overall factor of -1, but overall phase factors don’t matter. And we’ll remember the ± signs are for j = ℓ ± s, in that order.

**Example**

Consider the electron in the 3𝑑1 orbital of a lone scandium atom. A weak magnetic field is applied in the 𝑧 direction such that the perturbing Hamiltonian is given by: 𝛿𝐻 =𝜇𝐵(𝛼𝐿𝑧+𝛽𝑆𝑧) (α and β are real constants). Assume that when the 𝐽z and 𝐽2 operators are applied to the electron's state, the results are 3ℏ/2 and 35ℏ2/4 respectively. Calculate the first-order correction to the electron's energy due to the perturbing Hamiltonian without using the projection theorem (based on the Wigner-Eckart theorem).

So we know that the electron has quantum numbers s=1/2, ℓ = 2, j = ?, and mj = 3/2. Since j(j+1) = 35/4, we must have j = 5/2 (trial and error, or do quadratic formula). Then we just need to break this |ℓsjmj> state down into |ℓsmℓms> states. Well we can use our formula above:



Filling in our numbers,



j = ℓ + s, so we use the top sign,



Now we’ll use first order perturbation theory (see those files),

