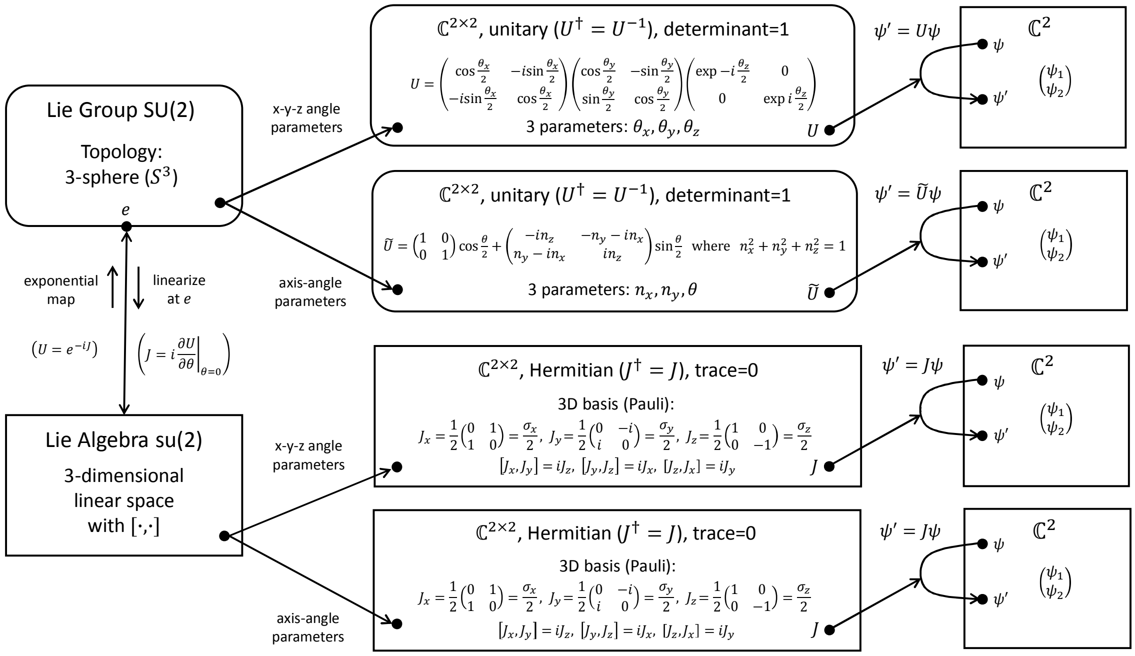
3.4 SU(2): Defining Representation with Axis-Angle Parameters



Let's return to the 2-dimensional or defining representation of SU(2). The upper branch of the diagram shows again the matrix expression that we have seen before, whereas the lower branch shows a new way to parametrize a unitary 2×2 matrix with determinant one. Instead of using the angles of rotation about the x, y, and z axes to specify a transformation, we now use an axis, given by the unit vector with components n_x, n_y, n_z, where $n_x^2 + n_y^2 + n_z^2 = 1$, and the angle of rotation θ about this axis to specify a transformation. The resulting matrix expression is shown in the lower branch of the diagram. This transformation can be rewritten more compactly in terms of Pauli matrices as

$$\tilde{U}(n_x, n_y, n_z, \theta) = I\cos\frac{\theta}{2} - i(n_x\sigma_x + n_y\sigma_y + n_z\sigma_z)\sin\frac{\theta}{2}$$

or, using vector notation, as $\tilde{U}(\vec{n}, \theta) = I\cos\frac{\theta}{2} - i(\vec{n}\cdot\vec{\sigma})\sin\frac{\theta}{2}$, where $\vec{n} = (n_x, n_y, n_z)^T$ and $\vec{\sigma} = (\sigma_x, \sigma_y, \sigma_z)^T$, a vector of 2×2 matrices [QFTGA, Ch. 15; GTNut, Ch. IV.5].

The matrix expressions in the upper and lower branch look very different. Nevertheless, they describe the exact same *set* of matrices (namely, unitary 2×2 matrices with determinant one) and each element in that set has the same neighboring elements (same local topology). It's the *same* representation parametrized in two different ways!

Just like before, this representation has three free parameters. The total number of parameters in the expression is four, n_x, n_y, n_z, θ, but the constraint on the unit vector reduces the number of parameters by one. In other words, we need two parameters (e.g., longitude and latitude) to specify the orientation of the axis in space and one parameter to specify the rotation angle about this axis. Somewhat sloppily, we listed the three free parameters in the diagram as n_x, n_y, and θ, implying that n_z is determined by the constraint (this is not exactly true because the sign of n_z remains ambiguous).

Let's derive the basis generators from our new transformation matrix with axis-angle parameters. The identity transformation is now represented by an arbitrary axis in space, let's take it to be $n_x = 0$, $n_y = 0$, $n_z = 1$, and no rotation, $\theta = 0$. Calculating the derivatives (times i) with respect to n_x, n_y, θ and evaluating them at the mentioned parameter values results in the basis generators 0, 0, $\sigma_z/2$. This is a problem because these generators span only a 1-dimensional space! To get around this issue, we evaluate the derivatives at a small, but nonzero, amount of rotation: $\theta = \varepsilon$. Now, we find the basis generators $\varepsilon\sigma_x/2$, $\varepsilon\sigma_y/2$, and $\sigma_z/2$. After normalizing them such that their eigenvalues become +½ and −½, we arrive at the same basis generators as before: $J_x = \sigma_x/2$, $J_y = \sigma_y/2$, and $J_z = \sigma_z/2$.

How do we get from the Lie algebra back to the group representation with axis-angle parameters? Exponentiating each basis generator separately and then multiplying the resulting transformation matrices together takes us to the original parametrization with three angles: $U(\theta_x, \theta_y, \theta_z) = \exp(-i\,\theta_x J_x) \cdot \exp(-i\,\theta_y J_y) \cdot \exp(-i\,\theta_z J_z)$. But what about exponentiating a linear combination of the basis generators, that is, a general element of the Lie algebra, in one shot? In fact, expanding $\exp(-i\,[n_x J_x + n_y J_y + n_z J_z]\theta) = \exp(-i\,[n_x\sigma_x + n_y\sigma_y + n_z\sigma_z]\theta/2)$ into a power series, evaluating the powers, and separating the series into a sine and cosine series yields exactly the transformation matrix $\widetilde{U}(n_x, n_y, n_z, \theta)$ with axis-angle parameters shown in the diagram [QFTGA, Ch. 15]! The reason why this works out so nicely is that the Pauli matrices are very special: they all square to the identity matrix, $\sigma_k^2 = I$, and they anticommute among each other, $\sigma_k\sigma_l + \sigma_l\sigma_k = 0$ for $k \neq l$, knocking out the mixed terms in the above power evaluation.

Can we reformulate higher-dimensional representations of SU(2) in terms of axis-angle parameters? Yes, we can use the same trick and evaluate $\exp(-i\,[n_x J_x + n_y J_y + n_z J_z]\theta)$, where the J_k are now the higher-dimensional basis generators. Unfortunately, these basis generators do not have the special properties of the Pauli matrices and we are not getting pretty expressions.

In quantum mechanics, $J_x = \sigma_x/2$, $J_y = \sigma_y/2$, and $J_z = \sigma_z/2$ are the operators for the spin components of a spin-½ particle along the x, y, and z axes, respectively. But what if we want to determine the spin along an arbitrary axis $\vec{n} = (n_x, n_y, n_z)^T$? Now we know that the appropriate operator is $J_{\vec{n}} = (n_x\sigma_x + n_y\sigma_y + n_z\sigma_z)/2 = \vec{n} \cdot \vec{\sigma}/2$. Why? Because $J_{\vec{n}}$ is the generator of rotation about the $\vec{n}$ axis, that is, $\exp(-i\,\theta J_{\vec{n}})$ rotates the spin state by the angle θ about this axis.

Going back to the original parametrization with three angles, you may have wondered why we multiplied the matrices in the particular order $\exp(-i\,\theta_x J_x) \cdot \exp(-i\,\theta_y J_y) \cdot \exp(-i\,\theta_z J_z)$ as opposed to, for example, $\exp(-i\,\theta'_y J_y) \cdot \exp(-i\,\theta'_x J_x)\,) \cdot \exp(-i\,\theta'_z J_z)$. Matrix multiplication generally doesn't commute and the two products are different. However, the *set* of all matrices described by the two products is the same, the difference is only in how its elements are parametrized. Multiplying the three matrices in any order yields the *same* representation of SU(2).