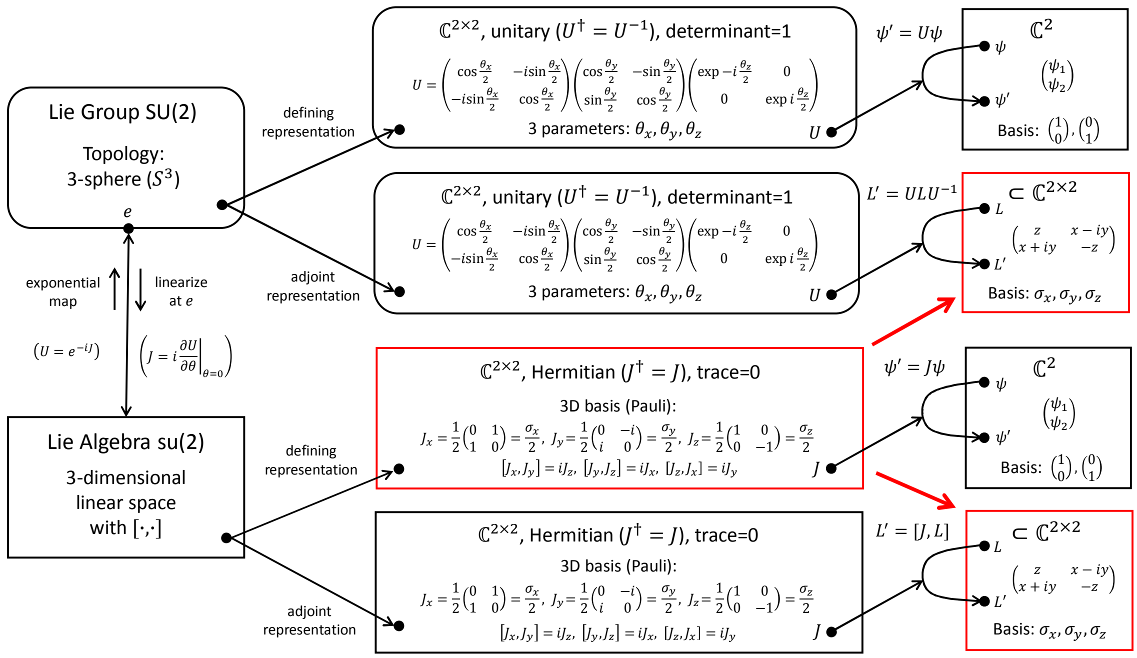
3.8 SU(2): Adjoint Representation; Transforming the Generators



For any Lie group there is a distinguished representation known as the *adjoint representation*. We know that every Lie group has a Lie algebra associated with it and that this Lie algebra is a vector space. The adjoint representation is the representation on this automatically available vector space. In other words, the adjoint representation acts on the generators!

For SU(2) the Lie algebra is three dimensional (it has three basis generators), hence the adjoint representation is also three dimensional. The diagram shows how the representation space of the adjoint representation is a copy of the (defining) Lie-algebra representation (red arrows). To keep the elements of this representation space distinct from the elements of the Lie algebra, we rename them from J to L. Note that instead of the usual column vectors, we now have matrices in the representation space! But these matrices together with the basis matrices, $\sigma_x, \sigma_y, \sigma_z$, form a regular 3-dimensional vector space. In particular, we can linearly combine the three basis matrices (using real numbers) to form a general element in the representation space. Later we will “unpack” these matrices into column vectors, bringing the representation into a more familiar form. (Note that we multiplied the basis matrices $J_i = \sigma_i/2$ by two to get rid of the factor ½; this is allowed because the basis of the representation space does not need to satisfy commutation relations.)

How does the adjoint representation of the Lie group act on the matrix L? It doesn’t act by simple matrix multiplication, as one might naively guess, but by means of the operation $L' = ULU^{-1}$ known as *conjugation* (also written as $Ad_U(L) = ULU^{-1}$). To understand why this is the correct operation, we need to consider two things: (i) For any given U, the map $G' = UGU^{-1}$ is a structure-preserving map (homomorphism) from the group to itself because $U(GH)U^{-1} = UGU^{-1}\,UHU^{-1}$. Note that U, G, H are all elements of the group. (ii) The map $G' = UGU^{-1}$ sends the identity element and its neighborhood to themselves. But this is exactly where the linear (tangent) space, that is, the Lie algebra, touches the group manifold! Thus, the above map induces a structure-preserving map from the algebra to itself given by $I + \varepsilon L' = U(I + \varepsilon L)U^{-1}$, or, after simplifying, $L' = ULU^{-1}$.

How does the adjoint representation of the Lie algebra act on the matrix L? Given a one-parameter set of group elements, we can find the corresponding algebra element by taking the derivative with respect to the parameter, setting the parameter to zero (assuming zero parametrizes the identity), and multiplying the result by i. For example, for a group element that acts like $U\psi = e^{-iJ\theta}\psi$, we find that the corresponding algebra element acts like $i[(-iJ)e^{-iJ\theta}\psi]_{\theta=0} = J\psi$, which is of course exactly what we would expect. Now returning to the adjoint representation, where a group element acts like $ULU^{-1} = e^{-iJ\theta}Le^{iJ\theta}$, we find that the corresponding algebra element acts like $i[(-iJ)e^{-iJ\theta}Le^{iJ\theta} + e^{-iJ\theta}L(iJ)e^{iJ\theta}]_{\theta=0} = JL - LJ = [J, L]$, where we used the product rule. Thus, the adjoint representation of the Lie algebra acts by means of the matrix commutator $L' = [J, L]$ (also written as $ad_J(L) = [J, L]$). Remarkably, the matrix commutator appears again! As a Lie bracket, it measures the (second-order) deviation from perfect commutation of two Lie-group elements. As the action of the adjoint representation, it transforms the Lie algebra. (The fact that the Lie-bracket operation must be preserved in the adjoint representation of the Lie algebra leads to the Jacobi identity.)

To make all this more concrete, it is helpful to study how the adjoint representation of SU(2) acts on the corresponding 3-dimensional column vector. Given the (real) 3-dimensional column vector $(x, y, z)^T$ and the basis matrices σ_x, σ_y, and σ_z we can construct the matrix $L = x\sigma_x + y\sigma_y + z\sigma_z$. Spelled out explicitly, we have $L = \begin{pmatrix} z & x - iy \\ x + iy & -z \end{pmatrix}$ showing how the three vector components get "packed" into the matrix. Next, we act with U on this matrix producing $L' = ULU^{-1}$. Finally, we "unpack" the transformed column vector $(x', y', z')^T$ by comparing L' with $x'\sigma_x + y'\sigma_y + z'\sigma_z$. Specifically, if we choose $U = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}$ and act with it on L, we get $L' = ULU^{-1} = \begin{pmatrix} z & (x - iy)e^{-i\theta} \\ (x + iy)e^{i\theta} & -z \end{pmatrix}$. After unpacking, we find that the three vector components transform like $x' = x\cos\theta - y\sin\theta$, $y' = x\sin\theta + y\cos\theta$, and $z' = z$. What is this? It is a simple rotation of Cartesian coordinates by the angle θ about the z axis!

Next, let's see how the adjoint representation of the Lie algebra su(2) acts on a 3-dimensional column vector. Again, we pack the vector into a matrix, then we act on it with the commutator, and finally we unpack the vector from the transformed matrix. Specifically, if we choose $J = J_z = \sigma_z/2$, we get $L' = [J, L] = [\sigma_z/2, (x\sigma_x + y\sigma_y + z\sigma_z)] = xi\sigma_y - yi\sigma_x + 0$ and, after unpacking, we find $x' = -iy,\ y' = ix$, and $z' = 0$. Indeed, $(x', y', z')^T$ is the displacement produced by a small rotation about the z axis (times i), in agreement with the transformation we found above!

How do the basis generators look when they are rewritten to act on column vectors $\vec{x}$ instead of matrices L? In other words, given that the J_i act like $L' = [J_i, L]$, what are the corresponding $\tilde{J}_i$ that act like $\vec{x}' = \tilde{J}_i\vec{x}$ or, in terms of components, $x'_k = \sum_j [\tilde{J}_i]_{kj} x_j$? Packing the vector components into the L matrix, $L = \sum_j x_j J_j$ and $L' = \sum_k x'_k J_k$, the basis generator J_i in the adjoint representation acts like $\sum_k x'_k J_k = [J_i, \sum_j x_j J_j]$. Moving the sum out of the commutator, we can write $\sum_k x'_k J_k = \sum_j x_j\, [J_i, J_j]$. Using the definition of the structure constants, $[J_i, J_j] = \sum_k c_{ijk} J_k$, we get $\sum_k x'_k J_k = \sum_{j,k} x_j c_{ijk} J_k$. Identifying matching coefficients of J_k, we find that the ith basis generator acts on the vector components like $x'_k = \sum_j c_{ijk} x_j$. Comparing this to $x'_k = \sum_j [\tilde{J}_i]_{kj} x_j$, we discover that the ith basis generator $\tilde{J}_i$ is made up of structure constants as follows: $[\tilde{J}_i]_{kj} = c_{ijk}$!