Exemplary calculations for arXiv: 2401.06064

Fynn Otto, Konrad Szymański

In this notebook we present an example of calculations involving characteristic polynomials – the content of the Example 2 of the paper arXiv:2401.06064. First, we will recreate the polynomial representation of SU(2): that is, the *representation acting on bivariate homogenous complex polynomials* (interpreted as a vector space). It isn't immediately obvious how polynomials corresponds to spin state vectors in quantum mechanics, and we translate the results to obtain a polynomial parameterization in the usual spin space spanned by states of definite total angular momentum, $\{|\,j, m\rangle : m = -j, \ldots, j\}$. This is then applied to provide a polynomial description of SU(2) characteristic functions, and to recreate the results presented in the paper.

In the following code, U (and V) denotes u* (v*), where the star is the complex conjugate. In the analytical sense, the variable and its conjugate are independent, just like real and imaginary parts of a complex number are – and it's much easier to manipulate expression in this way.
The polynomial representation works on – no surprise here – polynomials. A polynomial corresponding to a state $|\,\psi\rangle = (\psi_{-j}, \ldots, \psi_{j-1}, \psi_j)$ in irreducible representation of total angular momentum j (dimension 2j+1) is

```
In[∘]:= statetopoly[j_, ψ_, {z1_, z2_}] :=
  Sum[(-1)^(j - m) Sqrt[Binomial[2 j, j - m]] × ψ[[(j - m) + 1]] z1^(j + m) z2^(j - m), {m, -j, j}]
  (* Majorana polynomial for a given state ψ*)
```

Now, the polynomial is transformed by the action of SU(2) unitary by right multiplication of its complex variables with $\begin{pmatrix} u & -v \\ v^* & u^* \end{pmatrix}$. This is the transformed polynomial:

```
In[∘]:= transformpoly[poly_, {z1_, z2_}] := poly /. Thread[{z1, z2} → ({z1, z2}.( u  -v
                                                                             V   U ))]
```

From this, the coefficients of SU(2) representation parameterized with polynomials can be extracted. The procedure is the following: take any orthonormal basis, express its elements as polynomials, transform using the recipe above, and turn the transformed polynomials back into states.

In[∘]:=
```
getcoeff[j_, poly_, {z1_, z2_}] := Module[{coeff},
  coeff = Reverse@CoefficientList[poly, {z1, z2}, {2 j + 1, 2 j + 1}];
  coeff = Diagonal[coeff];
  coeff = coeff * Table[(-1)^(i + 1) / Sqrt[Binomial[2 * j, i - 1]], {i, 2 j + 1}]]
(*state for a given polynomial*)

SU2Unitary[j_] := Module[{basisstates, basispolys, z},
  basisstates = IdentityMatrix[2 j + 1];
  basispolys = transformpoly[statetopoly[j, #, {z1, z2}], {z1, z2}] & /@ basisstates;
  getcoeff[j, #, {z1, z2}] & /@ basispolys](*Hilbert state picture of
 the transformation defined in g: how are basis states transformed?*)
SU2Unitary[jmin_, jmax_] :=
 DiagonalMatrix[Hold /@ Table[SU2Unitary[j], {j, jmin, jmax, 1 / 2}]] // ReleaseHold //
  ArrayFlatten
```

It can be verified that the resulting matrix is indeed an unitary, provided the complex vector (u, v) is normalized:

In[∘]:=
```
SU2Unitary[1] // MatrixForm
FullSimplify[%.%† /. {V → v*, U → u*}, Assumptions → {Abs[u]^2 + Abs[v]^2 == 1}] // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} u^2 & -\sqrt{2}\ u\ V & V^2 \\ \sqrt{2}\ u\ v & -\frac{-\sqrt{2}\ u\ U+\sqrt{2}\ v\ V}{\sqrt{2}} & -\sqrt{2}\ U\ V \\ v^2 & \sqrt{2}\ U\ v & U^2 \end{pmatrix}$$

Out[∘]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Having a polynomial parameterization of SU(2) unitaries for arbitrary angular momentum, let’s recreate the Example 2 from the article: finding the probability of interconversion. First, some helper functions defining variables and matrices used in the semidefinite optimization:

In[∘]:=
```
SpinDensityOperator[variable_, jmax_] :=
 With[{indices = Flatten@Table[m^j, {j, 0, jmax, 1/2}, {m, -j, j}]},
  Table[variable_{Sequence@@Sort@{indices[[i]],indices[[j]]},Re} +
     Sign[j - i] I variable_{Sequence@@Sort@{indices[[i]],indices[[j]]},Im},
   {i, Length@indices}, {j, Length@indices}]]
SpinDensityOperatorVariables[variable_, jmax_] :=
 With[{indices = Flatten@Table[m^j, {j, 0, jmax, 1/2}, {m, -j, j}]},
  Flatten@{Table[variable_{Sequence@@Sort@{indices[[i]],indices[[j]]},Re}, {i, Length@indices},
     {j, i, Length@indices}], Table[variable_{Sequence@@Sort@{indices[[i]],indices[[j]]},Im},
     {i, Length@indices}, {j, i + 1, Length@indices}]}]
MaxDim[jmax_] := (1 + jmax) (1 + 2 jmax)(*==Sum[2j+1,{j,0,jlim,1/2}];*)
SliceMatrix[mat_] := Module[{jmax},
   jmax = 1/4 (-3 + Sqrt[1 + 8 Length[mat]]);
   Table[mat[[Range[MaxDim[j - 1/2] + 1, MaxDim[j]],
      Range[MaxDim[j - 1/2] + 1, MaxDim[j]]]], {j, 0, jmax, 1/2}]
  ] /; SquareMatrixQ[mat] ∧ IntegerQ[1/2 (-3 + Sqrt[1 + 8 Length[mat]])]
```

Now, let's define the initial ($|\psi\rangle$), target ($|\phi\rangle$), and auxiliary states α and β (which are subject to optimization). The characteristic functions have to coincide: $\chi_\psi = \chi_\phi\,\chi_\alpha + \chi_\beta$. Since we parameterized the SU(2) representations with polynomials, characteristic functions also have this form, and we can compare the coefficients.

There is some ambiguity (see theorem X) which has to be removed: polynomials differing by $u\,u^* + v\,v^* - 1$ should be treated as equal, since with the SU(2) parameterization constraints this polynomial evaluates to zero. This is provided by PolynomialReduce function.

(Note that the initial and target state has to be pure)

```
In[•]:= jlim = 3/2; dim = Sum[2 j + 1, {j, 0, jlim, 1/2}];
SU2 = SU2Unitary[0, jlim];

ψ = Normalize@{1, (*j=0*)
     0, 0, (*j=1/2*)
     0, 1, 0, (*j=1,m=0*)
     0, 0, 0, 2(*j=3/2,m=3/2*)
   };
ψ = PadRight[ψ, dim];

ϕ = Normalize@{0,
     1, 0(*j=1/2,m=-1/2*)
   };
ϕ = PadRight[ϕ, dim];

χψ = ψ*.SU2.ψ; (* characteristic function of the initial state*)
χϕ = ϕ*.SU2.ϕ;(*... and the final state*)

α = SpinDensityOperator[alpha, jlim];
(*auxiliary state when transformation successful -- free variable, to be determined*)
χα = Tr[α.SU2];(*... and its characteristic function*)
β = SpinDensityOperator[beta, jlim]; (*end state when transformation unsuccessful*)
χβ = Tr[β.SU2];(*... and its characteristic function*)


charfnequality = χψ - χϕ χα - χβ;
(*constraint: the characteristic function of the initial state should be
  equal to the right hand side,χψ==χϕ χα+χβ; and this should be zero *)
charfnequality = PolynomialReduce[charfnequality, u U + v V - 1, {u, U, v, V}] // Last;
(*The canonical part,
after removal of the terms not contributing to the characteristic function.*)
expr = CoefficientList[charfnequality, {u, U, v, V}];
(*the polynomial should be zero, so the coefficients should as well*)
```

To the semidefinite optimization itself. The success probability is the trace of the auxiliary density operator α, and its structure captures the information lost during the covariant transformation. On the other hand, the operator β is the state after unsuccessful transformation: this is the part of the state which has rotational degrees of freedom incompatible with the target state.
The matrices in the result are split into diagonal components corresponding to sectors of definite angular momenta.

```
In[∘]:= αvars = SpinDensityOperatorVariables[alpha, jlim];
βvars = SpinDensityOperatorVariables[beta, jlim];
vars = Join[αvars, βvars];


expr = Join[Re@expr, Im@expr] // ComplexExpand // Flatten // Union;
(*final list of linear constraints. Each element of this list should be zero.*)
zeros = ConstantArray[0, Dimensions@expr];

{sol, dgap} = SemidefiniteOptimization[-Tr[α],
   VectorGreaterEqual[{α, 0}, {"SemidefiniteCone", dim}] &&
    VectorGreaterEqual[{β, 0}, {"SemidefiniteCone", dim}] &&
    expr == zeros &&
    Tr[β] ≤ 1, vars, {"PrimalMinimizerRules", "DualityGap"}, Tolerance → 10^-8];
Print["Success probability: ", Tr[α /. sol]];
Print["Auxiliary density operator α=",
 MatrixForm[#]_((Length[#]-1)/2) & /@ SliceMatrix@Chop[α /. sol, 10^-8]]
Print["Unsuccessful transformation result β=",
 MatrixForm[#]_((Length[#]-1)/2) & /@ SliceMatrix@Chop[β /. sol, 10^-8]]
```

Success probability: 0.333333

Auxiliary density operator $\alpha = \left\{ (0)_0, \begin{pmatrix} 0 & 0 \\ 0 & 0.333333 \end{pmatrix}_{\frac{1}{2}}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}_1, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}_{\frac{3}{2}} \right\}$

Unsuccessful transformation result $\beta = \left\{ (0)_0, \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}_{\frac{1}{2}}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}_1, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.666667 \end{pmatrix}_{\frac{3}{2}} \right\}$