6.S966: Exam 3, Spring 2025

Solutions

- This is a closed book exam. One page (8 1/2 in. by 11 in.) of notes, front and back, are permitted. Calculators are not permitted.
- The total exam time is 1 hour and 50 minutes.
- The problems are not necessarily in any order of difficulty.
- Record all your answers in the places provided. If you run out of room for an answer, continue on a blank page and mark it clearly.
- If a question seems vague or under-specified to you, make an assumption, write it down, and solve the problem given your assumption.
- If you absolutely *have* to ask a question, come to the front.
- **Write your name on every piece of paper.**

Name: _____ MIT Email: _____

Question	Points	Score
1	30	
2	15	
3	55	
Total:	100	

Name: _____

Is it equivariant?

1. (30 points) Determine whether the following proposed operations are **equivariant** with respect to the group $SO(3)$ or not.

   (a) Apply a ReLU (`max(x, 0)`) to spherical harmonic coefficients transforming as $\bigoplus_{\ell=0}^{\ell_{\max}} \ell$, representing a scalar function on S^2. Is this operation equivariant under $SO(3)$?

   ◯ Equivariant

   √ **Not equivariant**

   Solution: Explain your answer: Not equivariant. The coefficients of spherical harmonics transform as irreps of $SO(3)$ which can change sign underrotation. Thus, a ReLU activation function will act differently on these coefficients depending on arbitrary orientation of the data.

   (b) Apply a ReLU (`max(x, 0)`) pointwise to a scalar signal on S^2. Is this operation equivariant under $SO(3)$?

   √ **Equivariant**

   ◯ Not equivariant

   Solution: Explain your answer: Equivariant. A scalar function on the sphere transforms as a permutation representation, which means any transformation of the signal will only change the location of a value but not the value itself. The action of the ReLU and the permutation commute, so we do not break equivariance.

   (c) Multiply two scalar functions on S^2 pointwise. Assume both rotate together under $SO(3)$. Is the result equivariant?

   √ **Equivariant**

   ◯ Not equivariant

   Solution: Explain your answer: Equivariant. Point-wise multiplication of scalar signals is equivariant because the value resulting from the multiplication does not change with rotation. Again, the representation of rotation on the scalar signal is a permutation representation.

Name: _____

(d) Take the tensor product of spherical harmonic coefficients with themselves, where the coefficients transform as $\bigoplus_{\ell=0}^{\ell_{\max}} \ell$. Is this operation equivariant under $SO(3)$?

Solution:

√ **Equivariant**

◯ Not equivariant

Explain your answer: Equivariant. A tensor product of two direct sums of irreps preserve the group structure because $D^{\Gamma_i}(g) \otimes D^{\Gamma_j}(g) = D^{\Gamma_i \otimes \Gamma_j}(g)$.

(e) Multiply matching coefficients from a direct sum of spherical harmonics (e.g., $c_{\ell=1,m=0} \times c_{\ell=1,m=0}$). Is the result equivariant?

Solution:

◯ Equivariant

√ **Not equivariant**

Explain your answer: Not equivariant. The scalar multiplication of equivariant irreps is not equivariant.

(f) Given two copies of the same 1D irrep A of a group G, scale one by a and the other by b. Does this preserve equivariance?

Solution:

√ **Equivariant**

◯ Not equivariant

Explain your answer: Equivariant. We can always multiply an irrep by a constant matrix, which for a 1D irrep is a single scalar.

(g) Given a 2D irrep E of a group G, multiply the first component by a and the second by b. Is this operation equivariant?

Solution:

◯ Equivariant

√ **Not equivariant**

Explain your answer: Not equivariant. In order for group action on E to commute with the operation, the scalars applied to E must transform as a constant matrix.

(h) Given two copies of a 2D irrep E of a group G, scale each copy by a scalar multiple of the identity (`a * np.eye(2)` and `b * np.eye(2)`), then sum them. Does this preserve equivariance?

Solution:

√ **Equivariant**

◯ Not equivariant

Explain your answer: Equivariant. By Schur's Lemma, we can mix irreps of the same type, as long as we are doing so with constant matrices applied to each irrep.

Name: _____

Vibrational Modes of Simple Lattices

2. (15 points) In class, we analyzed vibrational modes in point-symmetric objects. We now extend this to *periodic systems* with translational symmetry, analyzing vibrational modes in simple lattices.

   (a) Consider a 1D periodic lattice with identical atoms spaced by $\mathbf{a} = a\hat{x}$. The translation symmetries of this lattice consists of shifts by integer multiples of $\mathbf{a}$. The translation irreps are 1D (tranlations are Abelian) and indexed by a continuous label $k \in [-\pi/a, \pi/a)$. These irreps are defined by

   $$\rho_k(T_n) = e^{ikna}.$$

   where n is an integer (positive, negative or zero). Briefly explain why the irreps are labeled by values in the range $k \in [-\pi/a, \pi/a)$ (called the **first Brillouin zone**). *Hint: Let $k' = k + 2\pi/a$, where $k \in [-\pi/a, \pi/a)$. Expand $e^{ik'na}$.*

   Solution:

   $$e^{ik'na} = e^{i(k+\frac{2\pi}{a})na} = e^{ikna}e^{i2\pi n} = e^{ikna}1$$

   (b) The 3D simple cubic lattice has identical atoms at positions defined by integer multiples of the lattice lattice vectors $(\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3) = (a\hat{x}, a\hat{y}, a\hat{z})$. Its vibrational modes are labeled by wavevector $\vec{k} \in [-\pi/a, \pi/a)^3$, i.e. inside the first *Brillouin zone* of the reciprocal lattice with lattice vectors $(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3) = (\frac{2\pi}{a}\hat{x}, \frac{2\pi}{a}\hat{y}, \frac{2\pi}{a}\hat{z})$. The modes transform under the *little group* at $\vec{k} = K$, the subgroup of O_h satisfying

   $$K^{\text{vec}}(g)\vec{k} = \vec{k} + n_1\mathbf{b}_1 + n_2\mathbf{b}_2 + n_3\mathbf{b}_3.$$

   That is, symmetries that leave $\vec{k}$ invariant modulo a reciprocal lattice vector.
   At fixed $\vec{k} = K$, vibrational modes transform under

   $$K^{\text{perm}}(g) \otimes K^{\text{vec}}(g),$$

   where K^{perm} acts on atom positions and K^{vec} on 3D displacements. With one atom per unit cell in the 3D cubic lattice, there are 3 degrees of freedom per $\vec{k}$. Rigid motions are not subtracted, as they do not correspond to irreps at nonzero $\vec{k}$.

   At $\vec{k}_\Gamma = (0,0,0)$, the little group is all of O_h. The atom is fixed by all symmetries, so $\Gamma^{\text{perm}}(g) = A_{1g}$, and $\Gamma^{\text{vec}}(g) = T_{1u}$. What is the irrep decomposition of the vibrational modes at $\vec{k}_\Gamma$? Explain your reasoning.

   Solution: They transform as T_{1u}. A scalar times any non-scalar irrep equals the non-scalar irrep.

(c) Vibrational modes belonging to different dimensions of the same copy of a given irrep must have the same "energy". How many distinct energies do the vibrational modes of the simple cubic lattice have at $\vec{k}_\Gamma = (0,0,0)$? Explain your reasoning.

> **Solution:** Since the vibrational modes transform as a single copy of T_{1u} at $\vec{k} = (0,0,0)$, there is only one distinct "energy".

(d) You will analyze how the vibrational modes of the simple cubic lattice transform under the *little group* at the following $\vec{k}$-points:

- $\vec{k}_Y = \frac{\pi}{a}(0,1,0)$ — invariant under the subgroup D_{4h}
- $\vec{k}_\Sigma = \frac{\pi}{a}(u,u,0)$ with $0 < u < 1$ — invariant under C_{2v}

Using the character tables for D_{4h}, and C_{2v} provided in the reference section, determine how the 3 vibrational modes of the simple cubic lattice decompose into irreps at each $\vec{k}$. Based on this, how many distinct vibrational "energies" are possible at each point?

i. How do the 3 simple cubic vibrational modes transform at $\vec{k}_Y$ (invariant under D_{4h})? How many distinct energies can the vibrational modes of the simple cubic lattice have at $\vec{k}_Y$? Explain your reasoning.

> **Solution:** Y^{vec} transforms as $A_{2u} + E_u$ under D_{4h}, thus there are two distinct energies at $\vec{k}_Y$.

ii. How do the 3 simple cubic vibrational modes transform at $\vec{k}_\Sigma$ (invariant under C_{2v})? How many distinct energies can the vibrational modes of the simple cubic lattice have at $\vec{k}_\Sigma$? Explain your reasoning.

> **Solution:** Σ^{vec} transforms as $A_1 + B_1 + B_2$ under C_{2v}, thus there are three distinct energies at $\vec{k}_\Sigma$.

Name: _____

The Tesseract

3. (55 points) The tesseract has had a surprisingly active pop culture career — appearing in science fiction films like *Interstellar* (as a physical representation of time in higher dimensions) and superhero stories like *The Avengers* (as a mysterious cube-shaped energy source). But in mathematics, the tesseract is a 4D hypercube: a highly symmetric geometric object. In this problem, you'll use your knowledge of 2D square (D_4) and 3D cube (O_h) symmetries to construct and explore the discrete symmetry group of the tesseract — no special effects needed.

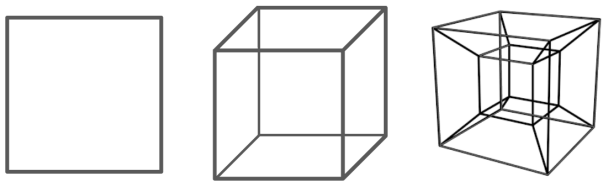

Figure 1: 2D Square, 3D Cube, 4D Tesseract

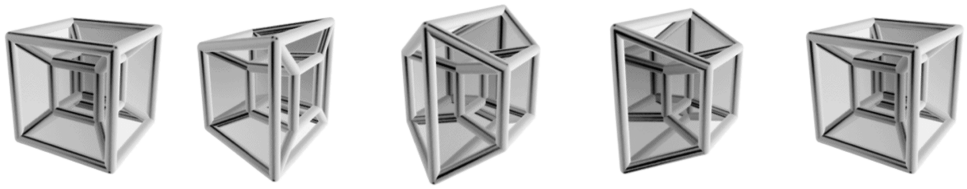

Figure 2: 3D projection of a tesseract while performing a simple rotation about a plane which bisects the figure from front-left to back-right and top to bottom. Source: Wikipedia, created by Jason Hise.

Name: _____

(a) For all code snippets below, you can assume the following libraries have been imported

```
1 import numpy as np
2 import scipy.linalg
3 from symm4ml import groups, rep, so3
```

i. Recall that we can generate the 8 element symmetry group of the square D_4 a 2D 4-fold (90°) rotation and a 2D mirror across the x axis.

```
4  so2_gen = np.array([[0, 1], [-1., 0]])
5  rot_90_2D = scipy.linalg.expm(2 * np.pi / 4 * so2_gen)
6
7  square_gen = np.stack([
8      rot_90_2D,  # 90 degree rotation
9      np.diag([1, -1])  # 2D mirror across x-axis
10 ])
11
12 square_group = groups.generate_group(square_gen)
13 print("Square group 2D vector rep: ", square_group.shape)
14 >> Square group 2D vector rep: (8, 2, 2)
```

Can the 2D mirror across the x-axis (line 9) be generated using the $SO(2)$ generator? Explain your reasoning.

Solution: No, it cannot, it has determinant -1. 2D mirror is an element of $O(2)$.

ii. Describe how the function `groups.generate_group` uses `square_gen` to generate `square_group`.

Solution: `groups.generate_group` multiplies the matrices (elements) from `square_gen` and successive matrices produced from this procedure to build up the group until closure is achieved, no additional matrices (elements) are created by multiplying known matrices (elements).

iii. The symmetry group of the cube (O_h, same as the octahedra) of 48 elements can be generated with the generators of the square and one additional rotation, a 4-fold (90° rotation) in the yz plane (whose normal vector is along the 3D x-axis) or a 4-fold (90° rotation) in the xz plane (whose normal vector is along the 3D y-axis).

```
15 rot_90_yz_3D = so3.axis_angle_to_matrix(np.array([1, 0, 0]), np.pi/2)
16 rot_90_xz_3D = so3.axis_angle_to_matrix(np.array([0, 1, 0]), np.pi/2)
17
18 square_gen_to_3D = rep.direct_sum(square_gen, np.ones([len(square_gen), 1, 1]))
19
20 # This works
21 cube_gen = np.concatenate([square_gen_to_3D, rot_90_yz_3D[np.newaxis]])
22 # So does this
23 cube_gen = np.concatenate([square_gen_to_3D, rot_90_xz_3D[np.newaxis]])
24
25 cube_group = groups.generate_group(cube_gen)
26 print("Cube group 3D vector rep: ", cube_group.shape)
27 >> Cube group 3D vector rep: (48, 3, 3)
```

How does the function `so3.axis_angle_to_matrix` use the given arguements to construct the matrices `rot_90_yz_3D` and `rot_90_xz_3D`? What representation does this function return and which generators are used to do so?

Solution: `so3.axis_angle_to_matrix` multiples the angle (`np.pi/2` in this case) by the unit vector of the axis (`np.array([1, 0, 0])` or `np.array([0, 1, 0])`) to get the Lie parameters (θ_x, θ_y, and θ_z) that are passed to the exponential parameterization of the group $SO(3)$ by the $L = 1$ generators.

iv. Describe what is happening in line 18 (`square_gen_to_3d = ...`) and why it's necessary to use to do to use `square_gen` to generate the symmetry of the cube?

Solution: We need to trivially promote the 2D representations to 3D to match dimensions of the 3D rotations used to rotate the 2D square to 3D.

v. We can follow an analogous procedure to generate the symmetry group of the tesseract (4D cube) B_4 with 384 elements. In this case, we need to add one additional generator corresponding to single plane 4-fold rotation.

```
28 def so4_generators():
29     ...
30
31 so4_gen = so4_generators() # Get the 6 generators
32 # Assume 4D coords are (x, y, z, w)
33 so4_gen_labels = ['Lxy', 'Lxz', 'Lxw', 'Lyz', 'Lyw', 'Lzw']
34
35 rot_90_4D = scipy.linalg.expm(np.pi/2 * so4_gen[so4_gen_labels.index(???)])
36
37 cube_gen_to_4d = rep.direct_sum(cube_gen, np.ones([len(cube_gen), 1, 1]))
38
39 tesseract_gen = np.concatenate([cube_gen_to_4d, rot_90_4D[np.newaxis]])
40
41 tesseract_group = groups.generate_group(tesseract_gen)
42 print("Tesseract group 4D vector rep: ", tesseract_group.shape)
43 >> Tesseract group 4D vector rep: (384, 4, 4)
```

Based on the procedure we used above to generate the symmetry of the cube from the symmetry of the square, give the labels of the three $SO(4)$ generators (three strings in `so4_gen_labels`) that we could plug into the `???`s in line 35 to generate the group of the tesseract. Explain your reasoning.

Solution: We can use any generator that mixes x, y, or z with w, i.e. `Lxw`, `Lyw`, or `Lzw`. This "pops" up the cube into the 4th dimension.

(b) Summarizing from above:

- D_4 is the symmetry group of the square in 2D (order 8),
- O_h is the symmetry group of the cube in 3D (order 48),
- B_4 is the symmetry group of the tesseract in 4D (order 384).

i. How does one construct the left cosets of a group G with respect to a subgroup H? Give the **number of cosets** and the **number of elements per coset** in terms of the size of G and H, i.e. $|G|$ and $|H|$.

Solution: Left cosets are the set of sets of elements constructed from, $gH = \{gh \text{ for an element in } G\}$ for all elements in G. There are $\frac{|G|}{|H|}$ cosets and each coset has $|H|$ elements.

ii. D_4 partitions the elements of the group O_h into 6 distinct cosets $|O_h|/|D_4| = 6$. Consider the square as one face of a cube. What do these cosets represent geometrically? Explain your reasoning.

Solution: The cube has 6 faces, and each face has internal symmetry isomorphic to D_4. But the full cube group O_h also includes symmetries that map one face to another. So the cosets O_h/D_4 correspond to which face is "selected". There are 6 such cosets, so $|O_h| = 6 \cdot 8 = 48$.

iii. The tesseract is alternatively called the 8-cell because it can be thought of as built from 8 cubes. $|B_4|/|O_h| = 8$. What do these cosets represent geometrically?

Solution: The tesseract has 8 cube-shaped cells. Each has symmetry group O_h, but the full symmetry group B_4 includes symmetries that permute the cells. So the cosets B_4/O_h correspond to which cube you're in. Hence $|B_4| = 8 \cdot 48 = 384$.

Name: _____

(c) If we try to get irreps of B_4 using our function `rep.infer_irreps` our `python` kernel quickly runs out of memory. Let's see why.

i. If $|B_4|$ is 384, what are the 3 dimensions of regular representation of $|B_4|$?

Solution: Since $|B_4|$ is 384, the regular representation has shape `(384, 384, 384)`.

ii. At some point, `rep.infer_irreps` calls `linalg.infer_change_of_basis(reg_rep, reg_rep)` where `reg_rep` is the regular representation. What is the shape (dimensions) of each matrix produced when `linalg.infer_change_of_basis` computes the **Kronecker sum** of each element of `reg_rep` (`reg_rep[i]` for `i` in `range(384)`) with itself? Explain your reasoning.

Solution: The shape of a Kronecker sum of two square matrices $M \times M$ and $N \times N$ has shape $(MN) \times (MN)$. In this case, each element is represented by an 384×384 matrix so the resulting matrix from the Kronecker sum is $384^2 \times 384^2$ so 384^4 total numbers.

iii. Suppose we are using `float32` numbers (4 bytes) to represent this matrix. Note that $100^4 \times 4$ bytes $= 400 \times 10^6$ bytes $= 400$ megabytes (MB). Give a rough estimate (within an order of magnitude) for how much memory in gigabytes (GB) $= 10^9$ bytes each of these matrices takes to store.

Solution:

$$384 = 3.84 \times 10^2 \tag{1}$$

$$(3.84 \times 10^2)^4 = (3.84)^4 \times 100^4 \tag{2}$$

$$\approx 4^4 \times 100^4 = 2^8 \times 100^4 = 256 \times 100^4 \text{ float32 numbers (4 bytes / number)} \tag{3}$$

256×400 megabytes (MB) ≈ 100 GB.

(d) To compute the irreps of B_4, Prof. Smidt instead created a new function that takes inspiration from `lie.infer_irreps_from_tensor_products` to create a new function `infer_irreps_from_tensor_products_of_vector_rep` that uses the vector matrix representation rather than generators.

i. Why can't we use `lie.infer_irreps_from_tensor_products` directly to get the irreps of B_4?

> **Solution:** B_4 is a finite group and not parameterizable by continuous parameters which is required for a Lie group.

ii. B_4 has 20 conjugacy classes. The Wonderful Orthogonality Theorem for Character tells us that the number of conjugacy classes directly tells us the number of irreps of a group. How many irreps must B_4 have? Explain how the Wonderful Orthogonality Theorem for Character gives this constraint.

> **Solution:** The Wonderful Orthogonality Theorem for Character tells us the number of conjugacy classes equals the number of irreps of the group. Since the characters of the conjugacy classes for a given irrep gives an N dimensional vector and the characters or distinct irreps are orthogonal to one another, the max number of irreps we can have equals the number of conjugacy classes. Furthermore (and optional for students to express), we know that in order to ensure that any vectorspace is uniquely decomposable into irreps, the irreps must span this space (so we also can't have less than N irreps).

Name: _____

(e) In Exam 2, you examined the irrep basis functions for the faces of the cube. The six basis functions spanned 3 distinct irreps of O_h: one 1D irrep, one 2D irrep and one 3D irrep. The irrep basis functions for the 8 cells of the tesseract look very analgous to these basis functions and also span 3 distinct irreps of B_4: one 1D irrep, one 3D irrep, and one 4D irrep. We give the plots of the cube face and tesseract cell coefficients side by side below.

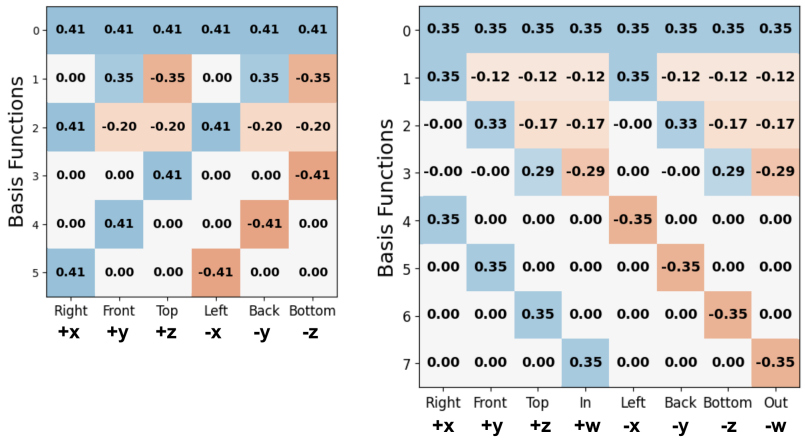

Figure 3: (Left) Cube face basis functions from Exam 2 (Right) Tesseract cell basis functions

i. If the 2D irrep basis functions 1 and 2 for the cube faces (Left) are proportional to the polynomials $y^2 - z^2$ and $2x^2 - y^2 - z^2$, respectively, what polynomials in (x, y, z, w) are the 3D irrep basis functions 1, 2, and 3 for the tesseract cells (Right) proportional to, according to the right plot above? Indicate which polynomial corresponds to each basis function.

Solution: The basis functions are proportional to $(3x^2 - y^2 - z^2 - w^2, 2y^2 - z^2 - w^2, z^2 - w^2)$, respectively.

ii. If the 3D irrep basis functions 3, 4, and 5 for the cube faces (Left) are proportional to the polynomials z, y, x, respectively, what polynomials in (x, y, z, w) are the 4D irrep basis functions 4, 5, 6, and 7 for the tesseract cells (Right) proportional to, according to the right plot above? Indicate which polynomial corresponds to each basis function.

Solution: The basis functions are proportional to (x, y, z, w), respectively.

Name: _____