6.S966: Exam 1, Spring 2025

Do not tear exam booklet apart!

- 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 hours and 20 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	15	
2	30	
3	55	
Total:	100	

Name: ______________________________

Parsing Proofs

1. (15 points) In this problem, we will go through the proof that gives us the relationship between the order (number of elements) of a (finite) group G and the dimensions of the group's irreducible representations.

$$\sum_j \ell_j^2 = |G| = h$$

To prove this, we will decompose the regular representation D^{reg} into irreps using the Wonderful Orthogonality Theorem for Character.

$$\sum_{k'=1}^{k} N_{k'} \left[\chi^{\Gamma_i}(C_{k'})\right] \chi^{\Gamma_j}(C_{k'}) = h\delta_{\Gamma_i,\Gamma_j}$$

(a) Describe how the (left) regular representation is constructed from the group's multiplication table.

(b) Explain why the regular representation has a non-zero trace only for the identity element and use this fact to determine the characters for each conjugacy class k in terms of the group order h.

Name: ____________________________

(c) The characters for any representation can be written as a linear combination of the characters of irreps, thus we can write the characters of the regular representation D^{reg} as

$$\chi^{\text{reg}}(C_k) = \sum_{\Gamma_i} a_i \chi^{(\Gamma_i)}(C_k)$$

where $\sum_{\Gamma_i}$ is the sum over irreducible representations. The coefficients a_i are given by

$$a_i = \frac{1}{h} \sum_k N_k \left[\chi^{(\Gamma_i)}(C_k)\right]^* \chi^{\text{reg}}(C_k).$$

Use this relationship and your answers above to show how many copies of each irrep are in D^{reg}.

(d) Use this result to show that $\sum_i \ell_i^2 = h$ from the expression for $\chi^{\text{reg}}(E)$.

Interpreting Outputs

2. (30 points) In the following questions, you will be shown code snippets that use functions that you have coded in the exercises and be asked to interpret the output. You may assume that all necessary imports have been made. Refer to the docstrings for these functions provided at the end of your exam booklet, before the "Work space" pages.

```
import numpy as np
from symm4ml import groups, linalg, rep
```

(a) The group C_{6v} can be generated with the following two matrices:

```
rot_mat = lambda theta: np.array([
    [np.cos(theta), np.sin(theta)],
    [-np.sin(theta), np.cos(theta)]
])
mirror_x = np.array([[1., 0], [0, -1]])

generators = [rot_mat(2 * np.pi / 6), mirror_x]
C6v_vec = groups.generate_group(np.stack(generators, axis=0))

print(C6v_vec.shape)
>> (12, 2, 2)
```

Which of the following sets of operations generate the group C_{6v}? In other words, if these lists were assigned to `generators`, the resulting operations would form C_{6v}. **Select all that apply and explain your reasoning.**

□ `[rot_mat(-2 * np.pi / 6), mirror_x]`

□ `[rot_mat(2 * np.pi / 6), rot_mat(-2 * np.pi / 6)]`

□ `[rot_mat(2 * np.pi / 6), np.array([[-1., 0], [0, 1]])]`

□ `[mirror_x, np.array([[-1., 0], [0, 1]])]`

Explain your answers:

(b) We then compute the multiplication table and irreps for C_{6v}.

```
C6v_table = groups.make_multiplication_table(C6v_vec)
np.random.seed(5)
C6v_irreps = rep.infer_irreps(C6v_table)
```

where `np.random.seed(5)` is fixing the output of `rep.infer_irreps`.

i. What representation does `rep.infer_irreps` use to infer irreps? And why does it use this representation?

ii. What part of `rep.infer_irreps` (or function that `rep.infer_irreps` calls) uses randomness and why is randomness used?

(c) We then compute the subgroups of C_{6v} and we store the elements for all subgroups of order 6.

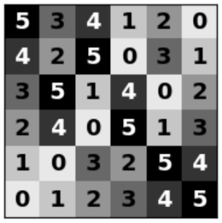
```
C6v_subs = list(groups.subgroups(C6v_table))
C6v_subs_lengths = [len(h) for h in C6v_subs]
indices_len6 = np.nonzero(C6v_subs_lengths == 6)[0]
subgroups_len6 = np.array([list(C6v_subs[i]) for i in indices_len6])

print(subgroups_len6)
>> [[ 2  3  4  5 10 11]
 [ 1  3  4  6  9 11]
 [ 0  3  4  7  8 11]]
```

i. Three elements are common between all three order 6 subgroups, `[3, 4, 11]`. What symmetry operation must one of these elements correspond to? Explain your reasoning.

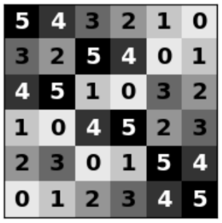
ii. The subgroups correspond to either C_{3v} or C_6. Below are the multiplication tables for the three order 6 subgroups. Two are isomorphic, and one is not. Identify which table corresponds to which group, and explain your reasoning. Hint: Recognize patterns in the table; you don't need to find an explicit isomorphism. You can assume the elements are ordered as in the `subgroups_len6` lists.

table_list[0]



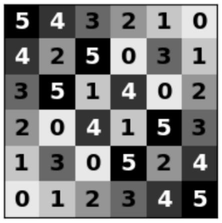
5	3	4	1	2	0
4	2	5	0	3	1
3	5	1	4	0	2
2	4	0	5	1	3
1	0	3	2	5	4
0	1	2	3	4	5

table_list[1]



5	4	3	2	1	0
3	2	5	4	0	1
4	5	1	0	3	2
1	0	4	5	2	3
2	3	0	1	5	4
0	1	2	3	4	5

table_list[2]



5	4	3	2	1	0
4	2	5	0	3	1
3	5	1	4	0	2
2	0	4	1	5	3
1	3	0	5	2	4
0	1	2	3	4	5

(d) Suppose that we have a representation of one of the 2D representations of C_{6v}, E_2, stored in the variable `C6v_E2`.

i. We calculate the following:

```
cob = linalg.infer_change_of_basis(C6v_E2, C6v_E2)
print(cob.round(2))
print(cob.shape)
>> [[[ 0.71 0.  ]
  [ 0.    0.71]]]
>> (1, 2, 2)
```

Explain the dimensions of the output `cob`. What does the zeroth dimension of the output `cob.shape[0]` mean, and what theorem can we use given this output to determine the irreducibility or reducibility of `C6v_E2`?

ii. We calculate the following where the indices corresponding to a subgroup isomorphic to C_{3v} are stored in the variable `C3v_elem`:

```
cob = linalg.infer_change_of_basis(C6v_E2[C3v_elem], C6v_E2[C3v_elem])
print(cob.round(2))
print(cob.shape)
>>[[[0.71 0.  ]
  [0.   0.71]]]
>>(1, 2, 2)
```

What does this output tell us about the irreducibility or reducibility of the C_{6v} irrep E_2 under C_{3v}? Explain your reasoning.

iii. We calculate the following where the indices corresponding to a subgroup isomorphic to C_6 are stored in the variable `C6_elem`:

```
cob = linalg.infer_change_of_basis(C6v_E2[C6_elem], C6v_E2[C6_elem])
print(cob.round(2))
print(cob.shape)
>>[[[ 0.71  0.  ]
  [-0.    0.71]]
 [[-0.    0.71]
  [-0.71 -0.  ]]]
>> (2, 2, 2)
```

What does this output tell us about the irreducibility or reducibility of the C_{6v} irrep E_2 under C_6? Explain your reasoning.

Name: ____________________

Vibrational Modes of a Hexagon

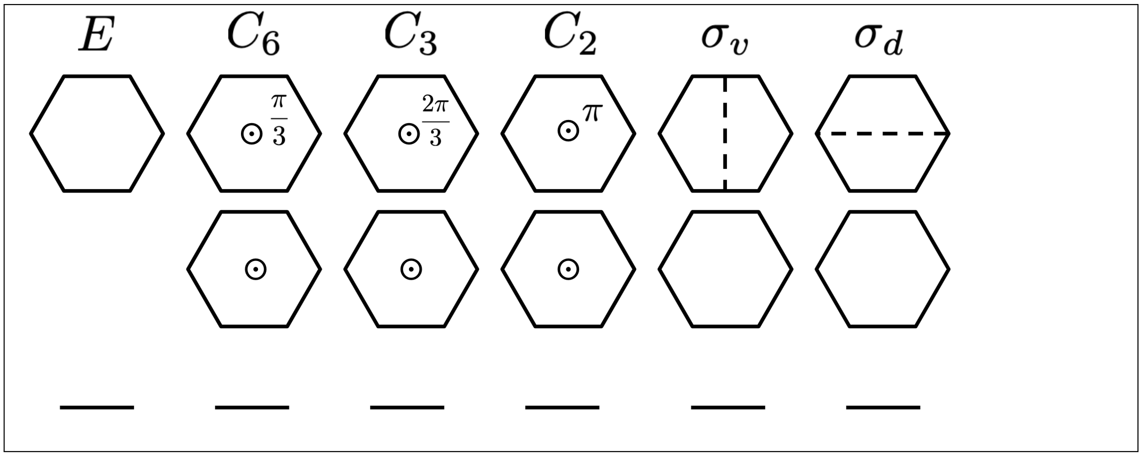
3. (55 points) In this problem, you will explore the vibrational modes and representations of a hexagon's vertices. The symmetry of a hexagon (ignoring in-plane mirror symmetry) is given by the point group C_{6v}. Below is its character table, with the usual class size numbers omitted.

C_{6v}	_E	_C_6	_C_3	_C_2	_σ_v	_σ_d
A_1	1	1	1	1	1	1
A_2	1	1	1	1	-1	-1
B_1	1	-1	1	-1	1	-1
B_2	1	-1	1	-1	-1	1
E_1	2	1	-1	-2	0	0
E_2	2	-1	-1	2	0	0

(a) Complete the diagram below to determine the number of elements in each conjugacy class. The first row shows one example element from each class acting on the hexagon's spatial representation.

i. In the second row, list or illustrate all other rotations or mirrors in the same conjugacy class. Clearly indicate the rotation angles for rotation elements and use dashed lines for mirror elements.

ii. In the third row, record the total number of elements in each conjugacy class. Hint: They should sum to 12.



(b) The 3D vector representation for the elements in the ***first*** row of part (a) is the following:

$$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad C_6 = \begin{pmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad C_3 = \begin{pmatrix} -\frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ -\frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$C_2 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \sigma_v = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \sigma_d = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

i. What are the characters of the 3D vector representation?

	E	C_6	C_3	C_2	σ_v	σ_d
Γ^{vec}	_____	_____	_____	_____	_____	_____

ii. Using the given character table and the Wonderful Orthogonality Theorem for Characters, decompose the 3D vector representation of C_{6v} into irreps of C_{6v}. Be sure to account for the number of elements in each conjugacy class in your calculations. Hint: The total dimension of the irreducible components must sum to 3.

(c) The 3D pseudovector representation transforms similar to the vector representation, except it does not change sign under inversion. This means for any inversion, rotoinversions (improper rotations) or mirrors, we "undo" the inversion contained in the matrix representation for the 3D vector (i.e. we multiply the matrix by `-1 * np.eye(d)`).

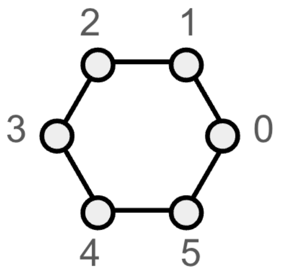
i. What are the characters of the 3D pseudovector under C_{6v}?

	E	C_6	C_3	C_2	σ_v	σ_d
$\Gamma^{\text{pseudovec}}$	_____	_____	_____	_____	_____	_____

ii. How does $\Gamma^{\text{pseudovec}}$ decompose into irreps of C_{6v}?

Name: ______________________________

(d) To determine the vibrational modes of a hexagon's vertices, we first construct the permutation representation of the group acting (from the left) on these vertices. Using the vertex ordering below, build the 6×6 permutation matrix for each element in the ***first row*** of the C_{6v} conjugacy class diagram from part (a). Only fill-in non-zero entries; leave other entries blank (they are assumed to be zero).



E

C_6

C_3

C_2

σ_v

σ_d

(e) The characters of the permutation representation of the vertices that you computed above should be the following:

	E	C_6	C_3	C_2	σ_v	σ_d
Γ^{vertices}	6	0	0	0	0	2

which decomposes into $A_1 + B_2 + E_1 + E_2$. Now, we are ready to compute the irreps of our vibrational modes.

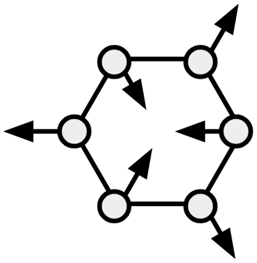
i. Use the following direct product table, to compute the irreps of $\Gamma^{\text{vertices}} \otimes \Gamma^{\text{vec}}$.

	A_1	A_2	B_1	B_2	E_1	E_2
A_1	A_1	A_2	B_1	B_2	E_1	E_2
A_2	A_2	A_1	B_2	B_1	E_1	E_2
B_1	B_1	B_2	A_1	A_2	E_2	E_1
B_2	B_2	B_1	A_2	A_1	E_2	E_1
E_1	E_1	E_1	E_2	E_2	$A_1 \oplus A_2 \oplus E_2$	$B_1 \oplus B_2 \oplus E_1$
E_2	E_2	E_2	E_1	E_1	$B_1 \oplus B_2 \oplus E_1$	$A_1 \oplus A_2 \oplus E_2$

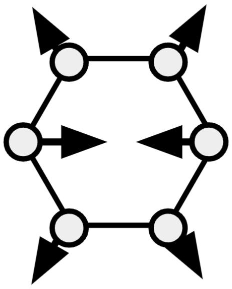
ii. Using your answers from above, deduce which irreps are contained in $\Gamma^{\text{vertices}} \otimes \Gamma^{\text{vec}} - \Gamma^{\text{translation}} - \Gamma^{\text{rotation}}$, where $\Gamma^{\text{translation}} = \Gamma^{\text{vec}}$ and $\Gamma^{\text{rotation}} = \Gamma^{\text{pseudovector}}$. You can check your answer by ensuring the total number of dimensions the irreps span is $3N - 6 = 18 - 6 = 12$.

(f) Below, we plot examples of vibrational modes that transform as specific irreps of C_{6v}, but the irrep label is missing. Use the symmetry of the distortion and the character table, to match the modes with their irrep. **Explain your reasoning.** Hint: Under which elements (represented by the conjugacy classes C_2, C_3, C_6, σ_v, and σ_d) is the distortion mode (pattern of displacements) invariant or not invariant? How does this connect to the character of the irreps the mode transforms as?

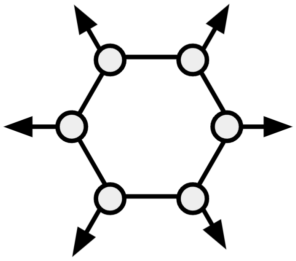
i.



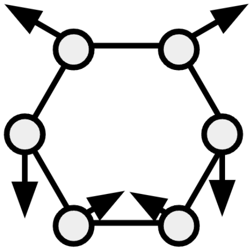
ii.



iii.



iv.



symm4ml Docstring listing

Modules listed in order: `groups, linalg, rep`

```
groups
----------------------
groups.generate_group:
Generate new group elements from matrices (group representations)
    Input:
        matrices: np.array of shape [n, d, d] of known elements
        decimals: int number of decimals to round to when comparing matrices
    Output:
        group: np.array of shape [m, d, d], where m is the size of the resultant
            group


groups.make_multiplication_table:
Makes multiplication table for group.
    Input:
        matrices: np.array of shape [n, d, d], n matrices of dimension d that form
             a group under matrix multiplication.
        tol: float numberical tolerance
    Output:
        Group multiplication table.
        np.array of shape [n, n] where entries correspond to indices of first dim
            of matrices.


groups.subgroups:
Find all subgroups of group.
    Input:
        table: np.array of shape [n, n] where the entry at [i, j] is the index of
            the product of the ith and jth elements in the group.
    Output:
        Yields tuples of elements that form subgroup.


linalg
----------------------
linalg.infer_change_of_basis:
Compute the change of basis matrix from X1 to X2.
    tip: Use the function nullspace
    Input:
        X1: an (n, d1, d1) array of n (d1, d1) matrices
        X2: an (n, d2, d2) array of n (d2, d2) matrices
    Output:
        Sols: An (m, d1, d2) array of m solutions.
        Each solution is a (d1, d2) matrix that satisfies X1 @ S = S @ X2.

linalg.nullspace:
Return the nullspace of the matrix.
    Input:
        matrix: an (n, d) matrix of n complex vectors of dimension d
        tol: a tolerance for the zero eigenvalue
    Output:
        Q: an (m, d) matrix containing orthogonal vectors spanning the nullspace (
            obtained by Gram-Schmidt)
        P: a (d, d) projector onto the span of the nullspace


rep
----------------------
rep.regular_representation:
Returns regular representation for group represented by a multiplication table.
    Input:
        table: np.array [n, n] where table[i, j] = k means i * j = k.
    Output:
        Regular representation. array [n, n, n] where reg_rep[i, :, :] = D(i) and
            D(i)e_j = e_{ij}.
                                Equivalently, D(g) |h> = |gh>


rep.decompose_rep_into_irreps:
Decomposes representation into irreducible representations.
    Input:
        rep: np.array [n, d, d] representation of group. rep[g] is a matrix that
            represents g-th element of group.
    Output:
        Irreducible representations. List of np.array [n, d_i, d_i] where d_i is a
             dimension of i-th irrep.


rep.infer_irreps:
Infers irreducible representations of group represented by multiplication table.
    Input:
        table: np.array [n, n] where table[i, j] = k means i * j = k.
    Output:
        Irreducible representations. List of np.array [n, d, d] where d is a
            dimension of irrep.
```

Work space

Work space