Lecture Notes for Modern Theoretical Condensed Matter

Alec Lau

Notes taken from a special topics course taught by Prof. Vedika Khemani in Spring 2020.

Contents

1 Thermalization & the Eigenstate Thermalization Hypothesis

Here we are thinking about strongly interacting highly excited systems. Do these systems reach thermal equilibrium under their own internal unitary dynamics? The answer is:

1. Yes, in thermalizing systems
2. No, in Many Body Localized systems.

For both cases, how and why do the dynamics either establish equilibrium or not? Also, how does a unitary reversible operation approach an apparently irreversible thermalization? The types of systems we're going to consider are:

1. Time-independent Hamiltonians $H(t) = H$. Here energy is conserved, and we have eigenstates of $H : H|\alpha\rangle = E_\alpha|\alpha\rangle$
2. Floquet systems: $H(t+T) = H(t)$. We do have eigenstates, but usually no conservations.
3. Random time-dependent Hamiltonians $H(t)$, with no symmetries. In general there are no eigenstates and no conservations.

We can think about thermalization in all of these systems, with those without eigenstates as the appropriate eventual equilibrium defined for the system.

In solving a quantum system, we have the Schrödinger Equation

$$i\hbar\frac{d}{dt}\left|\psi\right\rangle = H(t)\left|\psi(t)\right\rangle \tag{1}$$

$$U(t) = \tau e^{-i\int_0^t dt' H(t')} \tag{2}$$

$$\left|\psi(t)\right\rangle = U(t)\left|\psi(0)\right\rangle \tag{3}$$

$$U(t) = e^{-iHt} \tag{4}$$

when $H(t) = H$. In $\left|\psi(0)\right\rangle = \sum_\alpha c_\alpha \left|\alpha\right\rangle$, we have

$$\left|\psi(t)\right\rangle = \sum_\alpha c_\alpha e^{-iE_\alpha t}\left|\alpha\right\rangle \tag{5}$$

The complexity is hidden in E_α and $\left|\alpha\right\rangle$.

In traditional statistical mechanics, we have a system in a bath with energy S in E, respectively. The assumption is that $|E| >> |S|$. We assume the environment has constant temperature, chemical potential, etc. In general, if we have a state of the system $\rho(t)$, we want it to approach $\rho^S_{eq}(\tau, \mu, ...)$, the maximal entropy state. A classically chaotic system ergodically explores all of phase space. Then we can replace very detailed, complicated equations with more general statistical equations describing groups of particles. What is the analogue of this in many body quantum chaos? Classically, a signature of chaos is two functions that differ exponentially. This is opposed to, say, integral systems. But the Schrödinger equation is linear, so

$$\langle\psi_1(t)|\psi_2(t)\rangle = \langle\psi_1|\psi_2\rangle \tag{6}$$

(the Schrödinger Equation says that time evolution is unitary) To determine whether this ergodicity is the same as quantum many body chaos, we need to define quantum many body chaos. We don't know the right approach to this, but we do know how to define something else.

Definition 1. ***Thermalization*** *in a closed quantum system is a system that acts successfully as a bath for its subsystems and brings them to equilibrium at late times and for large sizes.*

Consider a subsystem $A \sqcup B =$ the system. Then we have

$$||A|| << ||B|| \tag{7}$$

$$\mathcal{H}_A = span\{|a\rangle_A\} \tag{8}$$

$$\mathcal{H}_B = span\{|b\rangle_B\} \tag{9}$$

$$\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B \tag{10}$$

where the total Hilbert space is spanned by

$$\{|a\rangle_A \otimes |b\rangle_B\} \tag{11}$$

with Hamiltonian

$$H = H_A + H_B + H_{AB} \tag{12}$$

The basic assumption is that there is *some* notion of locality. We have local degrees of freedom, such as spins on some site, particle, etc.

Definition 2. *A **subsystem** is some low-order construction in terms of local degrees of freedom. Observable operators on this subsystem are a vanishing subset of all operators one can define on this Hilbert space.*

We have pure or mixed states with density matrix ρ. $Tr(\rho) = 1$. We have

$$\rho = \sum_n p_n |n\rangle \langle n| \tag{13}$$

with $0 \leq p_n \geq 1$, and $Tr(\rho^2) \leq 1$, where you can tell that if $Tr(\rho^2) < 1$, then the state is mixed. The time evolution of this state is given by the Schrödinger equation

$$i\hbar \frac{d}{dt}\rho(t) = [H(t), \rho(t)] \tag{14}$$

Suppose we have the state AB with density matrix ρ_{AB} with density matrix of the subsystem A as $\rho_A = Tr_B(\rho_{AB})$. We define the system reaching equilibrium as

$$\lim_{t\to\infty} \rho_A = \rho_{eq} \tag{15}$$

where ρ_{eq} cares about all conservation laws in our system. If you have a many-body hamiltonian $H = \{|\alpha\rangle\}$, we have

$$\rho_\alpha = |\alpha\rangle \langle\alpha| \tag{16}$$

$$[\rho_\alpha, H] = 0 \tag{17}$$

There are classes of **integrable systems**, which have infinitely many local conserved quantities. These conserved quantities take the form of

$$N_q = \sum_i c_{q,i} O_{q,i}, \tag{18}$$

$$q = \{0, 1, 2, ...\}, f(L) \tag{19}$$

For the Generalized Gibbs Ensemble, we have

$$\rho_{GGE} = e^{-\beta(H - \mu_1 N_1 - \mu_2 N_2)} \tag{20}$$

Symmetries that act on the whole system satisfy

$$[N, H_{tot}] = 0 \tag{21}$$

Symmetries that look like a superposition of local operators

$$\sum_{i=1}^{V} c_i O_i \tag{22}$$

with

$$H = \sum_i H_i,\ S^z_{tot} = \sum_i \sigma^z_i,\ N = \sum_i n_i \tag{23}$$

Thus, for a thermalized system, we need to specify conserved quantities, and we need to have a sensible expectation value:

$$\langle E\rangle \propto V, \tag{24}$$

$$\langle \Delta E\rangle \propto V^\alpha, \alpha < 1 \tag{25}$$

$$\lim_{V\to} \frac{\langle \Delta E\rangle}{\langle E\rangle} \to 0 \tag{26}$$

then we have a sensible energy density.

If H is local, i.e. $H = \sum_i H_i$, and some state $|\psi_0\rangle = \otimes_i |\psi_i\rangle$, all product states have subextensive uncertainty in energy when H is local. This is proven via

$$(\Delta E)^2 = \langle\psi|H^2|\psi_0\rangle - |\langle\psi_0|H|\psi_0\rangle|^2 \tag{27}$$

If we have $|A| = C, |B| = V - C$, then we have the thermodynamic limit being $B \to \infty$, and $\frac{|A|}{|B|} \to 0$, so this is why we have the thermodynamic limit being represented as $\lim V \to \infty$.

We want $\rho_A \to \rho_A^{eq} = Tr_B(\rho_{AB}^{eq})$ as $V \to \infty, t \to \infty$ ($t \sim L^z$).

Equivalence of thermal states means that if, in the limit $V \to \infty$, all states gives the same limiting $\rho_A \to \rho_A^{eq}$ are thermal. Basically

1. If ρ is our candidtate thermal state
2. ρ can be mixed or pure
3. $\rho_A = Tr_B\rho$
4. ρ is thermal if $\rho_A \to \rho_A^{eq} \Rightarrow Tr_B(|n\rangle\langle n|) \to \rho_A^{eq}$

For floquet/$H(t)$ with no conservation laws, then

$$\rho_A^{eq} \propto 1_A \tag{28}$$

The Eigenstate Thermalization Hypothesis: the strong version states that every many-body eigenstate at $t > 0$ is a thermal state. Entropy is given by $S = k \log \Omega$, where Ω is the number of states at a given energy. Since $\frac{dS}{dE} = \frac{1}{T}$, we have energy well-defined and $E/V \leftrightarrow T$. We have

$$|\psi_0\rangle \to Tr_B |\psi_0(t)\rangle\langle\psi_0(t)| \tag{29}$$

$$= Tr_B \frac{e^{-\beta H}}{Z}, \text{ for } \frac{E}{V} \text{ in } |\psi_0\rangle \tag{30}$$

Again, the trouble with quantum statistical mechanics is that the eigenstates are invariant under the dynamics of the system. Suppose we have

$$|\psi_0\rangle = \sum_\alpha c_\alpha |\alpha\rangle \tag{31}$$

$$|\psi(t)\rangle = \sum_\alpha c_\alpha e^{-iE_\alpha t} |\alpha\rangle \tag{32}$$

$$\langle\psi(t)|O|\psi(t)\rangle = \sum_{\alpha,\beta} c^*_\beta c_\alpha e^{-i(E_\alpha - E_\beta)t} \langle\beta|O|\alpha\rangle \tag{33}$$

$$= \sum_\alpha |c_\alpha|^2 O_{\alpha,\alpha} + \sum_{\alpha\neq\beta} c^*_\beta c_\alpha e^{-i(E_\alpha - E_\beta t)} O_{\beta,\alpha} \tag{34}$$

We have $||\mathcal{H}_{AB}|| \propto e^V = 2L$ for spin 1/2 1-dimensional systems. We have $E_{2^L-1} - E_0$, the many-body bandwidth, is proportional to V. Then we have

$$\frac{E_0}{E_{n+1} - E_n} = \frac{\alpha V}{2^V} \propto e^{-V} \tag{35}$$

In Thermal equilibrium,

$$\sum_\alpha |c_\alpha|^2 O_{\alpha,\alpha} \to O_{eq} = TrO\rho_{eq}(t) \tag{36}$$

Then $O_{\alpha\alpha}$ must be thermal. This is the ETH.

Siednicki's statement says

$$\langle\alpha|O|\beta\rangle = O(E)\delta_{\alpha\beta} + e^{\frac{-S(E)}{2}} f(E,\omega) R_{\alpha,\beta} \tag{37}$$

where $O(E) = \langle O\rangle_{Th} + \mathcal{O}(V^{-1}), \langle O\rangle_{Th} = \frac{1}{Z} Tre^{-\beta H} O$ is smooth, δ is the diagonal part, $f(E,\omega)$ is a smooth function of its arguments, and R is a random number of mean 0 and standard deviation 1 ($E = \langle H_{th}\rangle = \frac{1}{Z} Tre^{-\beta H} H$).

So, when is ETH obeyed? Yes: (strong version) for all eigenstates at finite temperature. (weak version) for most eigenstates at finite temperature. This weak version occurres in many body quantum scars. For examples, the AKLT chain. No: In many body localized systems, integrable systems (generalized Gibbs ensemble)

2 Quantum Statistical Mechanics

The Transverse Field Ising Model (TFIM) is the system with Hamiltonian

$$H = -J \sum_i \sigma_i^z \sigma_{i+1}^z - h \sum_i \sigma_i^x \tag{38}$$

The ground state of H is ferromagnetically ordered when $J > h$, is a paramagnet for $h > J$, and the system undergoes a phase transition when $h = J$. For a system of L spins, let

$$\sigma_i^\alpha = 1 \otimes ... \otimes 1 \otimes \sigma^\alpha \otimes 1 \otimes ... \otimes 1, \alpha \in \{x, y, z\} \tag{39}$$

2.1 Block-diagonalization

The TFIM has a symmetry with the operator $P = \prod_i \sigma_i^z$, which flips all spins in the z basis. It is easy enough to create P and H using scipy's sparse matrix methods. Since P flips all spins, $P^2 = 1$, so P's eigenvalues are ± 1. Thus we can transform H into a block-diagonal matrix, corresponding to the +1 and -1 sectors. By finding the eigenvectors and eigenvalues of these matrices, we can greatly speed up the solving of the Hamiltonian: (the following code is in python using numpy and scipy sparse matrix methods)

```
def diagonalize_with_P_v1(H, P, L):
    '''assuming P is diagonal, i.e. using
      P = gen_op_prod(sz_list)'''
    diagP = P.diagonal()
    posP_inds = np.where(diagP==1)[0]
    posP_sector = np.ix_(posP_inds, posP_inds)
    negP_inds = np.where(diagP==-1)[0]
    negP_sector = np.ix_(negP_inds, negP_inds)
    Hfull = H.toarray()
    H_negP = Hfull[negP_sector]
    H_posP = Hfull[posP_sector]
    evals_negP = linalg.eigh(H_negP, eigvals_only=True)
    evals_posP = linalg.eigh(H_posP, eigvals_only=True)
    return [evals_negP, evals_posP]
```

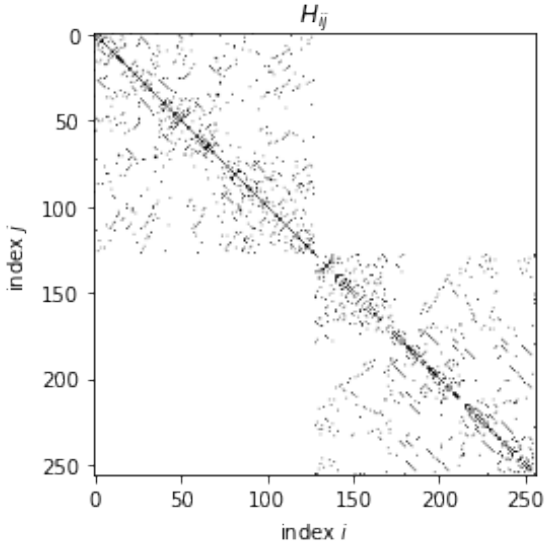
Here is an alternative method for block-diagonalizing H: (using argsort)

```
def permute_in_place(M, p):
```

```
    '''reorder rows and columns of matrix M
       according to permutation vector p'''
    N = M.shape[0]
    for i in range(N):
      M[:,i] = M[p,i]
    for i in range(N):
      M[i,:] = M[i,p]

def plot_matrix_structure(H):
    plt.imshow(np.where(np.abs(H)>1e-5,1,0),cmap='Greys')
    plt.xlabel(r'index $i$')
    plt.ylabel(r'index $j$')
    plt.title(r'$H_{ij}$')
    plt.show()

def diagonalize_with_P_v2(H, P, L, print_Hfull = False):
    '''assuming P is diagonal, i.e. using
       P = gen_op_prod(sz_list)'''

    def get_block_inds(conserved_op_eigs):
      _, inds = np.unique(conserved_op_eigs, return_index = True)
      return inds

    diagP = P.diagonal()
    perm = np.argsort(diagP)
    Hfull = H.toarray()
    permute_in_place(Hfull, perm)

    if print_Hfull:
      plot_matrix_structure(Hfull)

    _, block_inds = np.unique(diagP, return_index = True)
    block_inds = np.append(block_inds, [Hfull.shape[0]])

    evals = []
```

```
    for i in range(len(block_inds)-1):
        evals = np.append(evals, np.linalg.eigvalsh(Hfull[block_inds[i]:block_inds[i+1], b

    evals = np.sort(evals)

    return evals
```

After this block-diagonalization, we get the following matrix representation of H with $L = 8$:



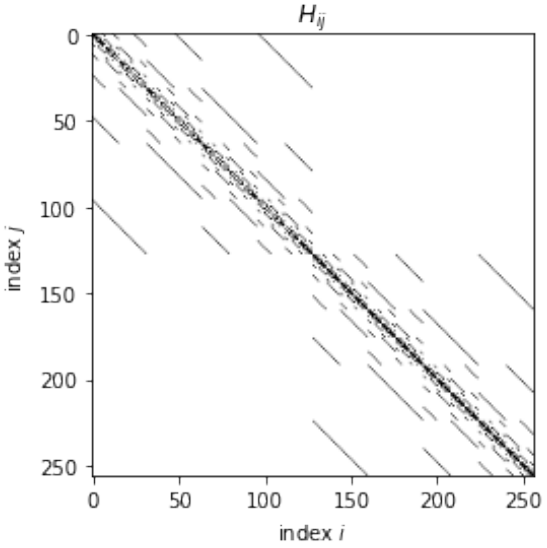
Thus we've partially split the Hamiltonian into two blocks of size $2^{L-1} \times 2^{L-1}$.

In general, to diagonalize H using parity symmetry, we need the eigenvectors of P. In the x-basis, they are of the form $|\pm_1, \pm_2, ..., \pm_L\rangle$. This is done in the following code:

```
def gen_parity_eigenstate(n, L):
    '''generates the n'th parity eigenstate
       by taking binary representation of n and
       translating 0 -> |+> and 1 -> |->'''
    b = np.binary_repr(n,width=L)
    state = [1,1] if b[0]=='1' else [1,-1]
    for j in range(1,L):
        if b[j]=='1':
            state = sparse.kron(state,[1,1],'csc')
        else:
            state = sparse.kron(state,[1,-1],'csc')
    state = state/ 2**(L/2) # properly normalize
    return state
```

```
def diagonalize_with_P_v3(H, P, L, print_Hfull = False):
'''assuming P is in x-basis, i.e. using
  P = gen_op_prod(sx_list)'''

even_projector = np.zeros((2**(L-1),2**L))
odd_projector = np.zeros((2**(L-1),2**L))
even_counter, odd_counter = 0, 0

for n in range(2**L):
  state = gen_parity_eigenstate(n, L)

  if state[0,0]==state[0,2**L-1]:
    even_projector[even_counter, :] = state.toarray()
    even_counter += 1
  else:
    odd_projector[odd_counter, :] = state.toarray()
    odd_counter += 1

if print_Hfull:
  U = np.zeros((2**L,2**L))
  U[:2**(L-1), :] = even_projector
  U[2**(L-1):, :] = odd_projector
  H_transformed = U@H@U.T
  plot_matrix_structure(H_transformed)

# Get the parity sectors of the Hamiltonian with the projectors
H_even = even_projector@H@even_projector.T
H_odd = odd_projector@H@odd_projector.T
evals_even = linalg.eigh(H_even, eigvals_only=True)
evals_odd = linalg.eigh(H_odd, eigvals_only=True)
return np.sort(np.concatenate((evals_even, evals_odd)))
```

Resulting in the following matrix:



2.2 Open boundary conditions, parity and reflection

For open systems, we have spatial reflection symmetry, with reflection operator $R : i \to L - i$. For periodic systems, we have R, but we also have spatial translation invariance, with translation operator $T : i \to i + 1$.

Remark 1. *Notice that* $[H, R] = [H, T] = 0, [R, T] \neq 0$.

For open boundary conditions, we can simultaneously diagonalize H using R and P. $R^2 = 1$, so the eigenvalues of R are ± 1. This simultaneous diagonalization should gives blocks of size $2^L/4$.

Let $|\Psi_P\rangle$ be the eigenstates of P, i.e. states of the form $|\pm_1, \pm_2, ..., \pm_L\rangle$. Then

$$|\Psi_{RP}\rangle := \frac{|\Psi_P\rangle \pm R|\Psi_P\rangle}{\sqrt{2}} \tag{40}$$

are eigenstates of both P and R. We implement this with the following code:

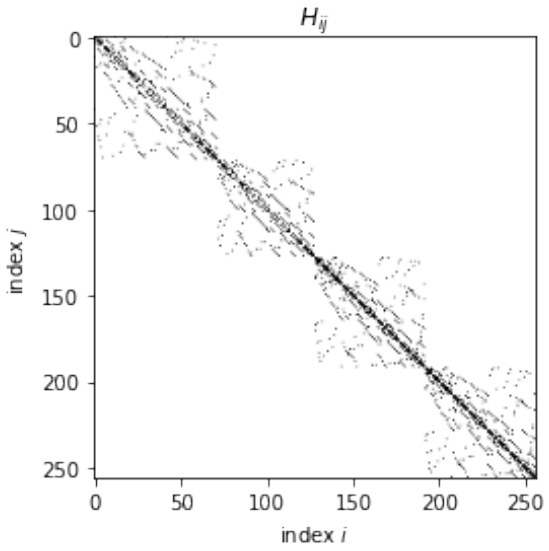
```
def bin2int(b):
    L = len(b)
    n = 0
    for i in range(L):
        if b[i] == '1':
            n += 2 **(L - i - 1)
    return n


def lex_order(b1, b2):
    '''order bit strings based on the value of their
```

```
    integers in base 10. Smallest first.'''
  n1, n2 = bin2int(b1), bin2int(b2)
  if n1 == n2:
    return 0
  elif n1 > n2:
    return 1
  else:
    return -1

def gen_reflection_op(L):
  N = 2**L
  row, col = np.arange(N), np.zeros(N)
  data = np.ones(N)
  for i in range(N):
    col[i] = bin2int(np.binary_repr(i,width=L)[::-1])
  R = sparse.csr_matrix((data, (row, col)), shape=(N, N))
  return R

def ispalindrome(b):
  return (bin2int(b) == bin2int(b[::-1]))

def get_bitstring_parity(b):
  return 1 if b.count('0')%2 == 0 else -1

def diagonalize_with_PR(H, P, R, L, print_Hfull = False):
  P_even_R_even_states = []
  P_even_R_odd_states = []
  P_odd_R_even_states = []
  P_odd_R_odd_states = []

  for i in range(2**L):
    b = np.binary_repr(i,width=L)

    if ispalindrome(b):
      state = gen_parity_eigenstate(i, L)
```

```
        peig = get_bitstring_parity(b)
        if peig > 0:
          P_even_R_even_states.append(state.toarray()[0])
        else:
          P_odd_R_even_states.append(state.toarray()[0])

      if lex_order(b, b[::-1]) <= 0:
        continue

      state = gen_parity_eigenstate(i, L)
      reflected_state = gen_parity_eigenstate(bin2int(b[::-1]), L)

      R_even_state = (state + reflected_state)/np.sqrt(2)
      R_odd_state = (state - reflected_state)/np.sqrt(2)

      peig = (R_even_state*P*R_even_state.H).toarray()[0,0]
      if peig > 0:
        P_even_R_even_states.append(R_even_state.toarray()[0])
        P_even_R_odd_states.append(R_odd_state.toarray()[0])
      else:
        P_odd_R_even_states.append(R_even_state.toarray()[0])
        P_odd_R_odd_states.append(R_odd_state.toarray()[0])

  if print_Hfull:
    U_PR = sparse.csr_matrix(np.array(P_even_R_even_states+P_even_R_odd_states+P_odd_
    H_PR = U_PR * H * U_PR.H
    plot_matrix_structure(H_PR.toarray())

  ee_projector = sparse.csr_matrix( np.array(P_even_R_even_states))
  eo_projector = sparse.csr_matrix( np.array(P_even_R_odd_states))
  oe_projector = sparse.csr_matrix( np.array(P_odd_R_even_states))
  oo_projector = sparse.csr_matrix( np.array(P_odd_R_odd_states))

  H_ee = ee_projector@H@ee_projector.H
  H_eo = eo_projector@H@eo_projector.H
```

```
    H_oe = oe_projector@H@oe_projector.H
    H_oo = oo_projector@H@oo_projector.H


    evals_ee = linalg.eigh(H_ee.toarray(), eigvals_only=True)
    evals_eo = linalg.eigh(H_eo.toarray(), eigvals_only=True)
    evals_oe = linalg.eigh(H_oe.toarray(), eigvals_only=True)
    evals_oo = linalg.eigh(H_oo.toarray(), eigvals_only=True)
    return np.sort(np.concatenate((evals_ee, evals_eo, evals_oe, evals_oo)))
```

This results in the diagonalization



2.3 Periodic boundary conditions, parity and translation

For periodic boundary conditions, we can use P and T. T splits the Hilbert space roughly into L blocks of length $2^L/L$. This is a much greater advantage than the factor of 2 from spatial reflection. P further splits each of these blocks into 2, giving a total of $2L$ blocks with size $\sim 2^L/(2L)$.

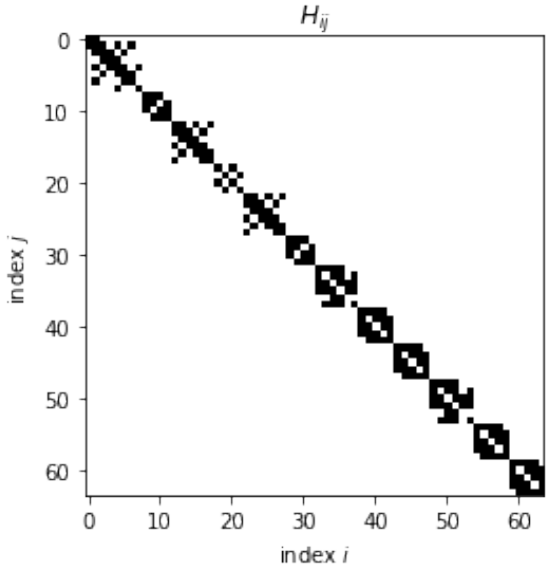
```
def translate_bitstring(b, d):
    '''translates the bitstring b by d positions '''
    b1 = b[0 : d]
    b2 = b[d :]
    return b2 + b1


def get_cycle_length(b):
    '''computes the number of translations
      allowed before bitstring b comes back
```

```
    to itself'''
  n = bin2int(b)
  L = len(b)
  for i in range(1,L):
    if n == bin2int(translate_bitstring(b, i)):
      return i
  return L

def get_T_representatives(L):
  '''get one representative from each set
    of basis states that are equivalent
    up to action of the translation op T'''
  reps = []
  for n in range(2**L):
    flag = False
    b = np.binary_repr(n, width=L)
    for d in range(1, L):
      if translate_bitstring(b, d) in reps:
        flag = True
    if not flag:
      reps.append(b)
  return reps

def get_PT_eigenstate(rep, k ,L, cycle):
    state = gen_parity_eigenstate(bin2int(rep), L)
    for i in range(1, cycle):
      state += np.exp(2 * np.pi * 1j * k * i/L) *
      gen_parity_eigenstate(bin2int(translate_bitstring(rep, i)), L)
    state = state/np.sqrt(cycle)
    return state

def diagonalize_with_PT(H, P, L, print_Hfull = False):
  tol = 0.1/L
  P_even_states = []
  P_odd_states = []
```

```
reps = get_T_representatives(L)
cycle_lengths = {r:get_cycle_length(r) for r in reps}
n_up = {r:r.count('1') for r in reps}

countkp = 0
countkm = 0

indsplus = []
indsminus = []

for k in range(L):
  indsplus.append(countkp)
  indsminus.append(countkm)

  for r in reps:
    tmp = k*cycle_lengths[r]/L
    if np.abs(np.round(tmp) - tmp) < tol:
      state = get_PT_eigenstate(r, k, L, cycle_lengths[r])
      peig = np.real( (state*P*state.H).toarray()[0,0])
      if peig > 0:
        P_even_states.append(state.toarray()[0])
        countkp = countkp+1
      else:
        P_odd_states.append(state.toarray()[0])
        countkm = countkm+1
indsplus.append(countkp)
indsminus.append(countkm)

U_PT = sparse.csr_matrix(np.array(P_even_states+P_odd_states))
H_PT = U_PT * H * U_PT.H
Hfull = H_PT.toarray()

if print_Hfull:
  plot_matrix_structure(Hfull)
```

```
    block_inds = np.append(np.array(indsplus), np.array(indsminus[1:])+indsplus[-1])

    evals = []
    for i in range(len(block_inds)-1):
        evals = np.append(evals, np.linalg.eigvalsh(Hfull[block_inds[i]:block_inds[i+1],

    evals = np.sort(evals)
    return evals
```

This results in the diagonalization



While R and T do not commute, in the $k = 0$ and $k = \pm\pi$ sectors, reflection symmetry/anti-symmetry can still be utilized. This does not provide a large speed-up in general as compared to the gains seen by T.

3 Quantum chaos