

Making sense of sensing?

Morten Hjorth-Jensen¹

Department of Physics, University of Oslo, Norway¹

March 2025

© 1999-2025, Morten Hjorth-Jensen. Released under CC Attribution-NonCommercial 4.0 license

Motivation and Content

These notes aim at linking entanglement in quantum mechanical systems with quantum sensing using simple examples. Numerical codes are included in order to illustrate basic elements of sensing. The examples are tailored to simple one- and two-qubit systems. The material contains

1. One-qubit system with basic elements of sensing
2. Linking analytically solvable case with many-qubit entanglement and sensing
3. Simple two-particle (or two-qubit) system to demonstrate entanglement and its links with sensing (to come)

More material will be added

In particular, we will include

- ▶ Quantum computing simulations of the above systems using the Variational Quantum Eigensolver algorithm
- ▶ Discussion of Fisher entropy and other measures
- ▶ Numerical codes for general time-dependent interactions
- ▶ Initial state preparations and final results and more

Feel free to come with suggestions for additions. You can access the material at [https:](https://github.com/mhjensenseminars/MachineLearningTalk/blob/master/doc/pub/sensing/ipynb/sensing.ipynb)

[//github.com/mhjensenseminars/MachineLearningTalk/
blob/master/doc/pub/sensing/ipynb/sensing.ipynb](https://github.com/mhjensenseminars/MachineLearningTalk/blob/master/doc/pub/sensing/ipynb/sensing.ipynb)

Literature

In the discussions here we have borrowed extensively from two Review of Modern Physics articles

1. **Quantum sensing**, C.L. Degen, F. Reinhard, and P. Cappellaro, Reviews of Modern Physics **89**, 035002 (2017), see <https://journals.aps.org/rmp/abstract/10.1103/RevModPhys.89.035002>
2. **Quantum metrology with nonclassical states of atomic ensembles**, L. Pezzè, A. Smerzi, M.K. Oberthaler, R. Schmied, and P. Treutlein, Reviews of Modern Physics **90**, 035005 (2018), see <https://journals.aps.org/rmp/abstract/10.1103/RevModPhys.90.035005>
3. See also recent work by Liu et al at <https://www.nature.com/articles/s41534-021-00507-x>. We may add codes which simulate these systems as well, stay tuned.

One-qubit system

In the first part of this notebook, we will analyze our systems using plain diagonalization and simple analytical manipulations.

Thereafter we will develop codes and material for performing a quantum computing simulation of the same systems (this material is not yet ready).

Our first encounter is a simple one-qubit system, described by a simple 2×2 Hamiltonian.

We start with a simple 2×2 Hamiltonian matrix expressed in terms of Pauli **X**, **Y** and **Z** matrices. But before we proceed, a simple reminder is appropriate.

Definitions: Single qubit gates

The Pauli matrices (and gate operations following therefrom) are defined as

$$\mathbf{X} \equiv \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{Y} \equiv \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \mathbf{Z} \equiv \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Pauli-**X** gate

The Pauli-**X** gate is also known as the **NOT** gate, which flips the state of the qubit.

$$\mathbf{X}|0\rangle = |1\rangle,$$

$$\mathbf{X}|1\rangle = |0\rangle.$$

The Pauli-**Y** gate flips the bit and multiplies the phase by i .

$$\mathbf{Y}|0\rangle = i|1\rangle,$$

$$\mathbf{Y}|1\rangle = -i|0\rangle.$$

The Pauli-**Z** gate multiplies only the phase of $|1\rangle$ by -1 .

$$\mathbf{Z}|0\rangle = |0\rangle,$$

$$\mathbf{Z}|1\rangle = -|1\rangle.$$

Hadamard gate

The Hadamard gate is defined as

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

It creates a superposition of the $|0\rangle$ and $|1\rangle$ states.

$$\mathbf{H}|0\rangle = \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle), \quad (1)$$

$$\mathbf{H}|1\rangle = \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle). \quad (2)$$

Note that we will use H as symbol for the Hadamard gate while we will reserve the notation \mathcal{H} for a given Hamiltonian.

Hamiltonian matrix

We define a hermitian matrix $H \in \mathbb{R}^{2 \times 2}$

$$\mathcal{H} = \begin{bmatrix} \mathcal{H}_{11} & \mathcal{H}_{12} \\ \mathcal{H}_{21} & \mathcal{H}_{22} \end{bmatrix},$$

We let $\mathcal{H} = \mathcal{H}_0 + \mathcal{H}_I$, where

$$\mathcal{H}_0 = \begin{bmatrix} E_0 & 0 \\ 0 & E_1 \end{bmatrix},$$

is a diagonal matrix. Similarly,

$$\mathcal{H}_I = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix},$$

where V_{ij} represent various interaction matrix elements and since we have a hermitian matrix, we require that $V_{21} = V_{12}^*$.

Interaction part

We will now label the interaction matrix elements, assuming that they have an explicit time dependence. We define

$$V_{11} = V_z(t)$$

$$V_{22} = -V_z(t)$$

$$V_{12} = V_x(t) - iV_y(t).$$

In the numerical example below we let $V_y(t) = 0$, $V_z(t) = tV_z$ and $V_x(t) = tV_x$ with V_z and V_x real-valued constants to be determined. In the same numerical example we let $t \in [0, 1]$.

Non-interacting solution

We can view H_0 as the non-interacting solution

$$\mathcal{H}_0|0\rangle = E_0|0\rangle,$$

and

$$\mathcal{H}_0|1\rangle = E_1|1\rangle,$$

where we have defined the orthogonal computational one-qubit basis states $|0\rangle$ and $|1\rangle$.

Rewriting with Pauli matrices

We rewrite H (and H_0 and H_I) via Pauli matrices

$$\mathcal{H}_0 = \mathcal{E}_{\text{avg}} I - \Delta E \mathbf{Z}, \quad \mathcal{E}_{\text{avg}} = \frac{E_0 + E_1}{2}, \quad \Delta E = \frac{E_1 - E_0}{2},$$

and

$$\mathcal{H}_I = V_z(t) \mathbf{Z} + V_x(t) \mathbf{X} + V_y(t) \mathbf{Y},$$

with $V_z(t) = V_{11} = -V_{22}$, $V_x(t) = \text{Re} V_{12}$ and $V_y(t) = \Im V_{12}$.

We let our Hamiltonian depend linearly on time t

$$\mathcal{H} = \mathcal{H}_0 + t \mathcal{H}_I,$$

with $t \in [0, 1]$, where the limits $t = 0$ and $t = 1$ represent the non-interacting (or unperturbed) and fully interacting system, respectively. This means that the various potential terms are given by $V_i(t) = t V_i$, with $i = \{x, y, z\}$ and V_i real-valued constants.

Selecting parameters

The model is an eigenvalue problem with only two available states. Here we set the parameters $E_1 = 0$, $E_2 = 4$, $V_{11} = -V_{22} = 3$ and $V_{12} = V_{21} = 0.2$.

The non-interacting solutions represent our computational basis. Pertinent to our choice of parameters, is that at $\lambda \geq 2/3$, the lowest eigenstate is dominated by $|1\rangle$ while the upper is $|0\rangle$. At $\lambda = 1$ the $|0\rangle$ mixing of the lowest eigenvalue is 1% while for $\lambda \leq 2/3$ we have a $|0\rangle$ component of more than 90%. The character of the eigenvectors has therefore been interchanged when passing $z = 2/3$. The value of the parameter V_{12} represents the strength of the coupling between the two states.

Setting up the matrix

This part is best seen using the jupyter-notebook

```
from matplotlib import pyplot as plt
import numpy as np
dim = 2
Hamiltonian = np.zeros((dim,dim))
e0 = 0.0
e1 = 4.0
Xnondiag = 0.20
Xdiag = 3.0
Eigenvalue = np.zeros(dim)
# setting up the Hamiltonian
Hamiltonian[0,0] = Xdiag+e0
Hamiltonian[0,1] = Xnondiag
Hamiltonian[1,0] = Hamiltonian[0,1]
Hamiltonian[1,1] = e1-Xdiag
# diagonalize and obtain eigenvalues, not necessarily sorted
EigValues, EigVectors = np.linalg.eig(Hamiltonian)
permute = EigValues.argsort()
EigValues = EigValues[permute]
# print only the lowest eigenvalue
print(EigValues[0])
```

Now rewrite it in terms of the identity matrix and the Pauli matrix X and Z

```
# Now rewrite it in terms of the identity matrix and the Pauli matrix
X = np.array([[0,1],[1,0]])
Y = np.array([[0,1j],[-1j,0]])
```

Initialization

To initialize a given system to a known quantum state, we first start with a known ground state $|0\rangle$. Then, depending on the type of information that we want to learn about the stimulus, the measurement scheme to be used, and the physical implementation of the quantum system, we choose some unitary operator U_{Init} such that it transforms our state $|0\rangle$ to a desired initial superposition state $|\psi_{\text{Init}}\rangle = a|0\rangle + b|1\rangle$ for some $a, b \in \mathbb{C}$ such that $|a|^2 + |b|^2 = 1$.

Effects of Stimulus

After the sensing state is initialized, it is exposed to the environment and evolves according to the time-evolution operator of the sensing Hamiltonian \hat{U}_H as

$$|\psi(t)\rangle = \hat{U}_H(0, t) |\psi_{\text{Init}}(0)\rangle$$

In general, $\hat{U}_H(0, t) = e^{\frac{i}{\hbar} \int_0^t \hat{H} d\tau}$ could be a complicated, non-analytical function for a time-dependent $V(t)$ (making \hat{H} time-dependent as well).

Slowly changing potential

However, in the case where $V(t)$ is constant or changes much more slowly than our sensing integration time, we can assume

$$|\psi(t)\rangle = \hat{U}_H(0, t) |\psi_{\text{Init}}(0)\rangle = e^{\frac{it}{\hbar} \hat{H}} |\psi_{\text{Init}}(0)\rangle$$

which means the sensing state evolves as

$$|\psi(t)\rangle = \left(e^{\frac{it}{\hbar} (E_0 - \frac{1}{2} \gamma V_Z)} |\lambda_0\rangle \langle \lambda_0| + e^{\frac{it}{\hbar} (E_1 + \frac{1}{2} \gamma V_Z)} |\lambda_1\rangle \langle \lambda_1| \right) |\psi_{\text{Init}}(0)\rangle$$

using the spectral decomposition and the final representation of the sensing Hamiltonian found previously (whiteboard notes).

Readout

After the sensing state has time-evolved in the presence of $V(t)$, it can be transformed again before a measurement is taken. The first part, the transformation to some desired read-out state, is performed by an operator \hat{U}_{Readout} (which is often $\hat{U}_{\text{Init}}^{-1}$, see Degen et al., 2017) where

$$|\psi_{\text{Final}}\rangle = \hat{U}_{\text{Readout}} |\psi(t)\rangle$$

Measurement

A measurement of this final state $|\psi_{\text{Final}}\rangle = a'|0\rangle + b'|1\rangle$ is made with respect to the basis $\{|0\rangle, |1\rangle\}$ where $|0\rangle$ is measured with probability

$$|\langle 0 | \psi_{\text{Final}} \rangle|^2 = |a'|^2,$$

and $|1\rangle$ is measured with probability

$$|\langle 1 | \psi_{\text{Final}} \rangle|^2 = |b'|^2.$$

After this measurement, the sensing state has been collapsed into one of the basis states, so no more information can be gained.

Multiple measurements

However, by having multiple quantum sensing elements time-evolving together or by repeating the process many times before the external stimulus $V(t)$ can change, a transition probability $p_{|0\rangle \rightarrow |1\rangle} = |\langle 1 | \psi_{\text{Final}} \rangle|^2 = |b'|^2$ can be estimated. The 'sensing' is then accomplished by taking a series of these transition probabilities as a time-series, and then using the results to estimate the sensed stimulus $V(t)$ [Degen et al., 2017].

Example

The simplest mathematical example of quantum sensing is sensing an external stimulus's effect on the splitting of the energy levels of an isolated system. Suppose our stimulus is constant and 'parallel' with our sensor, i.e. $V_z(t) = V_0$ and $V_x = 0$, and we choose our initialization and readout preparation operators to be the famous Hadamard gate

$$\hat{U}_{Had} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

since $\hat{U}_{Had}^{-1} = \hat{U}_{Had}$.

Evolution of initial state

The initial state is

$$|\psi_{\text{Init}}\rangle = \hat{U}_{\text{Had}} |0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

This need not necessarily be the same basis into which the system was initialized, but we'll assume it is so that we only have to keep track of one basis.

State evolution

The state evolves as

$$\begin{aligned} |\psi(t)\rangle &= \left(e^{\frac{it}{\hbar}(E_0 - \frac{1}{2}\gamma V_Z)} |0\rangle\langle 0| + e^{\frac{it}{\hbar}(E_1 + \frac{1}{2}\gamma V_Z)} |1\rangle\langle 1| \right) |\psi_{\text{Init}}(0)\rangle \\ &= \begin{pmatrix} e^{\frac{it}{\hbar}(E_0 - \frac{1}{2}\gamma V_Z)} & 0 \\ 0 & e^{\frac{it}{\hbar}(E_1 + \frac{1}{2}\gamma V_Z)} \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} e^{\frac{it}{\hbar}(E_0 - \frac{1}{2}\gamma V_Z)} \\ e^{\frac{it}{\hbar}(E_1 + \frac{1}{2}\gamma V_Z)} \end{pmatrix} \\ &= \frac{1}{\sqrt{2}} e^{\frac{it}{\hbar}(E_0 - \frac{1}{2}\gamma V_Z)} \begin{pmatrix} 1 \\ e^{\frac{it}{\hbar}(E_1 - E_0 + \gamma V_Z)} \end{pmatrix} \end{aligned}$$

Preparing for readout

This is then prepared for readout as

$$|\psi_{\text{Final}}\rangle = \frac{1}{2} \exp \frac{it}{\hbar} (E_0 - \frac{1}{2} \gamma V_Z) \left[\frac{1 + \exp \frac{it}{\hbar} (E_1 - E_0 + \gamma V_Z)}{1 - \exp \frac{it}{\hbar} (E_1 - E_0 + \gamma V_Z)} \right].$$

Transition probability

The transition probability

$$\begin{aligned} p_{|0\rangle \rightarrow |1\rangle} &= |\langle 1 | \psi_{\text{Final}} \rangle|^2 = \left| 1 - e^{\frac{it}{\hbar}(E_1 - E_0 + \gamma V_Z)} \right|^2 \\ &= \frac{1}{2} \left(1 - \cos \left(t \frac{(E_1 - E_0) + \gamma V_Z}{\hbar} \right) \right) \end{aligned}$$

Ramsey interferometry

We know the difference in energy between E_1 and E_0 , either since we constructed the system or by taking measurements without the external stimulus V , and we can control the time t for which the system is allowed to evolve under the external stimulus. Then we can fix t and take many measurements to estimate $p_{|0\rangle \rightarrow |1\rangle}$, which then makes finding $t \frac{\gamma V_Z}{\hbar}$ a simple phase-estimation problem which gives us γV_Z . The physical implementation of this process is known as Ramsey Interferometry, and it can be done with arbitrary initialization and readout preparation unitary operators.

Benefits of Entanglement

Up until now, we have said that we take many measurements of $|\psi_{\text{Final}}\rangle$ to estimate $p_{|0\rangle\rightarrow|1\rangle}$, but we have been glossing over the estimation process. Assuming we can take N measurements, either by having N experimental apparatuses running in parallel or by taking N different measurements of a (relatively) constant V with a single apparatus, the uncertainty in p , denoted as σ_p (this is a positive real number; not to be confused with the Pauli matrices), scales as

$$\sigma_p \propto \frac{1}{\sqrt{N}}$$

Ramsey interferometry

If we consider Ramsey Interferometry as an example, see https://en.wikipedia.org/wiki/Ramsey_interferometry, then the uncertainty in γV_Z and so in V_Z , denoted σ_V , scales as

$$\sigma_V \propto \sigma_p \propto \frac{1}{\sqrt{N}}$$

This relationship is known as the standard quantum limit (SQL) [Giovannetti et al., 2011], but can also be explained with the Law of Large Numbers from statistics, where measuring N similarly distributed, well-behaved random variables gives the sample mean as an estimator for the population mean and the sample variance divided by the size of the sample as an uncertainty in the estimate of the population mean.

More than one qubit

The nature of quantum systems allows for more information to be extracted by exploiting entanglement between quantum systems. This is the fundamental basis for the benefits of quantum computing over classical computing, and quantum sensing has similar benefits over classical sensing. Suppose we return to the example above, but rather than initializing N sensing qubits separately, we initialize $\frac{N}{n}$ groups each with n entangled quantum systems. Then we have

$$|\psi_{\text{init}}\rangle = \frac{1}{\sqrt{2^n}} (|0\rangle^{\otimes n} + |1\rangle^{\otimes n}),$$

where $|0\rangle^{\otimes n} = |0\rangle \otimes \dots \otimes |0\rangle$, n times.

After initialization

After initialization, each of the n sensing qubits evolves to pick up a relative phase factor of $e^{\frac{it}{\hbar}(E_1 - E_0 + \gamma V_Z)}$, which combined results in

$$|\psi(t)\rangle = \mathcal{N} \left(|0\rangle^{\otimes n} + e^{n \frac{it}{\hbar}(E_1 - E_0 + \gamma V_Z)} |1\rangle^{\otimes n} \right)$$

where \mathcal{N} is just a factor to take care of normalization.

Transition probability

The transition probability

$$P_{|0\rangle \rightarrow |1\rangle} = |\langle 1 | \psi_{\text{Final}} \rangle|^2 = \frac{1}{2} \left(1 - \cos \left(t \frac{n(E_1 - E_0) + n\gamma V_Z}{\hbar} \right) \right)$$

Role of entanglement

From this, we can see that through entangling n sensing qubits, the **signal** we are trying to sense increases from $V_Z \rightarrow nV_Z$, and with $\frac{N}{n}$ total measurements,

$$\sigma_V \propto \frac{1}{n} \sigma_p \propto \frac{1}{n} \left(\frac{1}{\sqrt{\frac{N}{n}}} \right) = \frac{1}{\sqrt{Nn}}$$

which means the error decreased by a factor of \sqrt{n} . In the case where $n = N$, the uncertainty now scales as

$$\sigma_V \propto \frac{1}{N}$$

which is known as the Heisenberg limit, and is the quantum-mechanically limited, maximal amount of information one can get from taking n quantum sensing measurements [Giovannetti et al., 2011].

Concluding remarks for the simple example

Quantum sensing is an emerging field with a large number of exciting applications. In terms of physical implementation, quantum sensors are able to naturally be more sensitive and sense on smaller scales than many classical sensors, while also often being directly tracable to fundamental physical constants.

Additionally, exploiting entanglement and the quantum nature of these devices allows for fundamentally more information to be collected than is allowed by classical measurements and the SQL. Quantum sensing fundamentally improves upon the physical and theoretical limitations of classical sensing, and it will be interesting to see how and what physical realizations develop to the point of practicality in the near future.

Two-qubit system, Computational basis

Our computational basis states

$$|00\rangle = |0\rangle_A \otimes |0\rangle_B = [1 \ 0 \ 0 \ 0]^T,$$

and

$$|01\rangle = |0\rangle_A \otimes |1\rangle_B = [0 \ 1 \ 0 \ 0]^T,$$

and

$$|10\rangle = |1\rangle_A \otimes |0\rangle_B = [0 \ 0 \ 1 \ 0]^T,$$

and finally

$$|11\rangle = |1\rangle_A \otimes |1\rangle_B = [0 \ 0 \ 0 \ 1]^T.$$

Simple system to illustrate entanglement

This system can be thought of as composed of two subsystems A and B . Each subsystem has computational basis states

$$|0\rangle_{A,B} = \begin{bmatrix} 1 & 0 \end{bmatrix}^T \quad |1\rangle_{A,B} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T.$$

The subsystems could represent single particles or composite many-particle systems of a given symmetry.

Computational basis states

This leads to the many-body computational basis states

$$|00\rangle = |0\rangle_A \otimes |0\rangle_B = [1 \ 0 \ 0 \ 0]^T,$$

and

$$|01\rangle = |0\rangle_A \otimes |1\rangle_B = [0 \ 1 \ 0 \ 0]^T,$$

and

$$|10\rangle = |1\rangle_A \otimes |0\rangle_B = [0 \ 0 \ 1 \ 0]^T,$$

and finally

$$|11\rangle = |1\rangle_A \otimes |1\rangle_B = [0 \ 0 \ 0 \ 1]^T.$$

Eigenstates of non-interacting Hamiltonian

These computational basis states define the eigenstates of the non-interacting Hamiltonian

$$H_0|00\rangle = \epsilon_{00}|00\rangle,$$

$$H_0|10\rangle = \epsilon_{10}|10\rangle,$$

$$H_0|01\rangle = \epsilon_{01}|01\rangle,$$

and

$$H_0|11\rangle = \epsilon_{11}|11\rangle.$$

Interaction part

The interacting part of the Hamiltonian H_I is given by the tensor product of two σ_x and σ_z matrices, respectively, that is

$$H_I = H_x \sigma_x \otimes \sigma_x + H_z \sigma_z \otimes \sigma_z,$$

where H_x and H_z are interaction strength parameters. Our final Hamiltonian matrix is given by

$$\mathbf{H} = \begin{bmatrix} \epsilon_{00} + H_z & 0 & 0 & H_x \\ 0 & \epsilon_{10} - H_z & H_x & 0 \\ 0 & H_x & \epsilon_{01} - H_z & 0 \\ H_x & 0 & 0 & \epsilon_{11} + H_z \end{bmatrix}.$$

Density matrices

The four eigenstates of the above Hamiltonian matrix can in turn be used to define density matrices. As an example, the density matrix of the first eigenstate (lowest energy E_0) Ψ_0 is given by the outerproduct

$$\rho_0 = (\alpha_{00}|00\rangle + \alpha_{10}|10\rangle + \alpha_{01}|01\rangle + \alpha_{11}|11\rangle)(\alpha_{00}^*\langle 00| + \alpha_{10}^*\langle 10| + \alpha_{01}^*\langle 01| + \alpha_{11}^*\langle 11|)$$

where the coefficients α_{ij} are the eigenvector coefficients resulting from the solution of the above eigenvalue problem.

Subsystems

We can then in turn define the density matrix for the subsets A or B as

$$\rho_A = \text{Tr}_B(\rho_0) = \langle 0|\rho_0|0\rangle_B + \langle 1|\rho_0|1\rangle_B,$$

or

$$\rho_B = \text{Tr}_A(\rho_0) = \langle 0|\rho_0|0\rangle_A + \langle 1|\rho_0|1\rangle_A.$$

Entropies

The density matrices for these subsets can be used to compute the so-called von Neumann entropy, which is one of the possible measures of entanglement. A pure state has entropy equal zero while entangled state have an entropy larger than zero. The von-Neumann entropy is defined as

$$S(A, B) = -\text{Tr}(\rho_{A,B} \log_2(\rho_{A,B})).$$

Understanding the example

The example here shows the above von Neumann entropy based on the density matrix for the lowest many-body state. We see clearly a jump in the entropy around the point where we have a level crossing. At interaction strength $\lambda = 0$ we have many-body states purely defined by their computational basis states. As we switch on the interaction strength, we obtain an increased degree of mixing and the entropy increases till we reach the level crossing point where we see an additional and sudden increase in entropy. Similar behaviors are observed for the other states. The most important result from this example is that entanglement is driven by the Hamiltonian itself and the strength of the interaction matrix elements and the non-interacting energies.

Code, best seen in jupyter-notebook

```
%matplotlib inline
from matplotlib import pyplot as plt
import numpy as np
from scipy.linalg import logm, expm
def log2M(a): # base 2 matrix logarithm
    return logm(a)/np.log(2.0)

dim = 4
Hamiltonian = np.zeros((dim,dim))
#number of lambda values
n = 40
lmbd = np.linspace(0.0,1.0,n)
Hx = 2.0
Hz = 3.0
# Non-diagonal part as sigma_x tensor product with sigma_x
sx = np.matrix([[0,1],[1,0]])
sx2 = Hx*np.kron(sx, sx)
# Diagonal part as sigma_z tensor product with sigma_z
sz = np.matrix([[1,0],[0,-1]])
sz2 = Hz*np.kron(sz, sz)
noninteracting = [0.0, 2.5, 6.5, 7.0]
D = np.diag(noninteracting)
Eigenvalue = np.zeros((dim,n))
Entropy = np.zeros(n)

for i in range(n):
    Hamiltonian = lmbd[i]*(sx2+sz2)+D
    # diagonalize and obtain eigenvalues, not necessarily sorted
```