January 30-February 3,2023: Quantum Computing, Quantum Machine Learning and Quantum Information Theories

Morten Hjorth-Jensen[1,2]

[1]Department of Physics, University of Oslo, Norway
[2]Department of Physics and Astronomy and Facility for Rare Isotope Beams, Michigan State University, USA

Jan 29, 2023

Overview, Composite Systems and Tensor Products

Composite systems and Tensor products.

1. Tensor products of Hilbert Spaces and definition of Computational Basis, partly repetition from last week
2. Quantum operations and special matrices
3. Simple Hamiltonians and other operators
4. First exercise set

Summary from last week

Last week we:

1. defined the state vector and the associated notation
2. introduced the inner product and showed how to calculate it in an orthonormal basis
3. introduced outer products and projection operators
4. introduced tensor products and showed how to contruct state vectors for multiple qubits

We will repeat some of these topics today and discuss also

1. quantum measurements are probabilistic
2. the idea of wavefunction collapse as a result of measurement Next lecture we will:

Definition of Computational basis states, repetition from last week

Assume we have a two-level system where the two states are represented by the state vectors $|\phi_0\rangle$ and $|\phi_1\rangle$, respectively. These states could represent selected or effective degrees of freedom for either a single particle (fermion or boson) or they could represent effective many-body degrees of freedon. In actual realizations of quantum computing we search often for candidate systems where we can use some low-lying states as computational basis states. But we are not limited to quantum computing. When doing many-body physics, due to the exploding degrees of freedom, we normally search after effective ways by which we can reduce the involved dimensionalities to a number of degrees of freedom we can handle by a given many-body method.

We will now relabel the above two states as two orthogonal and normalized basis (ONB) states

$$|\phi_0\rangle = |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

and

$$|\phi_1\rangle = |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

It is straight forward to see that $\langle 1|0\rangle = 0$. With these two states we can define the define the identity operator $\boldsymbol{I}$ as the sum of the outer products of these two states, namely

$$\boldsymbol{I} = \sum_{i=0}^{i=1} |i\rangle\langle i| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

We can further define the projection operators

$$\boldsymbol{P} = |0\rangle\langle 0| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$

and

$$\boldsymbol{Q} = |1\rangle\langle 1| = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

We note that $P^2 = P$, $Q^2 = Q$ (the operators are idempotent) and that their determinants are zero, meaning in turn that we cannot use these operators for unitary/orthogonal transformations. However, they play important roles in defining effective Hilbert spaces for many-body studies. Finally, before proceeding we note also that the two matrices commute and we have $\boldsymbol{PQ} = 0$ and $[\boldsymbol{P}, \boldsymbol{Q}] = 0$.

Superposition and more. Using the properties of ONBs we can expand a new state in terms of the above states. These states could also form a basis which is an eigenbasis of a selected Hamiltonian (more of this below).

We define now a new state which is a linear expansion in terms of these computational basis states

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

where the coefficients $\alpha = \langle 0|\psi\rangle$ and $\beta = \langle 1|\psi\rangle$ reresent the overlaps between the computational basis states and the state $|\psi\rangle$. In quantum speech, we say the state is in a superposition of the states $|0\rangle$ and $|1\rangle$.

Computing the inner product of $|\psi\rangle$ we obtain

$$\langle\psi|\psi\rangle = |\alpha|^2\langle 0|0\rangle + |\beta|^2\langle 1|1\rangle = |\alpha|^2 + |\beta|^2 = 1,$$

since the new basis, which is defined in terms of a a unitary/orthogonal transformation, preserves the orthogonality and norm of the original computational basis $|0\rangle$ and $|1\rangle$. To see this, consider the unitary transformation (show derivation of preserving orthogonality).

If we now act with the projection operators $\boldsymbol{P}$ and $\boldsymbol{Q}$ on the state $|\psi\rangle$ we get

$$\boldsymbol{P}|\psi\rangle = |0\rangle\langle 0|(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle,$$

that is we **project** out the $|0\rangle$ component of the state $|\psi\rangle$ with the coefficient α while $\boldsymbol{Q}$ projects out the $|1\rangle$ component with coefficient β as seen from

$$\boldsymbol{Q}|\psi\rangle = |1\rangle\langle 1|(\alpha|0\rangle + \beta|1\rangle) = \beta|1\rangle.$$

The above results can easily be derived by multiplying the pertinent matrices with the vectors $|0\rangle$ and $|1\rangle$, respectively.

Using the above linear expansion we can now define the density matrix of the state $|\psi\rangle$ as the outer product

$$\boldsymbol{\rho} = |\psi\rangle\langle\psi| = \alpha\alpha^*|0\rangle\langle 0| + \alpha\beta^*|0\rangle\langle 1| + \beta\alpha^*|1\rangle\langle 0| + \beta\beta^*|1\rangle\langle 1| = \begin{bmatrix} \alpha\alpha^* & \alpha\beta^* \\ \beta\alpha^* & \beta\beta^* \end{bmatrix}.$$

Finally, we note that the trace of the density matrix is simply given by unity

$$\mathrm{tr}\boldsymbol{\rho} = \alpha\alpha^* + \beta\beta^* = 1.$$

Tensor products and Composite systems

The state space of a composite physical system is the tensor product of the state spaces of the component systems. There are two parts to this:

1. Determining the basis for the composite system
2. Determining the components of the state vector

Last week we introduced the $|0\rangle$ and $|1\rangle$ states as our basis states for a single qubit system. We will now extend this to a two-qubit system.

Consider now two vectors with length $n = 2$, with elements

$$|x\rangle = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix},$$

and

$$|x\rangle = \begin{bmatrix} y_0 \\ y_1 \end{bmatrix}.$$

The tensor product of these two vectors is defined as

$$|x\rangle \otimes |y\rangle = |xy\rangle = \begin{bmatrix} x_0 y_0 \\ x_0 y_1 \\ x_1 y_0 \\ x_1 y_1 \end{bmatrix},$$

which is now a vector of length 4.

If we now go back to our original one-qubit basis states, we can form the following tensor products

$$|0\rangle \otimes |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = |00\rangle,$$

$$|0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = |01\rangle,$$

$$|1\rangle \otimes |0\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = |10\rangle,$$

and finally

$$|1\rangle \otimes |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = |11\rangle.$$

We have now four different states and we could try to make a new list by relabeling the states as follows $|00\rangle = |0\rangle$, $|01\rangle = |1\rangle$, $|10\rangle = |2\rangle$, $|11\rangle = |3\rangle$.

In similar ways we can define the tensor product of three qubits (or single-particle states) as

$$|0\rangle \otimes |0\rangle \otimes |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = |000\rangle,$$

which is a new vector of length eight. We note that with a single-particle basis given the states $|0\rangle$ and $|1\rangle$ we can, with N particles construct 2^N different states. This is something we can generalize to

- discuss ways of labeling states
- how to write a code which does it

The tensor product of two 2×2 matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ is given by

$$\boldsymbol{A} \times \boldsymbol{B} = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \otimes \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix} = \begin{bmatrix} a_{00}b_{00} & a_{00}b_{01} & a_{01}b_{00} & a_{01}b_{01} \\ a_{00}b_{10} & a_{00}b_{11} & a_{01}b_{10} & a_{01}b_{11} \\ a_{10}b_{00} & a_{10}b_{01} & a_{11}b_{00} & a_{11}b_{01} \\ a_{10}b_{10} & a_{10}b_{11} & a_{11}b_{10} & a_{11}b_{11} \end{bmatrix}$$

Measurements

The probability of a measurement on a quantum system giving a certain result is determined by the weight of the relevant basis state in the state vector. After the measurement, the system is in the state corresponding to the result of the measurement. The operators and gates discussed below are examples of operations we can perform on specific states.

Possible measurement

We can consider the state

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

1. A measurement can yield only one of the above states, either $|0\rangle$ or $|1\rangle$.
2. The probability of a measurement resulting in $|0\rangle$ is $\alpha^*\alpha = |\alpha|^2$.
3. The probability of a measurement resulting in $|1\rangle$ is $\beta^*\beta = |\beta|^2$.
4. And we note that the sum of the outcomes gives $\alpha^*\alpha + \beta^*\beta = 1$ since the two states are normalized.

After the measurement, the state of the system is the state associated with the result of the measurement.

We have already encountered the projection operators P and Q. Let us now look at other types of operations we can make on qubit states.

Different operators and gates

In quantum computing, the so-called Pauli matrices, and other simple 2×2 matrices, play an important role, ranging from the setup of quantum gates to a rewrite of creation and annihilation operators and other quantum mechanical

operators. Let us start with the familiar Pauli matrices and remind ourselves of some of their basic properties.

The Pauli matrices are defined as

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

$$\sigma_y = \begin{bmatrix} 0 & -\imath \\ \imath & 0 \end{bmatrix},$$

and

$$\sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

It is easy to show that the matrices obey the properties (being involutory)

$$\sigma_x\sigma_x = \sigma_y\sigma_y = \sigma_z\sigma_z = I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

that is their products with themselves result in the identity matrix $\boldsymbol{I}$. Furthermore, the Pauli matrices are unitary matrices meaning that their inverses are equal to their hermitian conjugated matrices. The determinants of the Pauli matrices are all equal to -1, as can be easily verified.

The Pauli matrices obey also the following commutation rules

$$[\sigma_x, \sigma_y] = 2\imath\sigma_z.$$

Before we proceed with other matrices and how they can be used to operate on various quantum mechanical states, let us try apply these matrices to our one-qubit states.

Assume we operate with σ_x on our basis state $|0\rangle$. This gives

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

that is we switch from $|0\rangle$ to $|1\rangle$ (often called a spin flip operation) and similary we have

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

This matrix plays an important role in quantum computing since we can link this with the classical **NOT** operation. If we send in bit 0, the **NOT** gate outputs bit 1 and vice versa. We can use the σ_x matrix to implement the quantum mechanical equivalent of a classical **NOT** gate. If we input what we could represent as bit 0 in terms of the basis state $|0\rangle$, operating on this state results in the state $|1\rangle$, which we in turn can interpret as the classical bit 1.

If we have a linear superposition of these states we obtain

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}.$$

The σ_y matrix introduces an imaginary sign, which we will later encounter in terms of so-called phase-shift operations.

The σ_z matrix has the following effect

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

and

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix},$$

which we will also link with a specific phase-shift.

Other important matrices. We end with presenting other operators (as matrices) which play an important role in quantum computing, the so-called Hadamard matrix (or gate as we will use later)

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

The action of the operator $\boldsymbol{H}$ on a computational basis state like $|0\rangle$ gives

$$\boldsymbol{H}|0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle),$$

and

$$\boldsymbol{H}|1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle),$$

that is we create a superposition of the states $|0\rangle$ and $|1\rangle$.

Another famous operation is the phase matrix given by

$$\boldsymbol{S} = \begin{bmatrix} 1 & 0 \\ 0 & \imath \end{bmatrix}.$$

Unitarity

The matrices we introduced here are so-called unitary matrices. This is an important element in quantum mechanics since the evolution of a closed quantum system is described by operations involving unitary operations only.

We have defined a new state $|\psi_p\rangle$ as a linear expansion in terms of an orthogonal and normalized basis (our computational basis) ϕ_λ

$$|\psi_i\rangle = \sum_j u_{ij}|\phi_j\rangle. \tag{1}$$

It is normal to choose a basis defined as the eigenfunctions of parts of the full Hamiltonian. The typical situation consists of the solutions of the one-body part of the Hamiltonian, that is we have

$$\hat{h}_0|\phi_i\rangle = \epsilon_i|\phi_i\rangle.$$

This is normally referred to as a single-particle basis $|\phi_i(\mathbf{r})\rangle$, defined by the quantum numbers i and $\mathbf{r}$.

A unitary transformation is important since it keeps the orthogonality. To see this consider first a basis of vectors $\mathbf{v}_i$,

$$\mathbf{v}_i = \begin{bmatrix} v_{i1} \\ \cdots \\ \cdots \\ v_{in} \end{bmatrix}$$

We assume that the basis is orthogonal, that is

$$\mathbf{v}_j^T\mathbf{v}_i = \delta_{ij}.$$

An orthogonal or unitary transformation

$$\mathbf{w}_i = \mathbf{U}\mathbf{v}_i,$$

preserves the dot product and orthogonality since

$$\mathbf{w}_j^T\mathbf{w}_i = (\mathbf{U}\mathbf{v}_j)^T\mathbf{U}\mathbf{v}_i = \mathbf{v}_j^T\mathbf{U}^T\mathbf{U}\mathbf{v}_i = \mathbf{v}_j^T\mathbf{v}_i = \delta_{ij}.$$

This means that if the coefficients $u_{p\lambda}$ belong to a unitary or orthogonal transformation (using the Dirac bra-ket notation)

$$|\psi_i\rangle = \sum_j u_{ij}|\phi_j\rangle.$$

orthogonality is preserved.

This propertry is extremely useful when we build up a basis of many-body determinant based states.

$_N$*otealsothatalthoughabasis*$\{|\phi_i\rangle\}$ contains an infinity of states, for practical calculations we have always to make some truncations.

Example. Assume we have two one-qubit states represented by

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix},$$

and

$$|\phi\rangle = \gamma|0\rangle + \delta|1\rangle = \begin{bmatrix} \gamma \\ \delta \end{bmatrix}.$$

We assume that the state $|\phi\rangle$ is obtained through a unitary transformation of $|\psi\rangle$ through a matrix $\boldsymbol{U}$ with its hermitian conjugate $\boldsymbol{U}^\dagger$ with matrix elements $u_{ij}^\dagger = u_{ji}^*$ and $\boldsymbol{I} = \boldsymbol{U}\boldsymbol{U}^\dagger = \boldsymbol{U}^\dagger\boldsymbol{U}$.

Note that this means that the hermitian conjugate of a unitary matrix is equal to its inverse. This has important consequences for what is called reversibility. We say quantum mechanics is a theory which is reversible with a probabilistic determinism. Classical mechanics on the other is reversible in a deterministic way, that is, knowing all initial conditions we can in principle determine the future motion of an object which obey the laws of motion of classical mechanics.

We have then

$$\begin{bmatrix} \gamma \\ \delta \end{bmatrix} = \begin{bmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

Since our original basis $|\psi\rangle$ is orthogonal and normalized with $|\alpha|^2 + |\beta|^2 = 1$, the new basis is also orthogonal and normalized, as we can see below here.

Only unitary transformations are of interest. Since the inverse of a hermitian matrix is equal to its hermitian conjugate/adjoint), unitary transformations are always reversible.

Why are only unitary transformations allowed? The key lies in the way the inner product tranforms.

To see this we rewrite the new basis from the previous example in its two components as

$$|\phi\rangle_i = \sum_j u_{ij}|\psi\rangle_j,$$

or in terms of a matrix-vector notatio we have

$$|\phi\rangle = \boldsymbol{U}|\psi\rangle,$$

We have already assumed that $\langle\psi|\psi\rangle = |\alpha|^2 + |\beta|^2 = 1$.

We have that

$$\langle\phi|_i = \sum_j u^*_{ij}\langle\psi|_j,$$

or in terms of a matrix-vector notation we have

$$\langle\phi| = \langle\psi|\boldsymbol{U}^\dagger.$$

Note that the two vectors are row vectors now.

If we stay with this notation we have

$$\langle\phi|\phi\rangle = \langle\psi\boldsymbol{U}^\dagger\boldsymbol{U}|\psi\rangle = \langle\psi|\psi\rangle = 1!$$

Unitary transformations are rotations in state space which preserve the length (the square root of the inner product) of the state vector.

Representation of states and Hamiltonians

Before we proceed we need several other definitions. Throughout these lectures we will assume that the interacting part of the Hamiltonian can be approximated by a two-body interaction. This means that our Hamiltonian can be written

as the sum of a onebody part, which includes kinetic energy and an eventual external field, and a twobody interaction

$$\hat{H} = \hat{H}_0 + \hat{H}_I = \sum_{i=1}^{N} \hat{h}_0(x_i) + \sum_{i<j}^{N} \hat{v}(r_{ij}), \tag{2}$$

with

$$H_0 = \sum_{i=1}^{N} \hat{h}_0(x_i). \tag{3}$$

The onebody part $u_{\rm ext}(x_i)$ is normally approximated by a harmonic oscillator potential or the Coulomb interaction an electron feels from the nucleus. However, other potentials are fully possible, such as one derived from the self-consistent solution of the Hartree-Fock equations.

Our Hamiltonian is invariant under the permutation (interchange) of two particles. Since we deal with fermions however, the total wave function is antisymmetric. Let $\hat{P}$ be an operator which interchanges two particles. Due to the symmetries we have ascribed to our Hamiltonian, this operator commutes with the total Hamiltonian,

$$[\hat{H}, \hat{P}] = 0,$$

meaning that $\Psi_\lambda(x_1, x_2, \dots, x_A)$ is an eigenfunction of $\hat{P}$ as well, that is

$$\hat{P}_{ij}\Psi_\lambda(x_1, x_2, \dots, x_i, \dots, x_j, \dots, x_N) = \beta\Psi_\lambda(x_1, x_2, \dots, x_i, \dots, x_j, \dots, x_N),$$

where β is the eigenvalue of $\hat{P}$. We have introduced the suffix ij in order to indicate that we permute particles i and j. The variable N refers to the number of particles. The Pauli principle tells us that the total wave function for a system of fermions has to be antisymmetric, resulting in the eigenvalue $\beta = -1$.

In our case we assume that we can approximate the exact eigenfunction with a Slater determinant

$$\Phi(x_1, x_2, \dots, x_N, \alpha, \beta, \dots, \sigma) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \psi_\alpha(x_1) & \psi_\alpha(x_2) & \dots & \dots & \psi_\alpha(x_A) \\ \psi_\beta(x_1) & \psi_\beta(x_2) & \dots & \dots & \psi_\beta(x_N) \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \psi_\sigma(x_1) & \psi_\sigma(x_2) & \dots & \dots & \psi_\sigma(x_N) \end{vmatrix}, \tag{4}$$

where x_i stand for the coordinates and spin values of a particle i and $\alpha, \beta, \dots, \gamma$ are quantum numbers needed to describe remaining quantum numbers.

If we deal with Fermions (identical and indistinguishable particles) we will form an ansatz for a given state in terms of so-called Slater determinants determined by a chosen basis of single-particle functions.

For a given $n \times n$ matrix $\mathbf{A}$ we can write its determinant

$$det(\mathbf{A}) = |\mathbf{A}| = \begin{vmatrix} a_{11} & a_{12} & \dots & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & \dots & a_{nn} \end{vmatrix},$$

in a more compact form as

$$|\mathbf{A}| = \sum_{i=1}^{n!} (-1)^{p_i} \hat{P}_i a_{11} a_{22} \dots a_{nn},$$

where $\hat{P}_i$ is a permutation operator which permutes the column indices $1, 2, 3, \dots, n$ and the sum runs over all $n!$ permutations. The quantity p_i represents the number of transpositions of column indices that are needed in order to bring a given permutation back to its initial ordering, in our case given by $a_{11} a_{22} \dots a_{nn}$ here.

A simple 2×2 determinant illustrates this. We have

$$det(\mathbf{A}) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = (-1)^0 a_{11} a_{22} + (-1)^1 a_{12} a_{21},$$

where in the last term we have interchanged the column indices 1 and 2. The natural ordering we have chosen is $a_{11} a_{22}$.

We define our new single-particle basis (this is a normal approach for Hartree-Fock theory) by performing a unitary transformation on our previous basis (labelled with greek indices) as

$$\psi_p^{new} = \sum_{\lambda} C_{p\lambda} \phi_{\lambda}. \tag{5}$$

In this case we vary the coefficients $C_{p\lambda}$. If the basis has infinitely many solutions, we need to truncate the above sum. We assume that the basis ϕ_{λ} is orthogonal.

If we stay with determinants, a feature which will become useful is to expand a new determinant in terms of a previous one that is defined by a given set of single-particle state functions. As discussed above, we can define a new basis that is a linear combination of another basis (assumed to be orthogonal and normalized). This means that if the coefficients $u_{p\lambda}$ belong to a unitary or orthogonal trasformation (using the Dirac bra-ket notation)

$$|p\rangle = \sum_{\lambda} u_{p\lambda} |\lambda\rangle,$$

orthogonality is preserved, as discussed above.

This propertry is extremely useful when we build up a basis of many-body Stater determinant based states.

Consider the following determinant

$$\begin{vmatrix} \alpha_1 b_{11} + \alpha_2 s b_{12} & a_{12} \\ \alpha_1 b_{21} + \alpha_2 b_{22} & a_{22} \end{vmatrix} = \alpha_1 \begin{vmatrix} b_{11} & a_{12} \\ b_{21} & a_{22} \end{vmatrix} + \alpha_2 \begin{vmatrix} b_{12} & a_{12} \\ b_{22} & a_{22} \end{vmatrix}$$

We can generalize this to an $n \times n$ matrix and have

$$\begin{vmatrix} a_{11} & a_{12} & \dots & \sum_{k=1}^{n} c_k b_{1k} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & \sum_{k=1}^{n} c_k b_{2k} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & \sum_{k=1}^{n} c_k b_{nk} & \dots & a_{nn} \end{vmatrix} = \sum_{k=1}^{n} c_k \begin{vmatrix} a_{11} & a_{12} & \dots & b_{1k} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & b_{2k} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & b_{nk} & \dots & a_{nn} \end{vmatrix}.$$

This is a property we will use in our discussions of different many-body methods in later chapters.

We can generalize the previous results, now with all elements a_{ij} being given as functions of linear combinations of various coefficients c and elements b_{ij},

$$\begin{vmatrix} \sum_{k=1}^{n} b_{1k} c_{k1} & \sum_{k=1}^{n} b_{1k} c_{k2} & \dots & \sum_{k=1}^{n} b_{1k} c_{kj} & \dots & \sum_{k=1}^{n} b_{1k} c_{kn} \\ \sum_{k=1}^{n} b_{2k} c_{k1} & \sum_{k=1}^{n} b_{2k} c_{k2} & \dots & \sum_{k=1}^{n} b_{2k} c_{kj} & \dots & \sum_{k=1}^{n} b_{2k} c_{kn} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \sum_{k=1}^{n} b_{nk} c_{k1} & \sum_{k=1}^{n} b_{nk} c_{k2} & \dots & \sum_{k=1}^{n} b_{nk} c_{kj} & \dots & \sum_{k=1}^{n} b_{nk} c_{kn} \end{vmatrix} = det(\mathbf{C}) det(\mathbf{B}),$$

where $det(\mathbf{C})$ and $det(\mathbf{B})$ are the determinants of $n \times n$ matrices with elements c_{ij} and b_{ij} respectively. This is a property we will use in our Hartree-Fock discussions. Convince yourself about the correctness of the above expression by setting $n = 2$.

With our definition of the new basis in terms of an orthogonal basis we have

$$\psi_p(x) = \sum_{\lambda} C_{p\lambda} \phi_\lambda(x).$$

If the coefficients $C_{p\lambda}$ belong to an orthogonal or unitary matrix, the new basis is also orthogonal. Our Slater determinant in the new basis $\psi_p(x)$ is written as

$$\frac{1}{\sqrt{N!}} \begin{vmatrix} \psi_p(x_1) & \psi_p(x_2) & \dots & \dots & \psi_p(x_N) \\ \psi_q(x_1) & \psi_q(x_2) & \dots & \dots & \psi_q(x_N) \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \psi_t(x_1) & \psi_t(x_2) & \dots & \dots & \psi_t(x_N) \end{vmatrix} = \frac{1}{\sqrt{N!}} \begin{vmatrix} \sum_\lambda C_{p\lambda}\phi_\lambda(x_1) & \sum_\lambda C_{p\lambda}\phi_\lambda(x_2) & \dots & \dots & \sum_\lambda C_{p\lambda}\phi_\lambda(\\ \sum_\lambda C_{q\lambda}\phi_\lambda(x_1) & \sum_\lambda C_{q\lambda}\phi_\lambda(x_2) & \dots & \dots & \sum_\lambda C_{q\lambda}\phi_\lambda(\\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \sum_\lambda C_{t\lambda}\phi_\lambda(x_1) & \sum_\lambda C_{t\lambda}\phi_\lambda(x_2) & \dots & \dots & \sum_\lambda C_{t\lambda}\phi_\lambda(\end{vmatrix}$$

which is nothing but $det(\mathbf{C})det(\Phi)$, with $det(\Phi)$ being the determinant given by the basis functions $\phi_\lambda(x)$.

Simple Hamiltonian models

In order to study get started with coding, we will study two simple Hamiltonian systems, one which we can use for a single qubit systems and one which has as basis functions a two-qubit system. These two simple Hamiltonians exhibit also something which is called level crossing, a feature which we will use in later studies of entanglement.

We study first a simple two-level system. Thereafter we extend our model to a four-level system which can be interpreted as composed of two separate (not necesseraly identical) subsystems.

We let our hamiltonian depend linearly on a strength parameter z

$$H = H_0 + \lambda H_{\rm I},$$

with $\lambda \in [0,1]$, where the limits $\lambda = 0$ and $\lambda = 1$ represent the non-interacting (or unperturbed) and fully interacting system, respectively. The model is an eigenvalue problem with only two available states, which we label $|0\rangle$ and $|1\rangle$, respectively. Below we will let state $|0\rangle$ represent the lowest state (often referred to as model-space state) with its pertinent eigenvalue and eigenvector whereas state $|1\rangle$ represents the eigenvalue of the excluded space. The non-interacting solutions to our problem are

$$H_0|0\rangle = \epsilon_0|0\rangle, \tag{6}$$

and

$$H_0|1\rangle = \epsilon_1|1\rangle, \tag{7}$$

with $\epsilon_0 < \epsilon_1$. We label the off-diagonal matrix elements X, while $X_0 = \langle 0|H_I|0\rangle$ and $X_1 = \langle 1|H_1|1\rangle$. The exact eigenvalue problem

$$\begin{pmatrix} \epsilon_0 + \lambda X_0 & \lambda X \\ zX & \epsilon_1 + \lambda X_1 \end{pmatrix} \tag{8}$$

yields

$$\begin{aligned} E(\lambda) = \ & \tfrac{1}{2}\{\epsilon_0 + \epsilon_1 + \lambda X_0 + \lambda X_1 \pm (\epsilon_1 - \epsilon_0 + \lambda X_1 - \lambda X_0) \\ & \times\sqrt{1 + \tfrac{4\lambda^2 X^2}{(\epsilon_1 - \epsilon_0 + \lambda X_1 - \lambda X_0)^2}}\Big\}. \end{aligned} \tag{9}$$

In the results below we set the parameters $\epsilon_0 = 0$, $\epsilon_1 = 4$, $X_0 = -X_1 = 3$ and $X = 0.2$. This eigenvalue problem can easily be rewritten in terms of the standard Pauli matrices. 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 X represents the strength of the coupling between the model space and the excluded space. The following code computes and plots the eigenvalues.

```
%matplotlib inline

from  matplotlib import pyplot as plt
import numpy as np
dim = 2
#Setting up a tridiagonal matrix and finding eigenvectors and eigenvalues
Hamiltonian = np.zeros((dim,dim))
#number of lambda values
```

```
n = 100
lmbd = np.linspace(0.,1.0,n)
e0 = 0.0
e1 = 4.0
X = 0.20
Xp = 3.0
Eigenvalue = np.zeros((dim,n))
for i in range(n):
    Hamiltonian[0,0] = lmbd[i]*Xp+e0
    Hamiltonian[0,1] = lmbd[i]*X
    Hamiltonian[1,0] = Hamiltonian[0,1]
    Hamiltonian[1,1] = e1+lmbd[i]*(-Xp)
    # diagonalize and obtain eigenvalues, not necessarily sorted
    EigValues, EigVectors = np.linalg.eig(Hamiltonian)
    # sort eigenvectors and eigenvalues
    permute = EigValues.argsort()
    EigValues = EigValues[permute]
    EigVectors = EigVectors[:,permute]
    Eigenvalue[0,i] = EigValues[0]
    Eigenvalue[1,i] = EigValues[1]
plt.plot(lmbd, Eigenvalue[0,:] ,'b-',lmbd, Eigenvalue[1,:],'g-',)
plt.xlabel('$\lambda$')
plt.ylabel('Eigenvalues')
plt.show()
```

This model exhibits a simple level crossing where the composition of the final interacting states change character as we gradually switch on the interaction.

We extend the simple two-level system to a four level system. This system can be thought of as composed of two subsystems A and B. Each subsystem has computational basis states

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

The subsystems could represent single particles or composite many-particle systems of a given symmetry. This leads to the many-body computational basis states

$$|00\rangle = |0\rangle_{\mathrm{A}} \otimes |0\rangle_{\mathrm{B}} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T,$$

and

$$|10\rangle = |1\rangle_{\mathrm{A}} \otimes |0\rangle_{\mathrm{B}} = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}^T,$$

and

$$|01\rangle = |0\rangle_{\mathrm{A}} \otimes |1\rangle_{\mathrm{B}} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T,$$

and finally

$$|11\rangle = |1\rangle_{\mathrm{A}} \otimes |1\rangle_{\mathrm{B}} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T.$$

These computational basis states define also 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

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

The interacting part of the Hamiltonian $H_{\rm I}$ is given by the tensor product of two σ_x and σ_z matrices, respectively, that is

$$H_{\rm I} = H_x\sigma_x \otimes \sigma_x + H_z\sigma_z \otimes \sigma_z,$$

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

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

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

$$\rho_0 = \left(\alpha_{00}|00\rangle\langle 00| + \alpha_{10}|10\rangle\langle 10| + \alpha_{01}|01\rangle\langle 01| + \alpha_{11}|11\rangle\langle 11|\right),$$

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

```
%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))

for i in range(n):
    Hamiltonian = lmbd[i]*(sx2+sz2)+D
    # diagonalize and obtain eigenvalues, not necessarily sorted
    EigValues, EigVectors = np.linalg.eig(Hamiltonian)
    # sort eigenvectors and eigenvalues
    permute = EigValues.argsort()
    EigValues = EigValues[permute]
```

```
        EigVectors = EigVectors[:,permute]
        # Compute density matrix for selected system state, here ground state
        DensityMatrix = np.zeros((dim,dim))
        DensityMatrix = np.outer(EigVectors[:,0],EigVectors[:,0])
        # Plotting eigenvalues
        Eigenvalue[0,i] = EigValues[0]
        Eigenvalue[1,i] = EigValues[1]
        Eigenvalue[2,i] = EigValues[2]
        Eigenvalue[3,i] = EigValues[3]
    plt.plot(lmbd, Eigenvalue[0,:] ,'b-',lmbd, Eigenvalue[1,:],'g-',)
    plt.plot(lmbd, Eigenvalue[2,:] ,'r-',lmbd, Eigenvalue[3,:],'y-',)
    plt.xlabel('$\lambda$')
    plt.ylabel('Eigenvalues')
    plt.show()
```

First exercise

The exercises we present each week are meant to build the basis for the two projects we will work on during the semester. The first project deals with implementing the so-called **Phase Estimation** and **Variational Quantum Eigensolver** algorithms for finding the eigenvalues and eigenvectors of selected Hamiltonians. Feel free to use the above codes in order to get started.

1. Write a function which sets up a one-qubit basis and apply the various Pauli matrices to these basis states.

2. Apply the Hadamard and Phase gates to the same one-qubit basis states.

The next lecture

In our next lecture, we will discuss

1. how much information is stored in a qubit

2. show how to make measurements on composite states

3. introduce the key concept of entanglement