

Qiskit Document Tutorials 勉強会

Qiskit Nature

電子構造

Emi ADACHI



Agenda

1. Qiskit Natureの概要と使い方の注意点
2. シュレディンガー方程式
3. ボルン・オッペンハイマー近似
4. Hartree Fock法
5. Hartree Fock法のコードの実行と説明
6. フェルミオンを量子ビットにマッピングする方法
7. 電子と量子ビットの特徴
8. Jordan Wigner変換
9. Jordan Wigner変換のコードの実行と説明
10. 参考文献、引用文献一覧



1.1 Qiskit Nature とは？

量子アルゴリズムを使って、自然科学の問題を調べたり、シミュレーションするモジュール
物理学、化学、生物学の複雑な問題に対処すること



1.2 Qiskit Natureを使うときの注意点

Migration Guide:

コードを書くときに、最新のものを使って書くときはAquaは一切使わない

Reason :

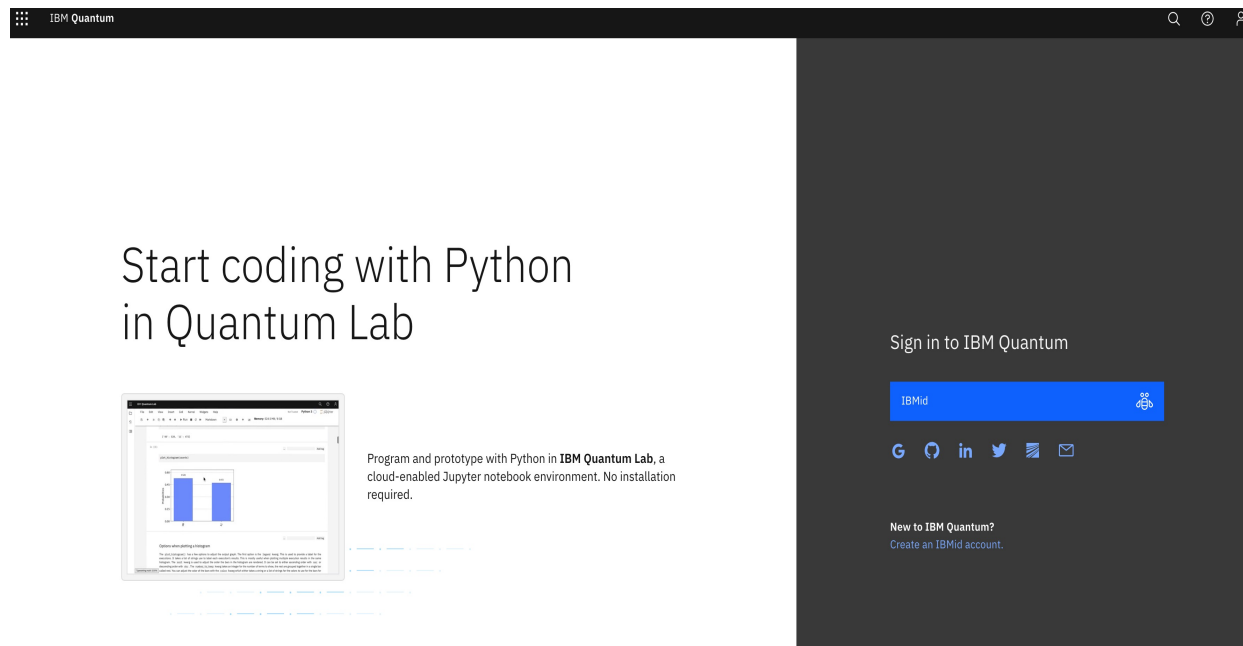
AquaとTerraを混ぜるとMigration上のエラーが発生する

出典 : https://qiskit.org/documentation/aqua_tutorials/Qiskit%20Algorithms%20Migration%20Guide.html



1.3 IBM Quantum Labの使用推奨

URL: <https://quantum-computing.ibm.com/>



2. シュレディンガー方程式

1. シュレディンガー方程式とは？
2. 時間に依存しないシュレディンガー方程式
3. ハミルトニアンとは？
4. 水素分子のシュレディンガー方程式

2.1 シュレディンガー方程式とは？

一般的な時間に依存するシュレディンガー方程式

$$i\hbar \frac{d}{dt} |\psi(t)\rangle = \mathcal{H} |\psi(t)\rangle$$

- 電子と原子核の状態が波動関数によって記述
- 飛び飛びのエネルギー準位の値を正確に求めることができる



2.2 時間に依存しないシュレディンガー方程式

$$\mathcal{H}_{\text{el}} |\Psi_n\rangle = E_n |\Psi_n\rangle$$

演算子 × 波動関数 = 固有値 × 波動関数

ハミルトニアン：観測可能な物理量をそれに対応する演算子で置き換えたエネルギー

2.3 ハミルトニアンとは？

$$\mathcal{H}(\boldsymbol{r}, \boldsymbol{R}) = T_n(\boldsymbol{R}) + T_e(\boldsymbol{R}) + U(\boldsymbol{r}, \boldsymbol{R})$$

原子核の運動エネルギー + 電子の運動エネルギー + 電子と原子核に働くクーロン相互作用

- ハミルトニアン：観測可能な物理量をそれに対応する演算子で置き換えたエネルギー
- 構成する電子の座標 \boldsymbol{r} と原子核の座標 \boldsymbol{R} の関数

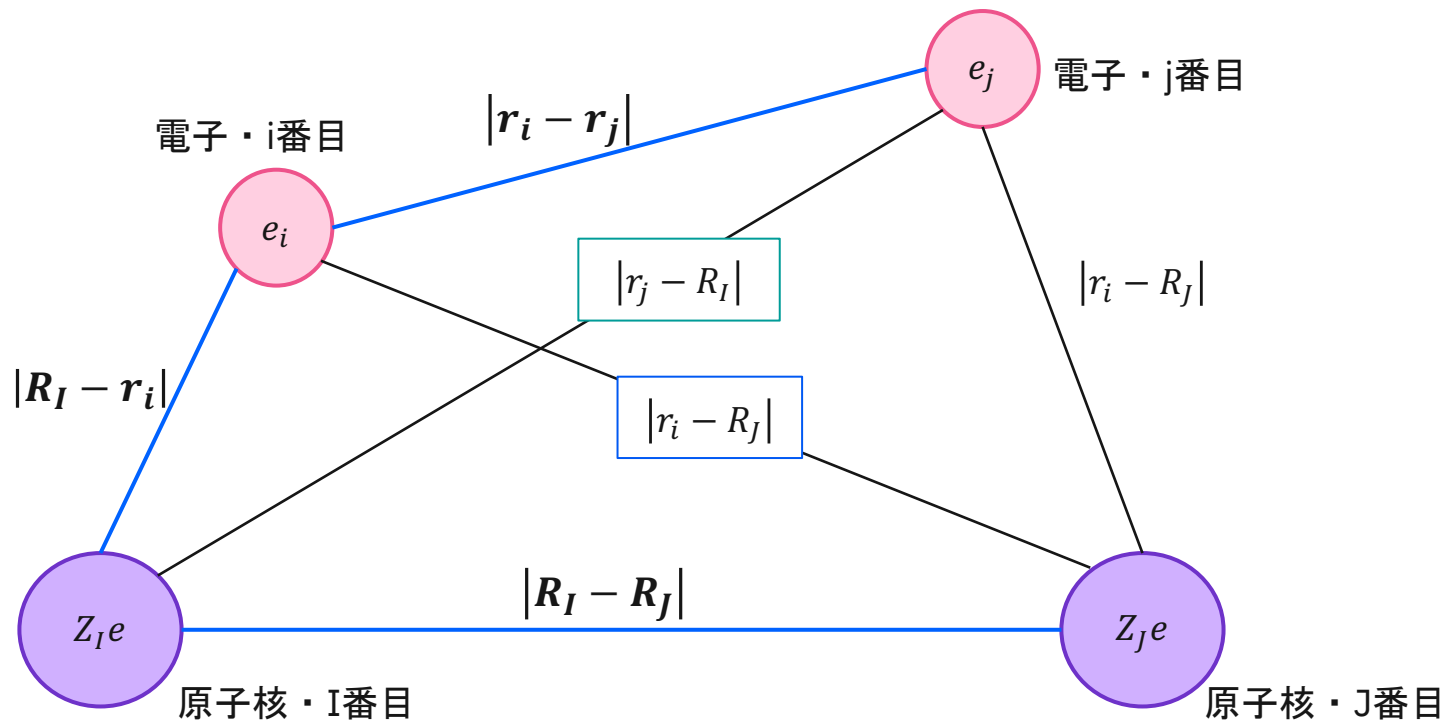
2.4 求めたいハミルトニアン

$$\mathcal{H} = \boxed{-\sum_I \frac{\nabla_{R_I}^2}{M_I}} \boxed{-\sum_i \frac{\nabla_{r_i}^2}{m_e}} \boxed{-\sum_I \sum_i \frac{Z_I e^2}{|R_I - r_i|} + \sum_i \sum_{j>i} \frac{e^2}{|r_i - r_j|} + \sum_I \sum_{J>I} \frac{Z_I Z_J e^2}{|R_I - R_J|}}$$

原子核の運動エネルギー + 電子の運動エネルギー + 電子と原子核に働くクーロン相互作用

- M : 原子核の質量
- m : 電子の質量
- R : 原子核の位置
- r : 電子の位置
- e : 電子
- Z : 電荷
- I, J : 原子核の番号
- i, j : 電子の番号

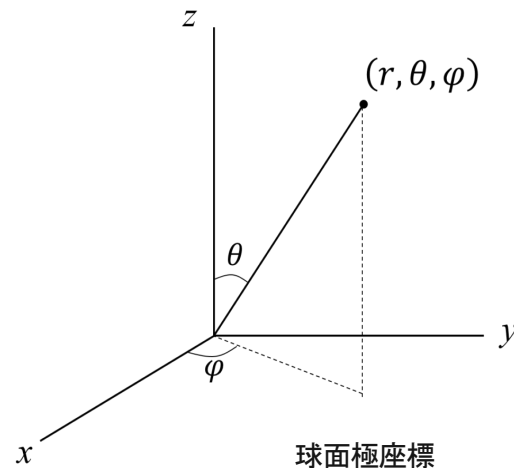
2.4 (続き) 原子核と電子の位置関係



2.5 水素分子のシュレディンガー方程式

$$\left[-\frac{\hbar^2}{2m} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) + U(x, y, z) \right] \psi(x, y, z) = E \psi(x, y, z)$$

- 波動関数 : (x, y, z) 座標で示される
- m : 電子の質量
- $-\frac{\hbar^2}{2m} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right)$: 電子と2つの原子核の運動エネルギー
- $U(x, y, z)$: クーロン静電引力によるポテンシャルエネルギー



3. ボルン・オッペンハイマー近似

1. ボルン・オッペンハイマー近似
2. 実際のハミルトニアンとの比較

3.1 ボルン・オッペンハイマー近似

電子の質量 \ll 原子核の質量 \rightarrow ゆっくりと動く原子核の運動エネルギーを無視



3.2 ボルン・オッペンハイマー近似前後の比較

一般のハミルトニアン

$$\mathcal{H} = \boxed{-\sum_I \frac{\nabla_{R_I}^2}{M_I}} \boxed{-\sum_i \frac{\nabla_{r_i}^2}{m_e}} \boxed{-\sum_I \sum_i \frac{Z_I e^2}{|R_I - r_i|} + \sum_i \sum_{j>i} \frac{e^2}{|r_i - r_j|} + \sum_I \sum_{J>I} \frac{Z_I Z_J e^2}{|R_I - R_J|}}$$

ボルン・オッペンハイマー近似後：原子核の運動エネルギーを無視したハミルトニアン

$$\mathcal{H}_{\text{el}} = \boxed{-\sum_i \frac{\nabla_{r_i}^2}{m_e}} \boxed{-\sum_I \sum_i \frac{Z_I e^2}{|R_I - r_i|} + \sum_i \sum_{j>i} \frac{e^2}{|r_i - r_j|}}.$$

4. Hartree- Fock法

1. 概要
2. 構成原理
3. Pauliの原理
4. Pauliの原理からわかる電子の性質
5. フントの規則

4.1 Hartree Fock法の概要

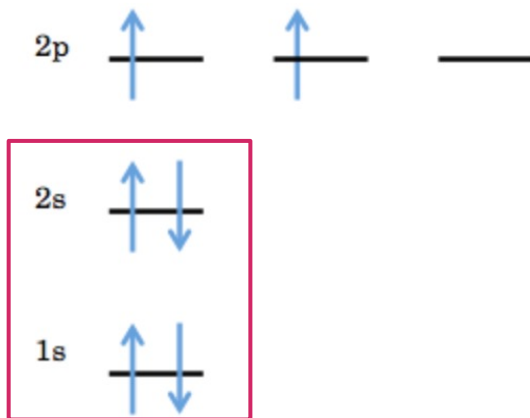
- 考え方：

電子間相互作用を近似的に取り扱うことにより、多電子系においても1電子波動関数を定義

- 目的：単一の電子配置で波動関数を近似したときに、エネルギー期待値が最小となるような分子軌道の組を求める
- 表現法：電子基底状態を単一の基底配置に対応するSlater行列式で表す
- Slater行列式：Pauliの原理を満足するために用いる行列式

4.2 構成原理

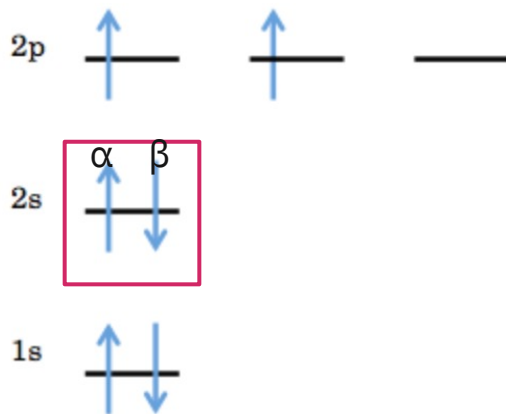
1. エネルギー準位の低い順から 1 つの準位に電子が 2 個ずつ詰まっていく
2. 1 つの準位に電子が 2 個入る時は、異なったスピンの状態をとる
3. 縮退順位に電子が入るときには、できる限り異なる順位に同じスピンの状態をとる



4.3 Pauliの原理

電子はフェルミ粒子であり、1つの状態に2個以上入ることができない。

1つの準位に2つの電子が入ると、スピンの向きは逆にならなければならない。

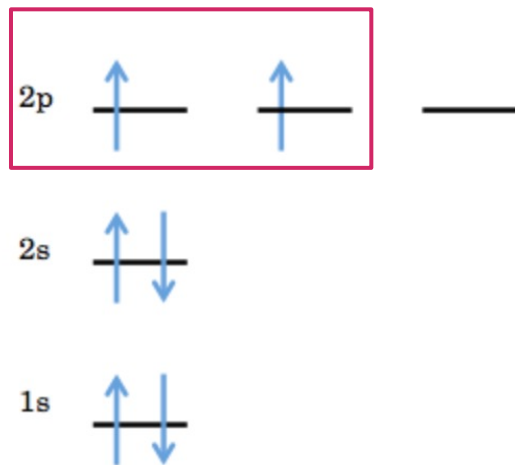


4.4 Pauliの原理からわかる電子の性質

- 2 個の電子の交換は反対称性
- 電子の位置座標とスピンの座標の両方を考慮した全波動関数は、電子の交換により符号が変わる

4.5 フントの規則

縮退副準位に 1 個ずつ入った電子は、スピンの向きを揃えた方が安定になる



5. Hartree Fock法のコード実行と説明

1. Hartree Fock法のプログラム
2. Qiskit Natureのオブジェクト・UnitsTypeとMoleculeについて
3. ドライバー・PySCFとは？



5.1 ハートリーフック法のプログラム

ハートリーフック法のプログラム

Qiskit natureのドライバーからUnitsTypeとMoleculeをインポートします

```
from qiskit_nature.drivers import UnitsType, Molecule
```

second_quantizationのドライバーからElectronicStructureDriverType, ElectronicStructureMoleculeDriverをインポート

```
from qiskit_nature.drivers.second_quantization import ElectronicStructureDriverType, ElectronicStructureMoleculeDriver
```

構造の位置を設定

電荷とスピン多重度の設定

全電荷: 0

スピン多重度: 1

```
molecule = Molecule(geometry=[[ 'H', [0., 0., 0.]],  
                                [ 'H', [0., 0., 0.735]]],  
                      charge=0, multiplicity=1)
```

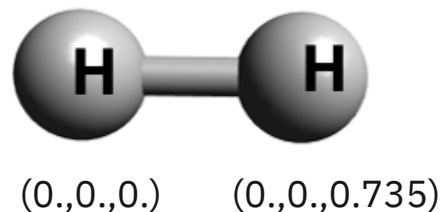
PySCFのドライバーを設定する

基底: STO-3G

```
driver = ElectronicStructureMoleculeDriver(molecule, basis='sto3g', driver_type=ElectronicStructureDriverType.PYSCF)
```

5.2 UnitsTypeとMoleculeについて

- UnitsType: 物理単位を示す
原子間の距離をオングストローム・Åに設定
- Molecule: 水素分子の定義をする
 - 位置: 原子座標を(x,y,z)軸の座標で設定
 - スピン多重度N: スピン状態に関する縮退度
 - 全電荷: 分子全体の総電荷
 - 自由度: 物理系の状態を記述するのに必要な最小限の変数の個数
 - 質量



水素分子の構造

5.3 PySCFについて



- Python-based Simulations of Chemistry Framework
- PySCFとは、Pythonで書かれた電子状態計算プラットフォーム
- 平均場法やポスト平均場法を用いて、分子、結晶、カスタムハミルトニアンの特徴のシミュレーションの実行が可能



6. フェルミオンを量子ビットにマッピングする方法

1. Jordan Wigner変換:

フェルミオンの演算子を、全く同じ性質を持つスピンの演算子に対応付ける

2. Bravyi Kitaev変換:

フェルミオンの第二量子化の生成消滅演算子をパウリ行列を用いて表すための方法の1つ
Jordan Wigner変換と比較すると、少ない量子ビットで計算することができる。

7. 電子と量子ビットの特徴

	電子	量子ビット
粒子の種類	フェルミオン	区別できる人工原子
状態	半整数のスピンを持つ粒子	エンタングルメント
特徴	波動関数の符号が反転する	同種粒子ではない

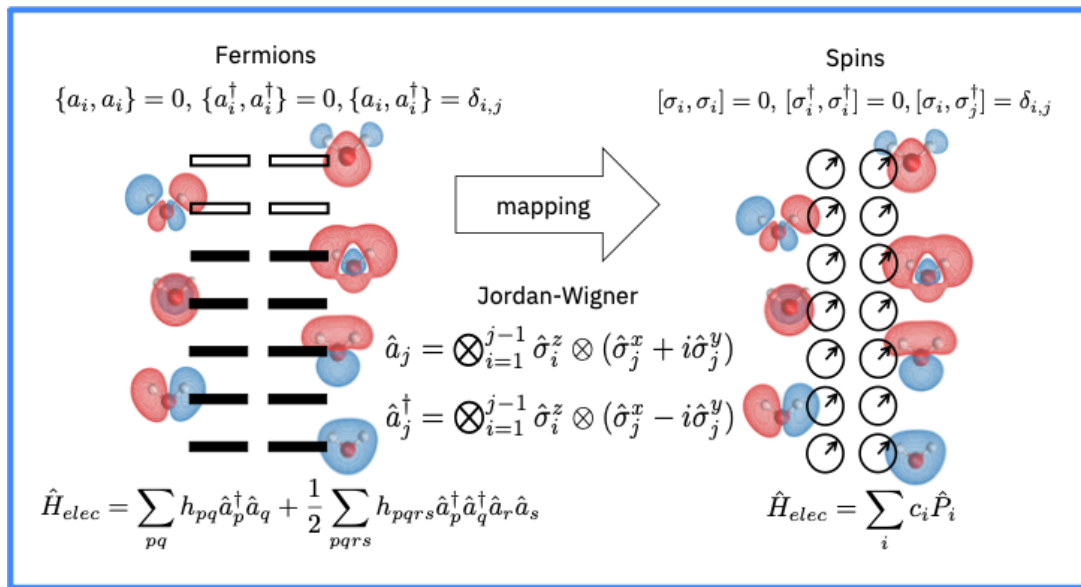
8. Jordan Wigner変換

1. Jordan Wigner変換とは？
2. 仕組み



8.1 Jordan Wigner 変換とは？

Jordan Wigner変換とは、電子の各スピン軌道を量子ビットにマッピングする



8.2 Jordan Wigner変換のしくみ

生成・消滅演算子の反交換関係を満足させ、波動関数を反対称化する

$$\hat{H}_{elec} = \sum_{pq} h_{pq} \hat{a}_p^\dagger \hat{a}_q + \frac{1}{2} \sum_{pqrs} h_{pqrs} \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_r \hat{a}_s$$

\hat{a}_p^\dagger :生成演算子, \hat{a}_p :消滅演算子



9. Jordan Wigner変換のコード説明

1. ドライバーからのインポート
2. 分子の情報をエンコードし、第二量子化を行う
3. Jordan Wigner変換
4. パリティマッピング



9.1 ドライバーからのインポート

```
# qiskit_nature.problems.second_quantizationのドライバーからElectronicStructureProblemをインポート
from qiskit_nature.problems.second_quantization import ElectronicStructureProblem

# qiskit_nature.converters.second_quantizationのドライバーからQubitConverterをインポート
from qiskit_nature.converters.second_quantization import QubitConverter

# qiskit_nature.mappers.second_quantizationのドライバーからJordanWignerMapper, ParityMapperをインポート
from qiskit_nature.mappers.second_quantization import JordanWignerMapper, ParityMapper
```



9.2 分子の情報をエンコードし、第二量子化を行う

```
# 分子の情報をエンコードするための電子のドライバー
```

```
es_problem = ElectronicStructureProblem(driver)
```

```
# second_quantizationドライバーと変換に基づいて作成された SecondQuantizedOp のリストを返す
```

```
# ハミルトニアン演算子、全粒子数演算子、全角運動量演算子、全磁化演算子、x,y,z双極子演算子を返す
```

```
second_q_op = es_problem.second_q_ops()
```

```
print(second_q_op[0])
```



9.3 Jordan Wigner変換

```
# Jordan Wigner変換を行うプログラム
# フェルミオン演算子をパウリ演算子に変換する
qubit_converter = QubitConverter(mapper=JordanWignerMapper())

# second_quantizationドライバーのQubitConverterのモジュール
# 量子ビット演算子に変換する
qubit_op = qubit_converter.convert(second_q_op[0])
print(qubit_op)
```



9.4 パリティマッピングで量子ビットを減らす

```
# パリティマッピング
```

```
# ビット数を減らす
```

```
qubit_converter = QubitConverter(mapper = ParityMapper(), two_qubit_reduction=True)  
qubit_op = qubit_converter.convert(second_q_op[0], num_particles=es_problem.num_particles)  
print(qubit_op)
```

- Jordan Wigner変換では計算する際、ビットが少なくとも4量子ビット必要
→対称性を考慮することで量子ビットの削減を可能にする
→そうすると、2量子ビットで計算することができる



参考文献、引用文献

- 大阪大学・藤井研究室・量子コンピューティング講義
- 安藤耕司研究室資料
- ザボ、オストランド『新しい量子化学』
- 量子化学 基礎から応用まで
- 「量子化学」のことが一冊でまるごとわかる
- すぐできる量子化学計算・ビギナーズマニュアル
- アトキンス物理化学
- 現代物理化学
- 実験医学HP
- Qiskit Nature Documentations
- Qiskit Global Summer School Chemistry
- Qiskit Chemistry Tutorial You tube
- IBM Quantum Open Lab
- Media記事 : ” Introducing Qiskit Nature ”
- Qiskit Textbook
- Qiskit Tutorials
- Quantum Tokyo
- IBM Quantum Challenge 2021

Thank you

Emi Adachi

