**0. Básicos**

Con PySCFDriver de qiskit\_nature, se crearon átomos y moléculas, calculando mediante el algoritmo VQE y el NumPyMinimumEigensolver la energía en el nivel fundamental.

He intentado obtener los autovectores resultantes después de las medidas, pero aún no he sido capaz.

Con los cálculos de VQE generado por nosotros será posible obtener los autovectores.

**Gráfico

Descripción generada automáticamente**

Barrido de distancias, calculando la energía con NumPyMinimumEigensolver, para el caso de dos átomos de Helio. No hay distancia de enace, en su forma natural, existe como átomos individuales en lugar de formar enlaces covalentes para crear moléculas.

Falta realizar medidas con el ordenador de IBM y comprobar que el coste computacional de un ordenador clásico y uno cuántico aumenta de forma exponencial y lineal, respectivamente.

**atom\_vqe**

También he resuelto un ansatz químico de un átomo de hidrógeno mediante el algoritmo vqe (Funciona pero creo que de esta forma crea dos electrones ya que los términos de interacción son no nulos)

**1. Hidrógeno**

Gráfico, Histograma

Descripción generada automáticamente

Barrido de distancias, calculando la energía con NumPyMinimumEigensolver, para el caso de una molécula de hidrógeno.

Gráfico

Descripción generada automáticamente con confianza baja

NumPyMinimumEigensolver y VQE. VQE: SLSQP, sin shots. Revisar esto.

Una captura de pantalla de un celular con texto e imagen

Descripción generada automáticamente con confianza media

Gráfico

Descripción generada automáticamente Gráfico, Gráfico de líneas

Descripción generada automáticamente

NumPyMinimumEigensolver y VQE. VQE: SLSQP, sin shots y con modelos de ruido de dos ordenadores: FakeVigo e ibmq\_qasm\_simulator. Revisar

Pantalla de un video juego

Descripción generada automáticamente con confianza media

Viendo el código creo que no están bien hechos con VQE. (Los siguiente sí)

Al principio hice los cálculos con el optimizador SPSA, luego descubrí que funciona mejor con COBYLA, poniendo el punto inicial. Además, al principio realicé las medidas empleando el ansatz TwoLocal, UCCSD funciona mejor.

optimizer = SPSA, shots = 128, ansatz = TwoLocal (sin estado inicial):Gráfico, Histograma

Descripción generada automáticamente Gráfico, Histograma

Descripción generada automáticamente

El TwoLocal solo tiene dos puertas lógicas, son muy pocas, igual por ese motivo los resultados son bastante peores.

optimizer = SPSA, shots = 128, ansatz = UCCSD (sin estado inicial):

Gráfico, Gráfico de líneas

Descripción generada automáticamenteDiagrama

Descripción generada automáticamente

optimizer = COBYLA, shots = 128 (con estado inicial):

Gráfico, Histograma

Descripción generada automáticamente

optimizer = COBYLA, shots = 16384 (con estado inicial):

Gráfico, Gráfico de líneas

Descripción generada automáticamente Diagrama

Descripción generada automáticamente

Se ha restado la energía obtenida por el algoritmo vqe a la energía 'teórica'. Se ha obtenido un histograma para los distintos casos, variando el número de shots. Las diferencias de energía tienden a la izquierda (el algoritmo VQE determina la energía siempre por encima de la del estado fundamental, si está por debajo es debido al ruido). Hay algún punto muy separado del 0 (puede deberse a mínimos locales, pero al usar COBYLA lo más probable es que sea debido a que el número de iteraciones es demasiado bajo).

--- Intentar ejecutar en un ordenador de IBM. ---

Intentar hacerlo de nuevo creando un nuevo algoritmo VQE con COBYLA, los resultados para shots bajos son algo extraños.

**2. Yukawa**

**2.1 h\_g\_coefficients**

Se ha intentado crear un hamiltoniano poniendo los coeficientes h\_pq y g\_pqrs simulando un átomo de hidrógeno con un solo electrón.

No se ha conseguido (supongo que está creando 2 electrones al crear el problem químico ya que los elementos de la matriz g\_pqrs son no nulos)



**2.2 hydrogen (cambiar el título)**

Se ha conseguido crear un hamiltoniano poniendo los coeficientes h\_pq y g\_pqrs a mano. Para ello, se ha creado el algoritmo VQE de una manera diferente, creando un ansatz forzando para que el número de partículas sea 1 y el de orbitales sea la mitad de los orbitales de spin.

Texto

Descripción generada automáticamentePantalla de computadora con letras

Descripción generada automáticamente con confianza media

Intentar entender el motivo de porque antes no funcionaba e intentar entender lo que se ha hecho para que funcione (ansatz UCCSD).

Cambiar los valores de la matriz h1\_b parece que no afectan al resultado, solo afectan los de h1\_a (es por el num\_particles de UCCSD, si se pone (0, 1) coge los valores de h1\_b) (el primer elemento indica el número de partículas con spin up y el segundo término las de spin down. Habrá que tener esto en cuenta cuando hagamos sistemas con más de una partícula.)

**2.3 yukawa\_1**

He creado el hamiltoniano del potencial de Yukawa, para Z=1. Lo he hecho a partir de las soluciones radiales de la ecuación de ondas. He usado los valores con l=0.

Por ejemplo, para h\_22 por qué se usa <2 0 0 | H | 2 0 0> pero no se uso <2 1 0 | H | 2 1 0>?

<2 0 0 | H | 2 1 0 > = 0, debido a que se anula porque los armónicos esféricos no son iguales.

Intentar automatizar el cálculo de los armónicos esféricos para obtener cualquiera de ellos a partir de los polinomios de Laguerre.

Ver qué pasa con valores de alpha distintos de 0 (si es 0 es igual que el átomo de hidrógeno)

**2.4 yukawa\_2**

Se ha mejorado el código de Yukawa\_1mediante los polinomios de Laguerre, los cuales me han permitido crear tantas funciones de ondas radiales como quiera. (citar bibilografía)

Si creo un hamiltoniano con muchos estados, ahora he estado probando con 7, al resolverlo con el algoritmo VQE bastantes veces me da como resultado la energía del segundo estado, -0.125. Entiendo que aumentando el número de shots se soluciona el problema, pero supongo que hay que tener en cuenta que cuanto mayor sea el número de estados, la medida será menos precisa. Cuanto más aumente N, más veces me da un valor incorrecto. Parece que me da la energía del segundo estado, -0.125.

Creo que he conseguido encontrar lo que estaba haciendo mal, aunque hay un par de cosas que no entiendo muy bien. Para solucionarlo he estado buscando y he encontrado esta página con un ejemplo diferente, pero que define el ansatz igual que nosotros. En el punto 5 dice que es importante inicializar las medidas con todos los parámetros en cero para así poder comenzar desde el estado inicial de Hartree-Fock que he definido en el ansatz.

También depende mucho del optimizador, por lo general, COBYLA es el que mejores resultados está dando.

**2.5 yukawa\_alpha**

Para unas medidas con shots = None:

No entiendo muy bien que hace exactamente, pero creo que no sigue el mismo proceso que si le introduces un número de shots. Esto pone en la documentación: shots (None or int) – The number of shots. If None and approximation is True, it calculates the exact expectation values. Otherwise, it calculates expectation values with sampling. Por eso siempre me da el mismo valor de -0.499999999999906.

Los valores con mayor N parecen fallar en algunos puntos del final.

Que hace exactamente este Estimator a diferencia de AerEstimator?

Si defino shots=1 y maxiter=1 dentro del optimizador, que es el número máximo de iteracciones que tiene el método del gradiente, que en este caso estamos usando el optimizador SPSA. Al hacer esto me sale como resultado la energía de uno de los estados y parece que todos se miden con una misma probabilidad. Se podría representar un gráfico de barras con las veces que se repite cada autoestado. Al ir aumentando shots y maxiter cada vez habrá un mayor número de valores en el autoestado de -0.5.

---Igual es interesante estudiar estos gráficos para ver hasta que punto merece la pena aumentar shots y maxiter teniendo en cuenta que al aumentarlos, el tiempo también aumenta---

Gráfico

Descripción generada automáticamente

**2.6 yukawa\_alpha\_exact/results\_1**

Resultado muy similar al obtenido en yukawa\_alpha.

Se ha obtenido la solucion exacta de la energía, obteniendo los autovalores del hamiltoniano de manera exacta, sin emplear el algoritmo VQE, para luego comprobar los datos obtenidos.

Importante:

Para un número elevado de N los tiempos empiezan a ser muy altos, por lo tanto, explicar cuál es el motivo principal para usar los ordenadores cuánticos: para problemas complejos con un número elevado de partículas, los ordenadores clásicos escalan exponencialmente, mientras que los cuánticos, linealmente.

Revisar si es interesante los casos de alpha = 1 (punto de corte que s eestudia en próximos apartados)

Para cualquier N puede observarse un comportamiento similar hasta alpha = 1, a partir de ese punto pueden observarse los diferentes comportamientos. Cuanto mayor es N, más tiende a 0. Revisar si para N infinito la energía nunca es superior a 0. (siempre hay punto de corte, aunque N sea infinito, en el drive hay artículos explicando esto, citarlos para comparar los puntos de corte obtenidos).

Gráfico

Descripción generada automáticamente

**Different\_base**

Z = 0.5

**Gráfico

Descripción generada automáticamente con confianza media Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media**

Z = 2

Diagrama

Descripción generada automáticamente con confianza media

He ampliado la zona para alpha>1 para Z<1 ya que antes salían resultados interesantes.

Z = 0.75

Gráfico

Descripción generada automáticamente con confianza media

Mostrar las gráficas que demuestran que, para obtener el punto crítico, utilizar bases con un Z distinto de 1 se obtienen mejores resultados. También varía con el valor de N.

Con Z = 0.75 y N=3 parece que se obtienen buenos resultados (ver bibilografía, para comparar los resultados).

Yo obtengo un valor entre 1.1408 y 1.1449. Con mathematica da un valor de 1.142.

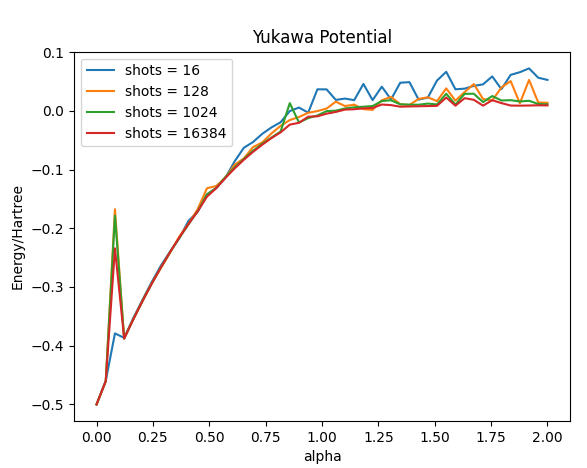
**2.7 yukawa\_alpha\_aer**

Se ha introducido un error mediante los shots.

En este caso si no meto la posición inicial, al contrario que el caso anterior, las medidas parece que funcionan mejor. No entiendo el motivo. Probar con COBYLA en vez de SPSA.

Cuanto más shots, menos error, como era de esperar. Pero más tiempo de ejecución.

He probado con el optimizador SPSA (creo que funciona por gradiente, el gradiente puede funcionar mal si encuentra minimos locales o si la pendiente es cercana a 0, por lo tanto, tengo que volver a tomar las medidas con COBYLA ya que parece que van mejor)

 Gráfico, Histograma

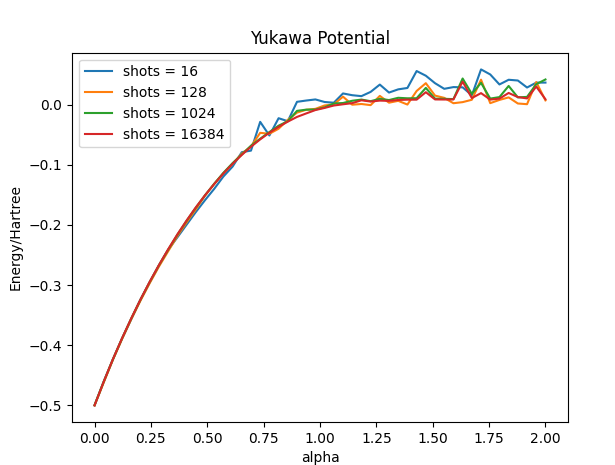
Descripción generada automáticamente

SPSA

Una de las veces, con el SPSA uno de los puntos se iba mucho. Raro ya que para ese punto se iba en todos los casos, con los diferentes shots. no entiendo muy bien el motivo. Creo que fallaba por no poner lo de initial\_point. Vuelvo a probar poniendolo. Tambien falla con eso. Parece que COBYLA funciona mejor.

He vuelto a hacerlo con COBYLA y obtengo exactamente los mismos resultados, con los mismos valores. Puede ser debido a lo de seed? Sí, cambiando seed se obtiene distinto resultado. Por lo que veo el primer valor siempre es el mismo, pero si vuelves a hacer una medida en ese mismo punto cambia el resultado. Si pones initial\_point siempre da lo mismo.

Lo he hecho con COBYLA tanto sin punto inicial, como con el. Interpretar los resultados en hydrogen\_noise\_distribution. Histogramas.

 Gráfico

Descripción generada automáticamente

COBYLA

Gráfico

Descripción generada automáticamente Imagen que contiene Gráfico

Descripción generada automáticamente

COBYLA con initial point

Poner aún así los resultados obtenidos con SPSA y con COBYLA, explicando cómo funciona cada uno.

También tengo que entender que hace exactamente el ansatz UCCSD. Igual otros tipos de ansatz ayuda a resolver sistemas atómicos y moleculares.

IMPORTANTE vqe.initial\_point: Con COBYLA me da siempre lo mismo, con SPSA no da lo mismo.

**2.8 yukawa\_alpha\_noise**

Modelo de ruido como en el hidrógeno.

Gráfico

Descripción generada automáticamente

SLSQP

SLSQP con initial\_point parece que no tiene error de medidas. Sin el parece que si lo tiene.

Gráfico

Descripción generada automáticamente

COBYLA

Hacer noise\_distribution

SLSQP con initial\_point parece que no tiene error de medidas.

Sin el parece que si lo tiene

Voy a hacer el modelo de ruido con este optimizador.

COBYLA parece que algo de error si que tiene y SPSA tiene mucho error. Prefiero hacerlos sin error de momento.

**2.9 yukawa\_alpha\_fixed**

Muchas medidas para un mismo alpha, para ver la dispersión de las medidas.

Con COBYLA dejo lo de initial\_point, pero quito la seed, para que no de siempre el mismo valor.

alpha=0.1

Gráfico

Descripción generada automáticamente con confianza baja

alpha=0.9

Gráfico

Descripción generada automáticamente

alpha=0.99

Gráfico

Descripción generada automáticamente

alpha=1.2 (relacionado con la sección 2.11)

Gráfico, Gráfico de barras

Descripción generada automáticamente

alpha=1.5

Gráfico, Histograma

Descripción generada automáticamente

A mayor shots, menor dispersión, como era de esperar.

También lo he hecho para SPSA:

alpha=0.1

Imagen que contiene Gráfico

Descripción generada automáticamente

alpha=1.5

Gráfico, Gráfico de barras, Histograma

Descripción generada automáticamente

SLSQP:

alpha=0.1 (hay algunos puntos en los cuales la energía es muy diferente a la esperada)

Gráfico, Gráfico de barras, Histograma

Descripción generada automáticamente

Comparando lo gráficos de Alpha=0.1 para los tres optimizadores, puede verse que COBYLA da unos resultados mucho mejores.

También se ha hecho para un aplhas cercanos a 1 (0.9 y 0.99) ya que en las anteriores pruebas se observó que al ir aumentando el valor de alpha los errores eran más significativos y se alejaban más del valor esperado.

Parece que solo dan resultados positivos para shots pequeños, al ir aumentando los disparos, el error se va reduciendo.

Se ha empleado COBYLA ya que es el que mejor funciona.

**2.10 yukawa\_3\_Z!=1**

He obtenido la matriz del hamiltoniano para un Z != 1, antes lo tenía mal, me faltaba una Z en la integral.

**2.11 yukawa\_alpha\_aer\_1.2 (punto crítico)**

Documento compartido de "problemas cuánticos".

Sí hay un punto de corte (en torno a 1.2 para nuestro caso) y sí hay estados de energía muy pequeña pero positiva por encima de este punto de corte. Podemos mirar cómo afecta el tema de computación cuántica a su cálculo.

Se ha estudiado la zona entre 0.9 y 1.3

Corta entre 1.10 y 1.15

Z=1:

Gráfico, Histograma

Descripción generada automáticamente

Z = 0.75, N = 3 (diferente base da mejores resultados que Z=1) (comparar los resultados con los de la bibliografía)

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Parece que fallaba por esto:

- sp.integrate(globals()[f'u\_{i}0']\*globals()[f'u\_{j}0'] \* Z \* sp.exp(-alpha\*r)/r, (r, 0, sp.oo)) <--- Estaba mal

- sp.integrate(globals()[f'u\_{i}0']\*globals()[f'u\_{j}0'] \* sp.exp(-alpha\*r)/r, (r, 0, sp.oo))

Arreglandolo corta entre 1.1204 y 1.1245, cuantos más shots, más cerca parece estar.

Falta pulir un poco esto porque para las que solo tienen un punto no sé qué error coger

**3. Helium**

**3.1 atom\_vqe**

He estado probando PySCFDriver con diferentes bases para no tener que montar en el hidrógeno las matrices del hamiltoniano a mano ya que las integrales de dos cuerpos se pueden complicar.

He probado con varias bases pero ninguna me da una matriz diagonal con autoestados -1/(2\*n^2). Además, las matrices de dos cuerpos me dan listas muy largas en vez de matrices.

**4. Potential well**

**4.1 potential\_well\_two\_body\_integral**

He hecho un potencial infinito de anchura L con dos electrones, creando el hamiltoniano con las integrales de un cuerpo y de dos cuerpos, para tener en cuenta las interacciones entre los electrones.

He calculado la energía sin interacción, ya que con interacción no puedo crear la matriz de interacción ya que las integrales se hacen infinitas.

Texto

Descripción generada automáticamente

Cuando no hay interacción obtengo con VQE siempre el mismo valor, 2 veces la energía del estado de n=1. Lo he hecho con COBYLA.

Se ha cambiado el potencial por uno de contacto, que es una delta de dirac, para poder obtener la matriz de interacción.

**4.2 potential\_well\_delta\_potential**

Calendario

Descripción generada automáticamente

COBYLA y N=3

Gráfico

Descripción generada automáticamente

COBYLA y N=2

Gráfico, Histograma

Descripción generada automáticamente

Por perturbaciones no da el mismo resultado ya que no se puede resolver este problema por perturbaciones.

Resultado obtenido: 9.8 + 1.5

Perturbaciones: 9.8 + 1

**4.3 potential\_well\_delta\_potential\_V0**

Barrido de valores de v\_0

N = 2:

Gráfico, Gráfico de líneas, Histograma

Descripción generada automáticamente Gráfico

Descripción generada automáticamente

El tiempo aumenta mucho al pasar a N = 3

COBYLA, N=3 para valores más altos de V\_0:

Gráfico, Gráfico de líneas

Descripción generada automáticamente

COBYLA, N=2 para valores más altos de V\_0 (Para V~0: pi^2+1.5\*v y para v grande: ~v/2)

**Gráfico, Gráfico de líneas

Descripción generada automáticamente**

Para v0 grandes no me sale lo esperado, mientras que para v0~0 si

Hasta 100 va bastante parecido, pero con N=3 debería saturar mucho más rápido

Si lo hacemos hasta v0=1, es todo muy fácil de explicar y para lo que queremos es suficiente.

El método que estamos usando, aunque no lo diga explícitamente es perturbativo. El sumatorio de la parte de la matriz de interacción, en realidad es el de primer orden (en la documentación ponen que podría haber términos superiores que se desprecian)

Nos quedamos con v0 menor que 1 para que pueda resolverse por este método perturbativo.