

hhl-non-hermitian

December 14, 2022

1 ECE 396 - HHL Algorithm

Problem 1: Implement a 3x3 invertible but non-hermitian A in Qiskit

```
[3]: import numpy as np
```

We need to define a 3 x 3 invertible but non-hermitian A in Qiskit. That is, the following two qualities must hold:

$$AA^{-1} = A^{-1}A = I_n$$
$$A \neq A^\dagger$$

If we restrict ourselves to only real values, the matrix we can choose must then follow the two qualities: invertible and $A \neq A^T$.

One such matrix is the following:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

```
[4]: A = np.array([[1, 0, 0], [0, 1, 0], [0, 1, 1]])  
print(A)
```

```
[[1 0 0]  
 [0 1 0]  
 [0 1 1]]
```

```
[5]: b = np.array([1, 2, 3])
```

Before I can use the HHL algorithm, we need to convert the non-hermitian matrix into a hermitian matrix form. We can complete this by performing the operation defined in the article. That is, define:

$$C = \begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix}$$

C is now hermitian and can be used to solve $C \cdot y = \begin{bmatrix} b \\ 0 \end{bmatrix}$; where $y = \begin{bmatrix} 0 \\ x \end{bmatrix}$

```
[6]: from qiskit import Aer
from qiskit.algorithms.linear_solvers.hhl import HHL

backend = Aer.get_backend('aer_simulator')
hhl = HHL(quantum_instance=backend)

accurate_solution = hhl.solve(A, b)
```

/tmp/ipykernel_806777/524707097.py:5: DeprecationWarning: The HHL class is deprecated as of Qiskit Terra 0.22.0 and will be removed no sooner than 3 months after the release date.
It is replaced by the tutorial at https://qiskit.org/textbook/ch-applications/hhl_tutorial.html

```
hhl = HHL(quantum_instance=backend)
```

```
-----
QiskitError                                Traceback (most recent call last)
/home/tybens/Desktop/Junior/ECE396/finalproj/hhl-non-hermitian.ipynb Cell 7' in
↳<cell line: 9>()
    <a href='vscode-notebook-cell:/home/tybens/Desktop/Junior/ECE396/finalproj/hhl-non-hermitian.ipynb#ch00000006?line=3'>4</a> backend = Aer.
↳get_backend('aer_simulator')
    <a href='vscode-notebook-cell:/home/tybens/Desktop/Junior/ECE396/finalproj/hhl-non-hermitian.ipynb#ch00000006?line=4'>5</a> hhl =
↳HHL(quantum_instance=backend)
----> <a href='vscode-notebook-cell:/home/tybens/Desktop/Junior/ECE396/finalproj/hhl-non-hermitian.ipynb#ch00000006?line=8'>9</a> accurate_solution = hhl.
↳solve(A, b)

File ~/local/lib/python3.10/site-packages/qiskit/algorithms/linear_solvers/hhl
↳py:542, in HHL.solve(self, matrix, vector, observable, observable_circuit,
↳post_processing)
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/algorithms/linear_solvers/hhl.py?line=536'>537</a>         raise ValueError(
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/algorithms/linear_solvers/hhl.py?line=537'>538</a>             "If observable
↳is passed, observable_circuit and post_processing cannot be set."
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/algorithms/linear_solvers/hhl.py?line=538'>539</a>         )
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/algorithms/linear_solvers/hhl.py?line=540'>541</a>     solution =
↳LinearSolverResult()
--> <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/algorithms/linear_solvers/hhl.py?line=541'>542</a>     solution.state = self.
↳construct_circuit(matrix, vector)
```

```

    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
    ↪algorithms/linear_solvers/hhl.py?line=542'>543</a> solution.euclidean_norm =
    ↪self._calculate_norm(solution.state)
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
    ↪algorithms/linear_solvers/hhl.py?line=544'>545</a> if observable is not None
    ↪or observable_circuit is not None:

File ~/local/lib/python3.10/site-packages/qiskit/algorithms/linear_solvers/hhl
    ↪py:357, in HHL.construct_circuit(self, matrix, vector, neg_vals)
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
    ↪algorithms/linear_solvers/hhl.py?line=354'>355</a> nb = int(np.
    ↪log2(len(vector)))
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
    ↪algorithms/linear_solvers/hhl.py?line=355'>356</a> vector_circuit =
    ↪QuantumCircuit(nb)
--> <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
    ↪algorithms/linear_solvers/hhl.py?line=356'>357</a> vector_circuit.
    ↪isometry(vector / np.linalg.norm(vector), list(range(nb)), None)
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
    ↪algorithms/linear_solvers/hhl.py?line=358'>359</a> # If state preparation is
    ↪probabilistic the number of qubit flags should increase
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
    ↪algorithms/linear_solvers/hhl.py?line=359'>360</a> nf = 1

File ~/local/lib/python3.10/site-packages/qiskit/extensions/quantum_initialize /
    ↪isometry.py:621, in iso(self, isometry, q_input, q_ancillas_for_output,
    ↪q_ancillas_zero, q_ancillas_dirty, epsilon)
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
    ↪extensions/quantum_initializer/isometry.py?line=616'>617</a> if
    ↪isinstance(q_ancillas_dirty, QuantumRegister):
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
    ↪extensions/quantum_initializer/isometry.py?line=617'>618</a>
    ↪q_ancillas_dirty = q_ancillas_dirty[:]
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
    ↪extensions/quantum_initializer/isometry.py?line=619'>620</a> return self.
    ↪append(
--> <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
    ↪extensions/quantum_initializer/isometry.py?line=620'>621</a>
    ↪Isometry(isometry, len(q_ancillas_zero), len(q_ancillas_dirty),
    ↪epsilon=epsilon),
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
    ↪extensions/quantum_initializer/isometry.py?line=621'>622</a> q_input +
    ↪q_ancillas_for_output + q_ancillas_zero + q_ancillas_dirty,
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
    ↪extensions/quantum_initializer/isometry.py?line=622'>623</a> )

File ~/local/lib/python3.10/site-packages/qiskit/extensions/quantum_initialize /
    ↪isometry.py:86, in Isometry.__init__(self, isometry, num_ancillas_zero,
    ↪num_ancillas_dirty, epsilon)
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
    ↪extensions/quantum_initializer/isometry.py?line=83'>84</a> m = np.
    ↪log2(isometry.shape[1])

```

```

    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
↳extensions/quantum_initializer/isometry.py?line=84'>85</a> if not n.
↳is_integer() or n < 0:
---> <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
↳extensions/quantum_initializer/isometry.py?line=85'>86</a>         raise
↳QiskitError(
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
↳extensions/quantum_initializer/isometry.py?line=86'>87</a>             "The number
↳of rows of the isometry is not a non negative power of 2."
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
↳extensions/quantum_initializer/isometry.py?line=87'>88</a>         )
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
↳extensions/quantum_initializer/isometry.py?line=88'>89</a> if not m.
↳is_integer() or m < 0:
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
↳extensions/quantum_initializer/isometry.py?line=89'>90</a>         raise
↳QiskitError(
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
↳extensions/quantum_initializer/isometry.py?line=90'>91</a>             "The number
↳of columns of the isometry is not a non negative power of 2."
    <a href='file:///home/tybens/.local/lib/python3.10/site-packages/qiskit/
↳extensions/quantum_initializer/isometry.py?line=91'>92</a>         )

QiskitError: 'The number of rows of the isometry is not a non negative power of
↳2.'
```

[]: