Optimising quantum circuits is generally hard

John van de Wetering University of Amsterdam
Matthew Amy Simon Fraser University

September 11th 2024 — IWQC

T-count optimisation: Motivation

- If you want to do fault-tolerant quantum computing, need a discrete gate set.
- Popular choice: Clifford+T, where $T = \mathrm{diag}(1, e^{i\frac{\pi}{4}})$.
- In many architectures Cliffords are cheap, but T is expensive.
- Hence: want to optimise T-count.

The T-count decision problem

The problem T-COUNT:

The T-count decision problem

The problem T-COUNT:
We are given

- unitary Clifford+T circuit implementing a unitary U, and
- an integer k.

The T-count decision problem

The problem T-COUNT:
We are given

- unitary Clifford+T circuit implementing a unitary U, and
- an integer k.

Decide whether there exists a unitary Clifford+T circuit that implements U which uses at most k T gates.

The T-count decision problem

The problem T-COUNT:
We are given

- unitary Clifford+T circuit implementing a unitary U, and
- an integer k.

Decide whether there exists a unitary Clifford+T circuit that implements U which uses at most k T gates.

Note: optimisation problem follows from decision problem via binary search with logarithmic number of queries in length of circuit.

Main result #1

Theorem

T-COUNT is NP-hard under polynomial-time Turing reductions.

Main result #1

Theorem

T-COUNT is NP-hard under polynomial-time Turing reductions.

Proof by reduction from Boolean satisfiability.

Proof

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function given in some poly-size description.
Goal: determine whether there is $\vec{x} \in \{0,1\}^n$ such that $f(\vec{x}) = 1$.

Proof

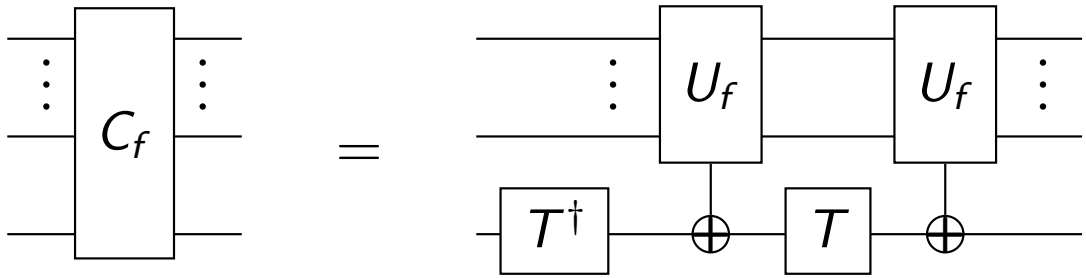
Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function given in some poly-size description.
Goal: determine whether there is $\vec{x} \in \{0,1\}^n$ such that $f(\vec{x}) = 1$.

- Using standard techniques build the classical oracle U_f implementing $U_f |\vec{x}, y\rangle = |\vec{x}, y \oplus f(\vec{x})\rangle$ as an $(n+1)$-qubit poly(n) size Clifford+T quantum circuit.

Proof

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function given in some poly-size description.
Goal: determine whether there is $\vec{x} \in \{0,1\}^n$ such that $f(\vec{x}) = 1$.

- Using standard techniques build the classical oracle U_f implementing $U_f |\vec{x}, y\rangle = |\vec{x}, y \oplus f(\vec{x})\rangle$ as an $(n+1)$-qubit poly(n) size Clifford+T quantum circuit.
- Consider the following circuit C_f:



- We see then that

$$C_f |\vec{x}, y\rangle = e^{i\frac{\pi}{4}(1-2y)f(\vec{x})} |\vec{x}, y\rangle .$$

Proof cont.

$$C_f \left|\vec{x}, y\right\rangle \;=\; e^{i\frac{\pi}{4}(1-2y)f(\vec{x})} \left|\vec{x}, y\right\rangle$$

- If f is not satisfiable: $C_f = \mathrm{id}$.

Proof cont.

$$C_f \left|\vec{x}, y\right\rangle \;=\; e^{i\frac{\pi}{4}(1-2y)f(\vec{x})} \left|\vec{x}, y\right\rangle$$

- If f is not satisfiable: $C_f = \mathrm{id}$.
- If f is *always* satisfiable:

$$C_f \left|\vec{x}, y\right\rangle = e^{i\frac{\pi}{4}(1-2y)} \left|\vec{x}, y\right\rangle \;=\; e^{i\frac{\pi}{4}} e^{-i\frac{\pi}{2}y} \left|\vec{x}, y\right\rangle \;=\; e^{i\frac{\pi}{4}} \left(I_n \otimes S^\dagger\right) \left|\vec{x}, y\right\rangle .$$

Proof cont.

$$C_f \left|\vec{x}, y\right\rangle \;=\; e^{i\frac{\pi}{4}(1-2y)f(\vec{x})} \left|\vec{x}, y\right\rangle$$

- If f is not satisfiable: $C_f = \mathrm{id}$.
- If f is *always* satisfiable:

$$C_f \left|\vec{x}, y\right\rangle = e^{i\frac{\pi}{4}(1-2y)} \left|\vec{x}, y\right\rangle \;=\; e^{i\frac{\pi}{4}} e^{-i\frac{\pi}{2}y} \left|\vec{x}, y\right\rangle \;=\; e^{i\frac{\pi}{4}} \left(I_n \otimes S^\dagger\right) \left|\vec{x}, y\right\rangle .$$

- Hence, in both cases C_f is Clifford, and must have T-count zero.

Proof cont.

Suppose now that $\exists \vec{z}_1, \vec{z}_2$ with $f(\vec{z}_1) = 1$ and $f(\vec{z}_2) = 0$. Then:

$$C_f \left|\vec{z}_1, 0\right\rangle = e^{i\frac{\pi}{4}} \left|\vec{z}_1, 0\right\rangle \qquad \text{and} \qquad C_f \left|\vec{z}_2, 0\right\rangle = \left|\vec{z}_2, 0\right\rangle .$$

- Claim: C_f is then non-Clifford.

Proof cont.

Suppose now that $\exists \vec{z}_1, \vec{z}_2$ with $f(\vec{z}_1) = 1$ and $f(\vec{z}_2) = 0$. Then:

$$C_f \left|\vec{z}_1, 0\right\rangle = e^{i\frac{\pi}{4}} \left|\vec{z}_1, 0\right\rangle \qquad \text{and} \qquad C_f \left|\vec{z}_2, 0\right\rangle = \left|\vec{z}_2, 0\right\rangle .$$

- Claim: C_f is then non-Clifford.
- Recall: U is Clifford if $U^\dagger \vec{P} U$ is Pauli for every Pauli $\vec{P}$.

Proof cont.

Suppose now that $\exists \vec{z}_1, \vec{z}_2$ with $f(\vec{z}_1) = 1$ and $f(\vec{z}_2) = 0$. Then:

$$C_f \left|\vec{z}_1, 0\right\rangle = e^{i\frac{\pi}{4}} \left|\vec{z}_1, 0\right\rangle \qquad \text{and} \qquad C_f \left|\vec{z}_2, 0\right\rangle = \left|\vec{z}_2, 0\right\rangle .$$

- Claim: C_f is then non-Clifford.
- Recall: U is Clifford if $U^\dagger \vec{P} U$ is Pauli for every Pauli $\vec{P}$.
- Take $X^{\vec{z}_1 \oplus \vec{z}_2} := X^{(\vec{z}_1 \oplus \vec{z}_2)_1} \otimes \cdots \otimes X^{(\vec{z}_1 \oplus \vec{z}_2)_n}$. Then:

$$C_f^\dagger X^{\vec{z}_1 \oplus \vec{z}_2} C_f \left|\vec{z}_1, 0\right\rangle = e^{i\frac{\pi}{4}} C_f^\dagger X^{\vec{z}_1 \oplus \vec{z}_2} \left|\vec{z}_1, 0\right\rangle = e^{i\frac{\pi}{4}} C_f^\dagger \left|\vec{z}_2, 0\right\rangle = e^{i\frac{\pi}{4}} \left|\vec{z}_2, 0\right\rangle .$$

- A Pauli can't introduce a $e^{i\frac{\pi}{4}}$ phase, so C_f not Clifford.

Proof cont.

Suppose now that $\exists \vec{z}_1, \vec{z}_2$ with $f(\vec{z}_1) = 1$ and $f(\vec{z}_2) = 0$. Then:

$$C_f \left|\vec{z}_1, 0\right\rangle = e^{i\frac{\pi}{4}} \left|\vec{z}_1, 0\right\rangle \qquad \text{and} \qquad C_f \left|\vec{z}_2, 0\right\rangle = \left|\vec{z}_2, 0\right\rangle .$$

- Claim: C_f is then non-Clifford.
- Recall: U is Clifford if $U^\dagger \vec{P} U$ is Pauli for every Pauli $\vec{P}$.
- Take $X^{\vec{z}_1 \oplus \vec{z}_2} := X^{(\vec{z}_1 \oplus \vec{z}_2)_1} \otimes \cdots \otimes X^{(\vec{z}_1 \oplus \vec{z}_2)_n}$. Then:

$$C_f^\dagger X^{\vec{z}_1 \oplus \vec{z}_2} C_f \left|\vec{z}_1, 0\right\rangle = e^{i\frac{\pi}{4}} C_f^\dagger X^{\vec{z}_1 \oplus \vec{z}_2} \left|\vec{z}_1, 0\right\rangle = e^{i\frac{\pi}{4}} C_f^\dagger \left|\vec{z}_2, 0\right\rangle = e^{i\frac{\pi}{4}} \left|\vec{z}_2, 0\right\rangle .$$

- A Pauli can't introduce a $e^{i\frac{\pi}{4}}$ phase, so C_f not Clifford.
- Hence, T-count of C_f is not zero.

The reduction

- Given f, construct C_f. Use T-COUNT oracle to determine whether minimal T-count of C_f is greater than 0.

The reduction

- Given f, construct C_f. Use T-COUNT oracle to determine whether minimal T-count of C_f is greater than 0.
- If it is, then C_f is non-Clifford, and hence f is satisfiable.

The reduction

- Given f, construct C_f. Use T-COUNT oracle to determine whether minimal T-count of C_f is greater than 0.
- If it is, then C_f is non-Clifford, and hence f is satisfiable.
- If it is not, then f is either *not* satisfiable or *everywhere* satisfiable:
  - Test $f(0 \cdots 0)$.
  - If $f(0 \cdots 0) = 0$, then it is not everywhere satisfiable, and hence must not be satisfiable.
  - Otherwise if $f(0 \cdots 0) = 1$, then we know that f is satisfiable.

The reduction

- Given f, construct C_f. Use T-COUNT oracle to determine whether minimal T-count of C_f is greater than 0.
- If it is, then C_f is non-Clifford, and hence f is satisfiable.
- If it is not, then f is either *not* satisfiable or *everywhere* satisfiable:
  - Test $f(0 \cdots 0)$.
  - If $f(0 \cdots 0) = 0$, then it is not everywhere satisfiable, and hence must not be satisfiable.
  - Otherwise if $f(0 \cdots 0) = 1$, then we know that f is satisfiable.

Theorem

T-COUNT is NP-hard under polynomial-time Turing reductions.

Upper bound

NQP = non-deterministic quantum poly-time.
complete problem: determine if two q circuits are *exactly* equal.

Upper bound

NQP = non-deterministic quantum poly-time.
complete problem: determine if two q circuits are *exactly* equal.

Theorem

T-COUNT is in $\mathrm{NP}^{\mathrm{NQP}}$.

Upper bound

NQP = non-deterministic quantum poly-time.
complete problem: determine if two q circuits are *exactly* equal.

Theorem

T-COUNT is in $\mathrm{NP}^{\mathrm{NQP}}$.

Proof: We are given poly-size circuit C and number k.
Decide: is there circuit C' with T-count $\leq k$ implementing same unitary?
Using NP, non-deterministically choose a C' with T-count $\leq k$,
then use NQP oracle to determine whether $C' = C$.
If any branch accepts, return true.

Upper bound

NQP = non-deterministic quantum poly-time.
complete problem: determine if two q circuits are *exactly* equal.

Theorem

T-COUNT is in $\mathrm{NP}^{\mathrm{NQP}}$.

Proof: We are given poly-size circuit C and number k.
Decide: is there circuit C' with T-count $\leq k$ implementing same unitary?
Using NP, non-deterministically choose a C' with T-count $\leq k$,
then use NQP oracle to determine whether $C' = C$.
If any branch accepts, return true.

Note: Boolean circuit minimalisation is complete for $\Sigma_2^P := \mathrm{NP}^{\mathrm{NP}}$.

Variations

We can reuse this argument to prove other hardness results.

CNOT-COUNT

We are given

- unitary Clifford+T circuit implementing a unitary U, and
- an integer k.

Decide whether there exists a unitary Clifford+T circuit that implements U which uses at most k CNOT gates.

CNOT-COUNT

We are given

- unitary Clifford+T circuit implementing a unitary U, and
- an integer k.

Decide whether there exists a unitary Clifford+T circuit that implements U which uses at most k CNOT gates.

Theorem

CNOT-COUNT is NP-hard under polynomial-time Turing reductions.

CNOT-COUNT

We are given

- unitary Clifford+T circuit implementing a unitary U, and
- an integer k.

Decide whether there exists a unitary Clifford+T circuit that implements U which uses at most k CNOT gates.

Theorem

CNOT-COUNT is NP-hard under polynomial-time Turing reductions.

Proof: analogous to above, because C_f has CNOT count 0 iff f is constant.

HAD-COUNT

We are given

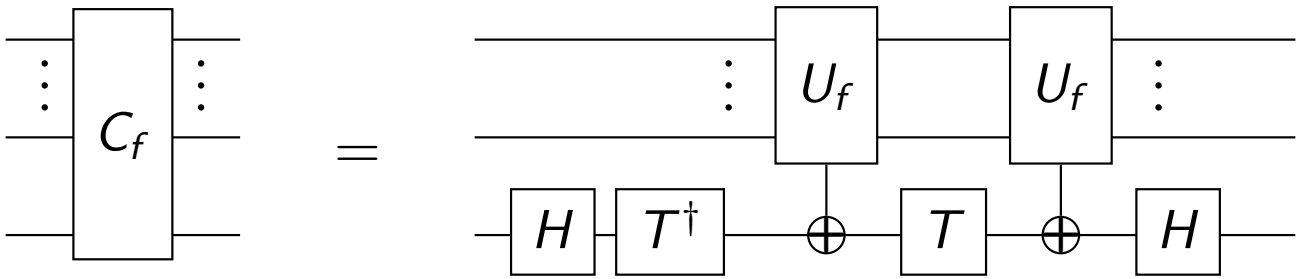
- unitary Clifford+T circuit implementing a unitary U, and
- an integer k.

Decide whether there exists a unitary Clifford+T circuit that implements U which uses at most k Hadamard gates.

Theorem

HAD-COUNT is NP-hard under polynomial-time Turing reductions.

Proof: analogous to above, but with slightly different circuit:



TOFFOLI-COUNT

We are given

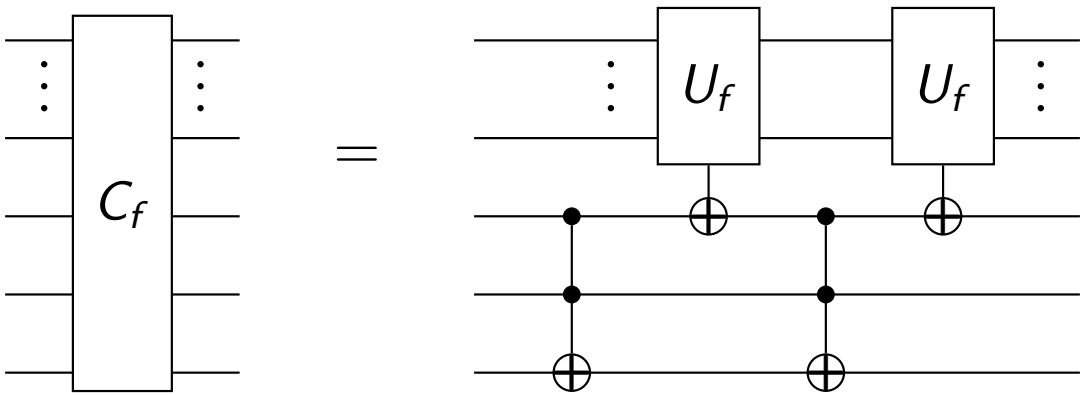
- unitary $\{\text{NOT}, \text{CNOT}, \text{Tof}\}$ circuit implementing a unitary U, and
- an integer k.

Decide whether there exists a unitary $\{\text{NOT}, \text{CNOT}, \text{Tof}\}$ circuit that implements U which uses at most k Toffoli gates.

Theorem

TOFFOLI-COUNT is NP-hard under polynomial-time Turing reductions.

Proof: analogous to other arguments but with the circuit:



Upper bounds

The $\mathrm{NP}^{\mathrm{NQP}}$ upper bound also applies to TOFFOLI-COUNT and HAD-COUNT, because there are only polynomial number of n-qubit circuits

- over $\{\mathrm{Tof}, \mathrm{CNOT}, \mathrm{NOT}\}$ that are Toffoli-free,
- over Clifford$+T$ that are Hadamard-free.

Upper bounds

The $\mathrm{NP}^{\mathrm{NQP}}$ upper bound also applies to TOFFOLI-COUNT and HAD-COUNT, because there are only polynomial number of n-qubit circuits

- over $\{\mathrm{Tof}, \mathrm{CNOT}, \mathrm{NOT}\}$ that are Toffoli-free,
- over Clifford+T that are Hadamard-free.

But, this does not apply to CNOT-COUNT:
there's an infinite amount of 1-qubit CNOT-free Clifford+T circuits.

Generic non-Clifford count

Fix non-Clifford gate G and error bound $\varepsilon > 0$.

Generic non-Clifford count

Fix non-Clifford gate G and error bound $\varepsilon > 0$.
The problem G-COUNT$_\varepsilon$ is: given

- A Clifford+G circuit C,
- an integer k,

determine if there exists Clifford+G circuit C' using at most k G gates such that

$$\exists \alpha : \| C - e^{i\alpha} C' \|_\infty \leq \varepsilon$$

Generic non-Clifford count

Fix non-Clifford gate G and error bound $\varepsilon > 0$.
The problem G-COUNT$_\varepsilon$ is: given

- A Clifford+G circuit C,
- an integer k,

determine if there exists Clifford+G circuit C' using at most k G gates such that

$$\exists \alpha : \|C - e^{i\alpha} C'\|_\infty \leq \varepsilon$$

Theorem

If $\varepsilon < \sin(\frac{\pi}{16}) \approx 0.195$, then G-COUNT$_\varepsilon$ is NP-hard under polynomial-time Turing reductions.

Generic non-Clifford count

Fix non-Clifford gate G and error bound $\varepsilon > 0$.
The problem G-COUNT$_\varepsilon$ is: given

- A Clifford+G circuit C,
- an integer k,

determine if there exists Clifford+G circuit C' using at most k G gates such that

$$\exists \alpha : \|C - e^{i\alpha} C'\|_\infty \leq \varepsilon$$

Theorem

If $\varepsilon < \sin(\frac{\pi}{16}) \approx 0.195$, then G-COUNT$_\varepsilon$ is NP-hard under polynomial-time Turing reductions.

Proof idea: If C_f is non-Clifford, argue any Clifford+G approximation of C_f must also contain at least one G gate, by giving a minimal distance between C_f and any Clifford.

Conclusion

The following circuit optimisation problems are NP-hard:

- Optimising T-count of Clifford+T circuits,
- Optimising CNOT-count of Clifford+T circuits,
- Optimising Hadamard-count of Clifford+T circuits,
- Optimising Toffoli-count of classical reversible circuits,
- Approximate G-count optimisation for *any* non-Clifford G.

Conclusion

The following circuit optimisation problems are NP-hard:

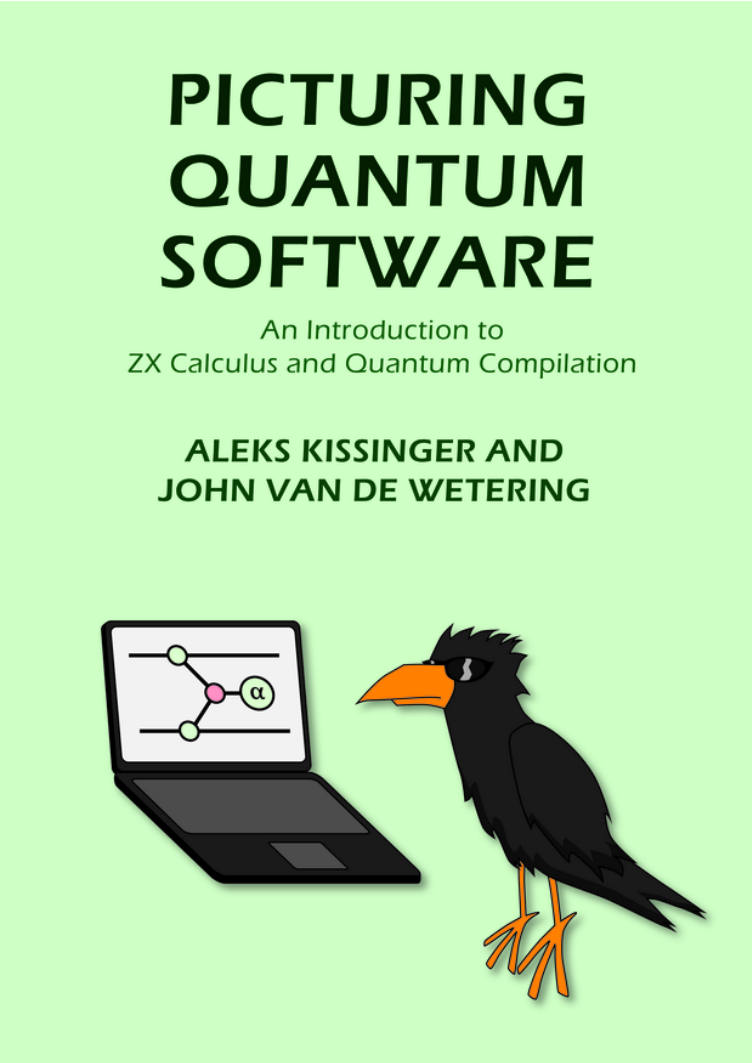
- Optimising T-count of Clifford+T circuits,
- Optimising CNOT-count of Clifford+T circuits,
- Optimising Hadamard-count of Clifford+T circuits,
- Optimising Toffoli-count of classical reversible circuits,
- Approximate G-count optimisation for *any* non-Clifford G.

In addition: we only needed to distinguish gate-count **zero** from gate-count **non-zero**, hence any modified cost function, like T-depth, will still result in an NP-hard problem.

Open questions

- Exact hardness of these problems? $\mathrm{NP}^{\mathrm{NQP}}$?
- Upper bound on CNOT-COUNT and G-COUNT$_\varepsilon$?
- Hardness of *exact* Clifford+G optimisation?
- Hardness of approximating the ideal gate count?
- Hardness of CNOT-COUNT on CNOT-circuits?
- Hardness of T-count in Hadamard-free Clifford+T circuits?

New book on quantum compilation!

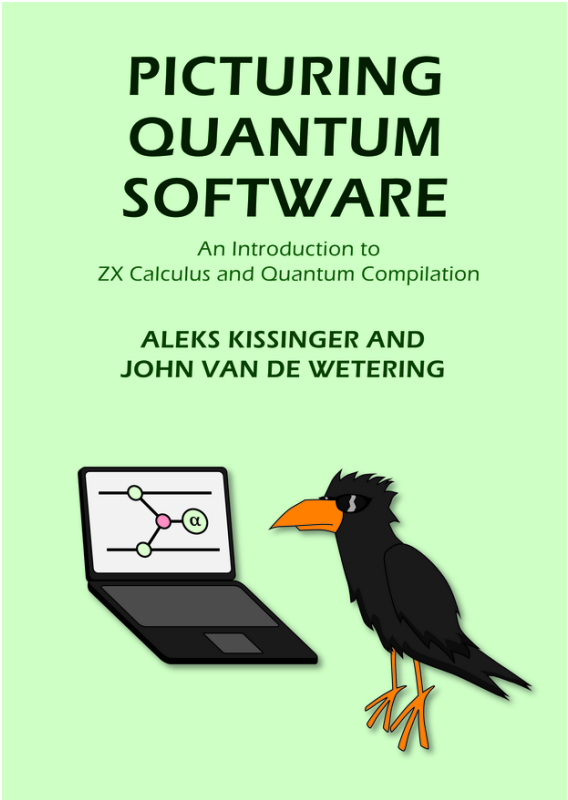


- Over 500 pages and 100 exercises!
- Synthesis of CNOT, Clifford, and Clifford+T circuits!
- Classical oracles!
- Measurement-based QC!
- Clifford+T synthesis, optimisation and catalysis!
- A new approach to understanding quantum error correction!
- And all this using ZX-diagrams!

`https://github.com/zxcalc/book`

Thank you for your attention!

vdW & Amy 2023, arXiv:2310.05958
Optimising quantum circuits is generally hard



`https://github.com/zxcalc/book`