SAND2016-11153PE

Exceptional service in the national interest

Sandia National Laboratories

Quantum Error Correcting Code Simulation

Ciarán Ryan-Anderson

Work with Andrew Landahl, Tzvetan Metodi

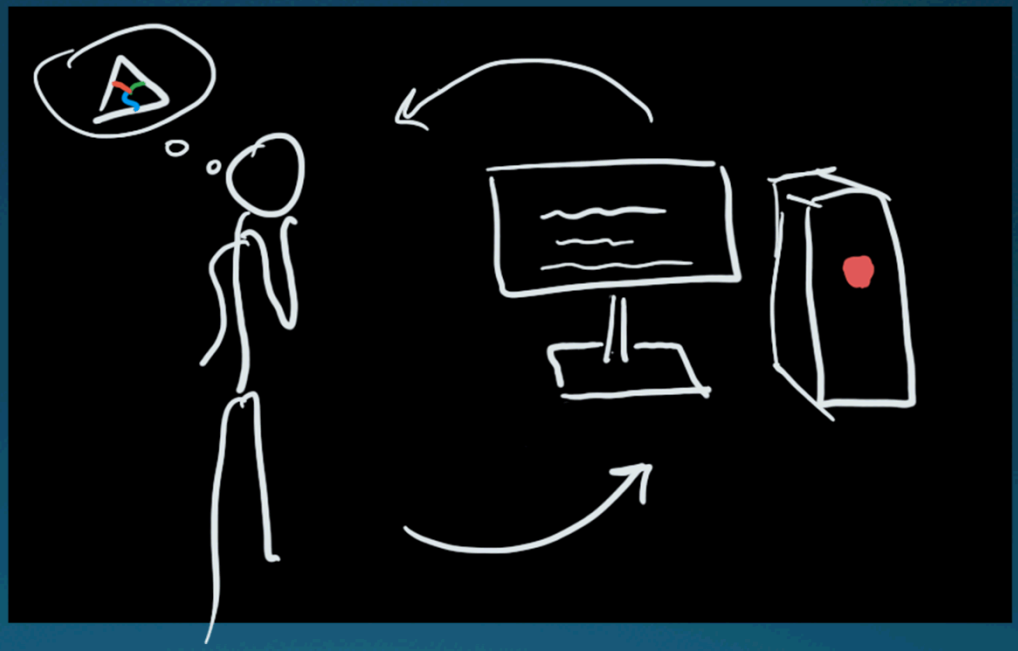
September 21, 2016

U.S. DEPARTMENT OF ENERGY | NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

A Program for QECCs

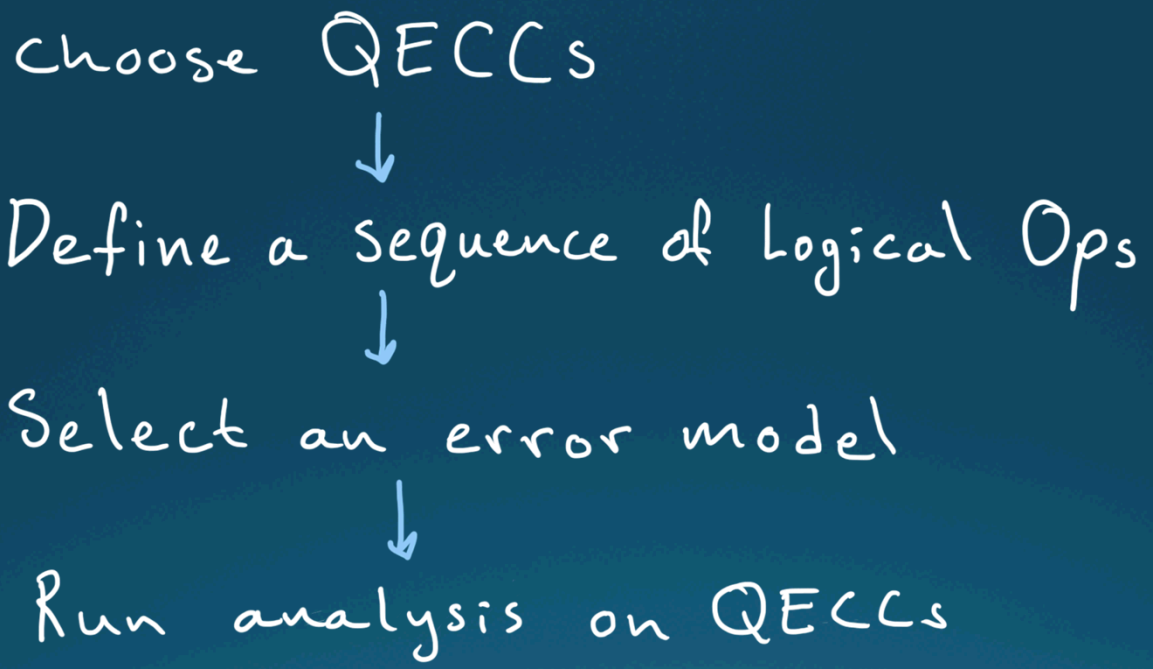
Performance
Estimator for
Codes
On
Surfaces



Focus

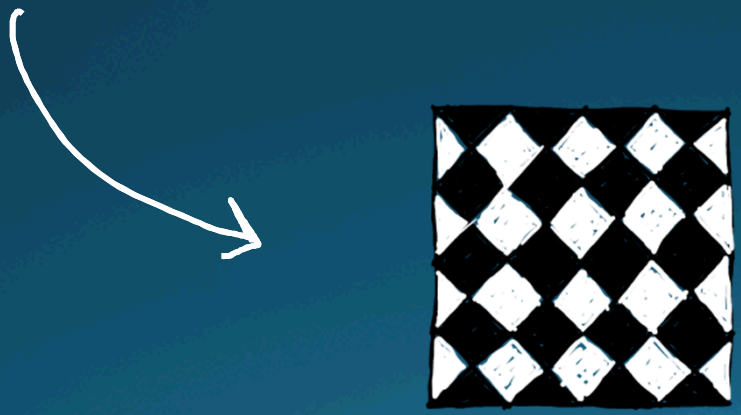
- Show the PECOS interface
- Present some concepts in QECC

Modular Steps



Choosing QECCs

```
import pecos as pc
Surface = pc.qecc.Surface4444(distance=5)
```



4.4.4.4 surface code

Stabilizers

$$S_i |\psi\rangle = +1 |\psi\rangle$$

4.4.4.4

$$2^n \rightarrow 2^{n-\#s} \rightarrow 2^1$$

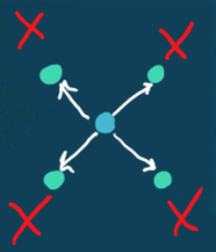
$$\prod_i S_i |\bar{\psi}\rangle = +1 |\bar{\psi}\rangle$$

Stabilizers

$$S_i |\psi\rangle = +1 |\psi\rangle$$

4.4.4.4

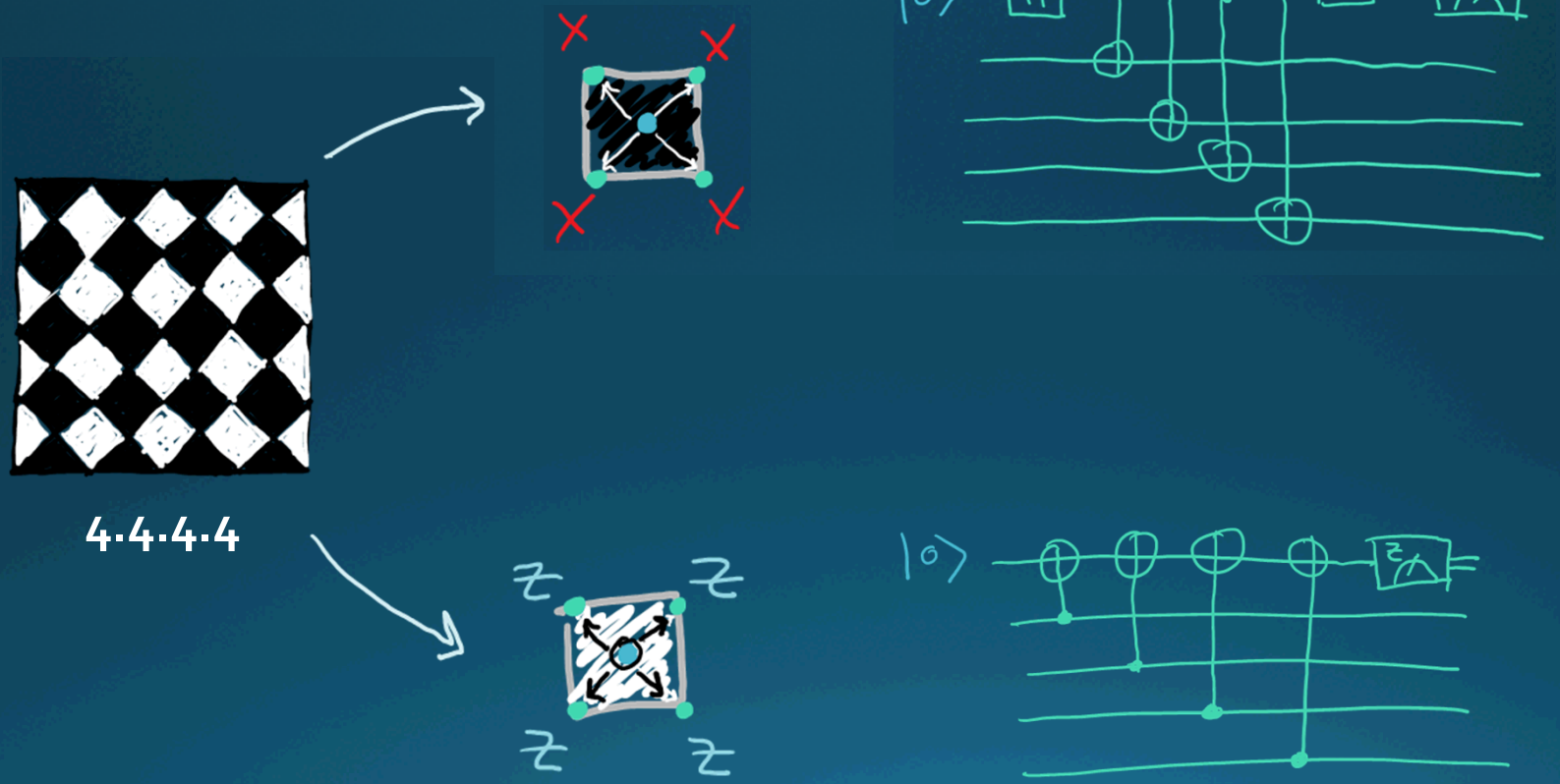
Stabilizers



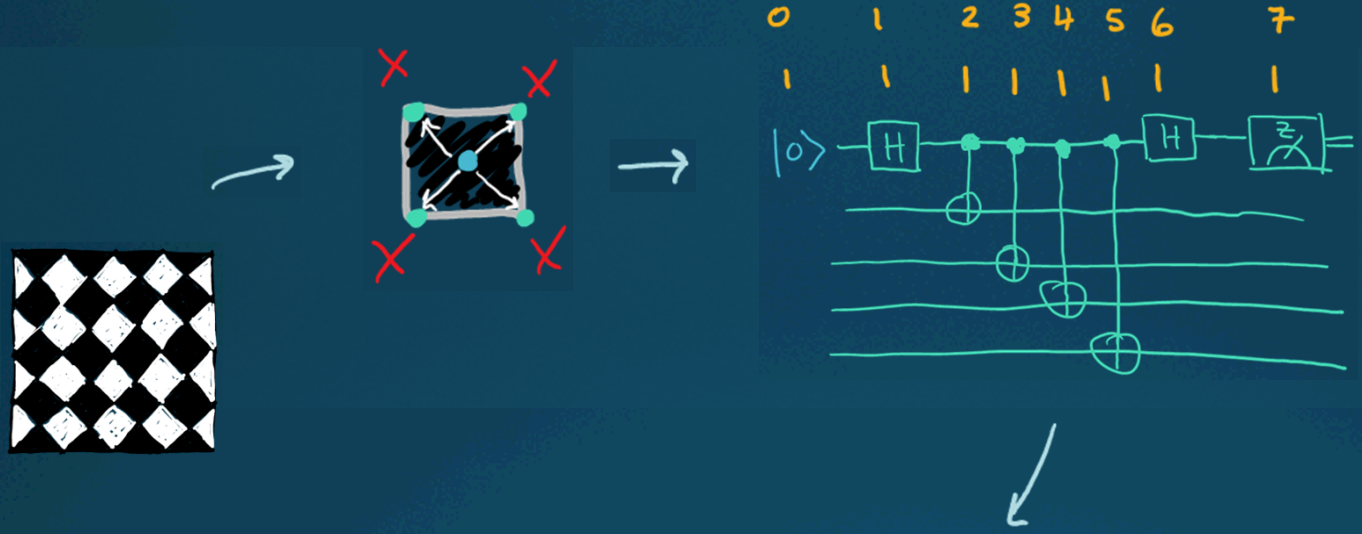
$$\begin{cases} |++++\rangle \longrightarrow 0 \\ |++-+\rangle \longrightarrow 1 \end{cases}$$

$$\begin{cases} |0101\rangle \longrightarrow 0 \\ |0111\rangle \longrightarrow 1 \end{cases}$$

Checks to Circuits



Checks to Circuits

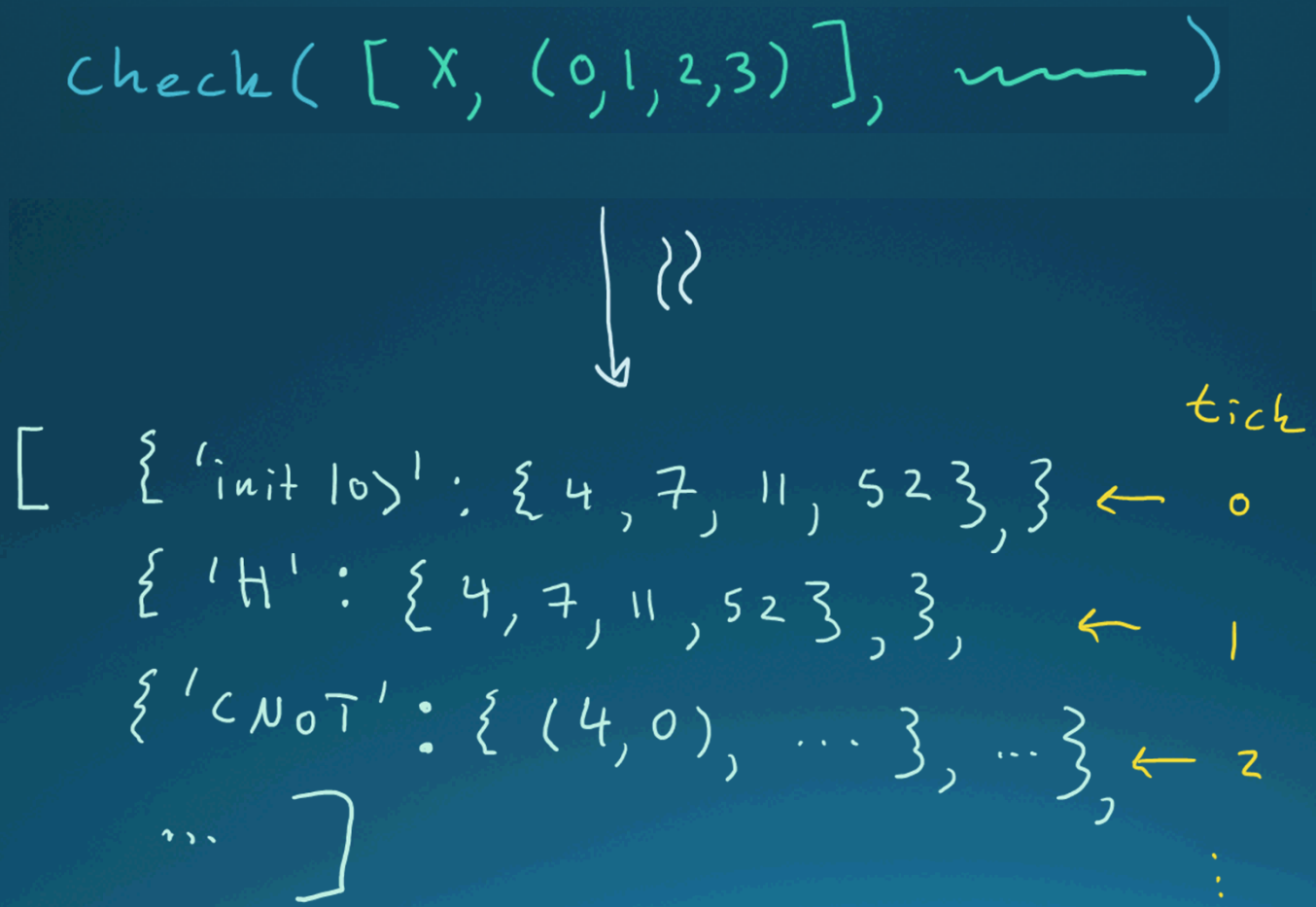


timing info

Check([(X,0), (X,1), ...], ...)

Check([X, (0,1,2,3)], ...)

Circuits

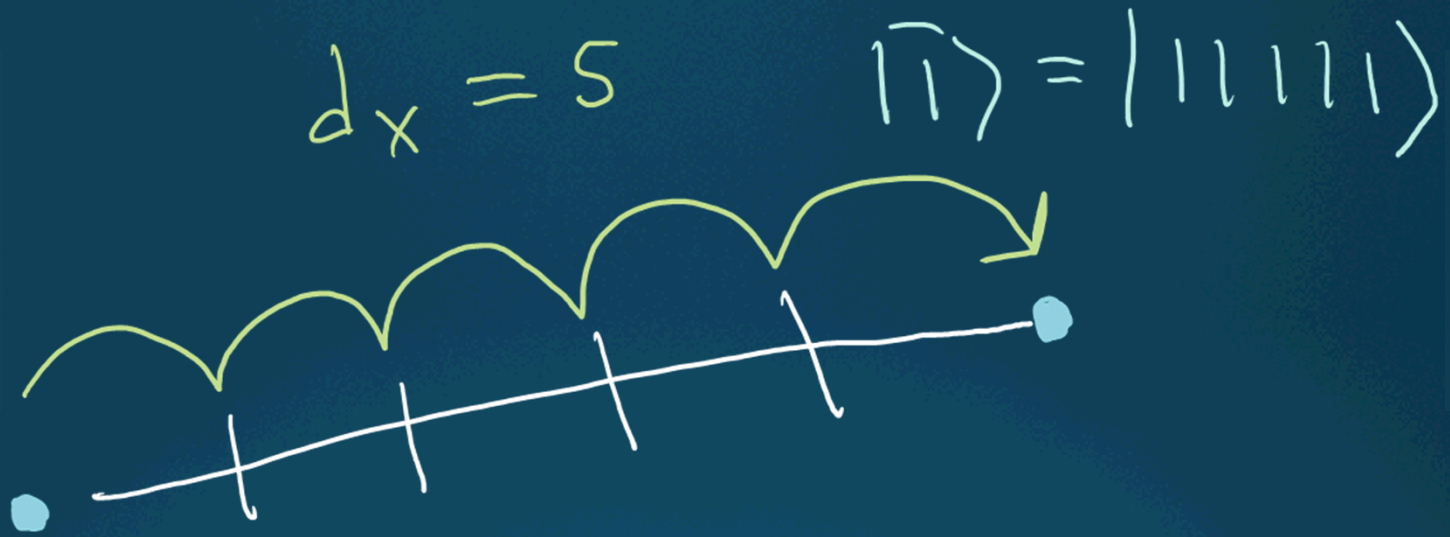


Checks to Circuits

```
Surface = pc.qecc.Surface4444(distance=5)
```

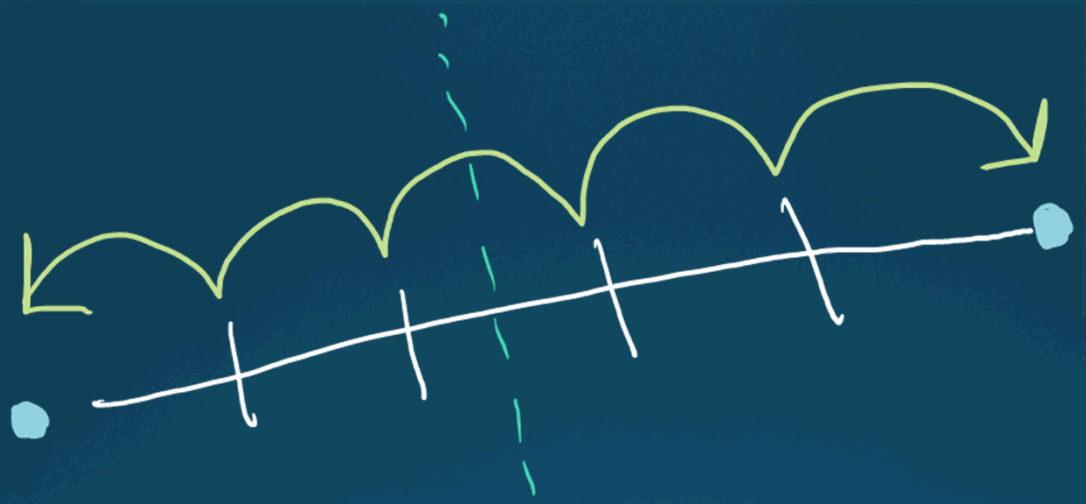
```
# Intialize an instance of a QECC
surface = pc.qecc.Surface4444(distance=5)
```

Distance



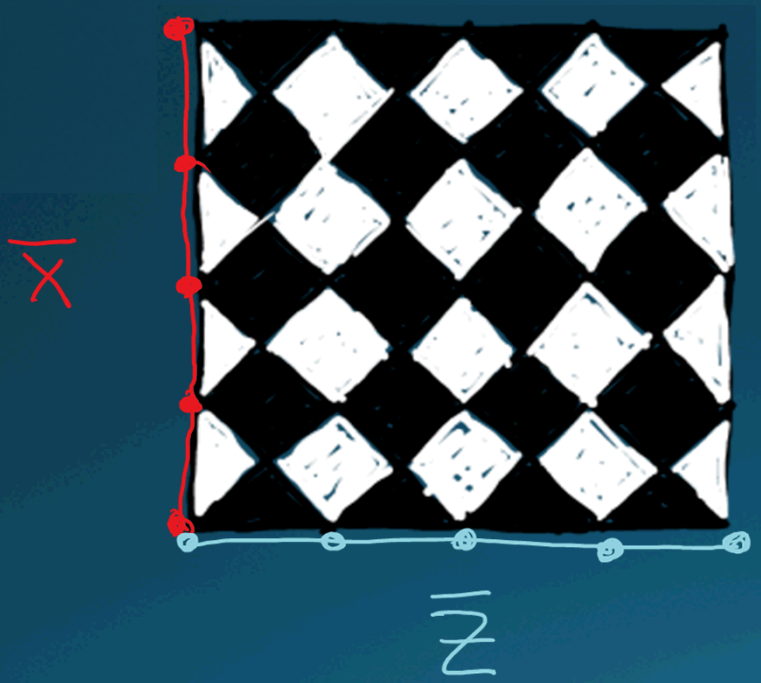
$|0\rangle = |00000\rangle$

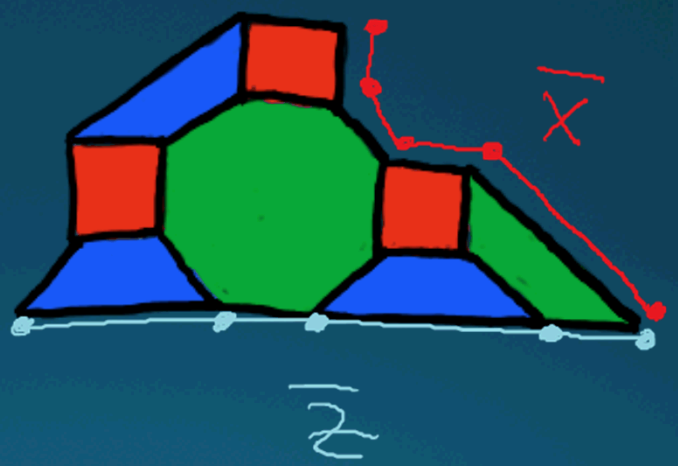
Distance



$$t = \left\lfloor \frac{d-1}{2} \right\rfloor$$

Distance





Logic

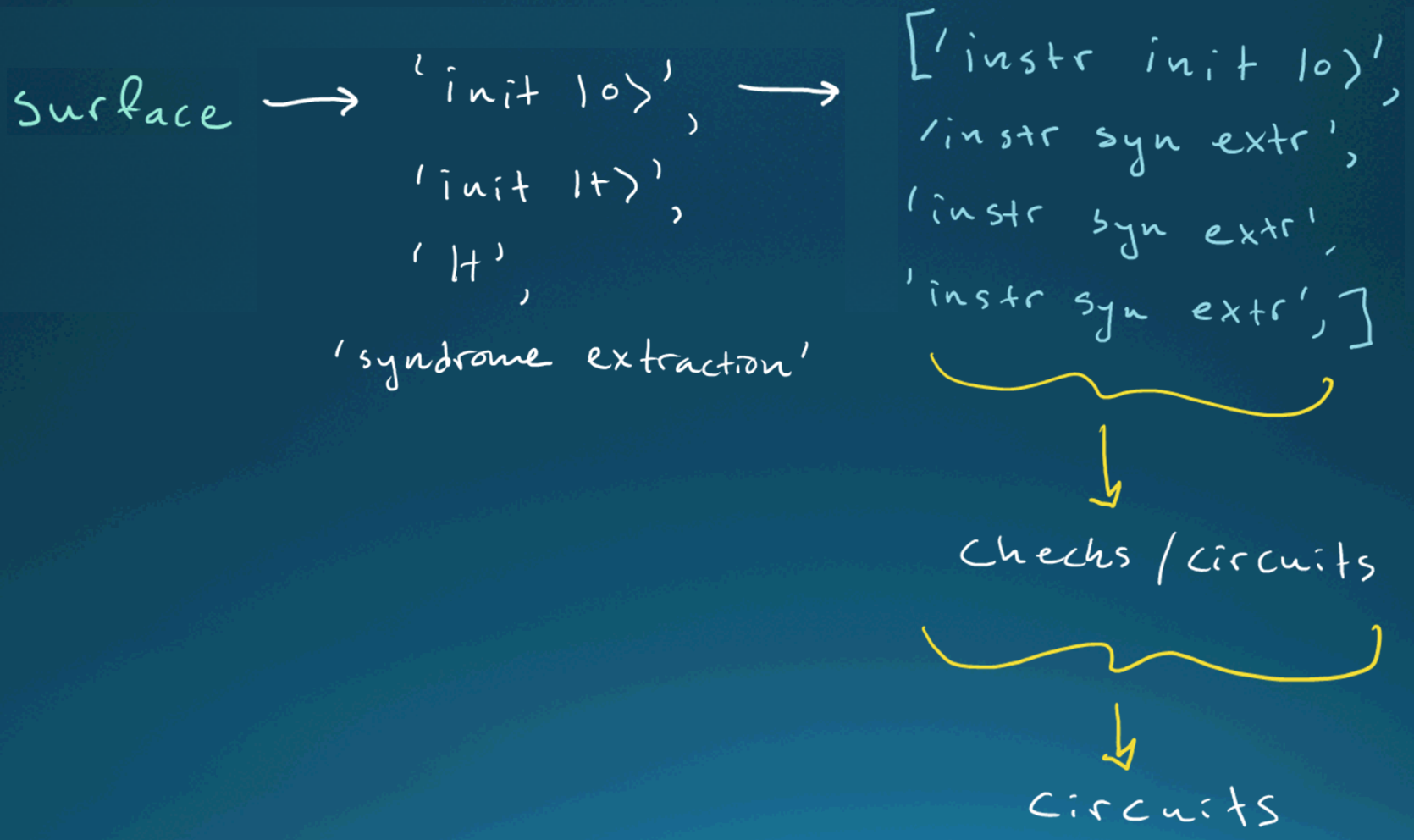
```
surface_layout = surface.layout
L = pc.Circuit.Logic(surface_layout)
```

Logical Gates

L.gate(surface, 'ideal init |psi>')

L.gate(surface, 'syndrome extraction')

Logical gates vs logical instructions



Logic

```
surface = pc.qecc.Surface4444(distance=5)

L = pc.circuit.Logic(layout=surface.layout)
L.gate(surface_code, 'ideal init |psi>')
L.gate(surface_code, 'syndrome_extraction')
```

Error generator/model

```
depolar = pc.error_gen.GateWise('symmetric_depolarizing')
```

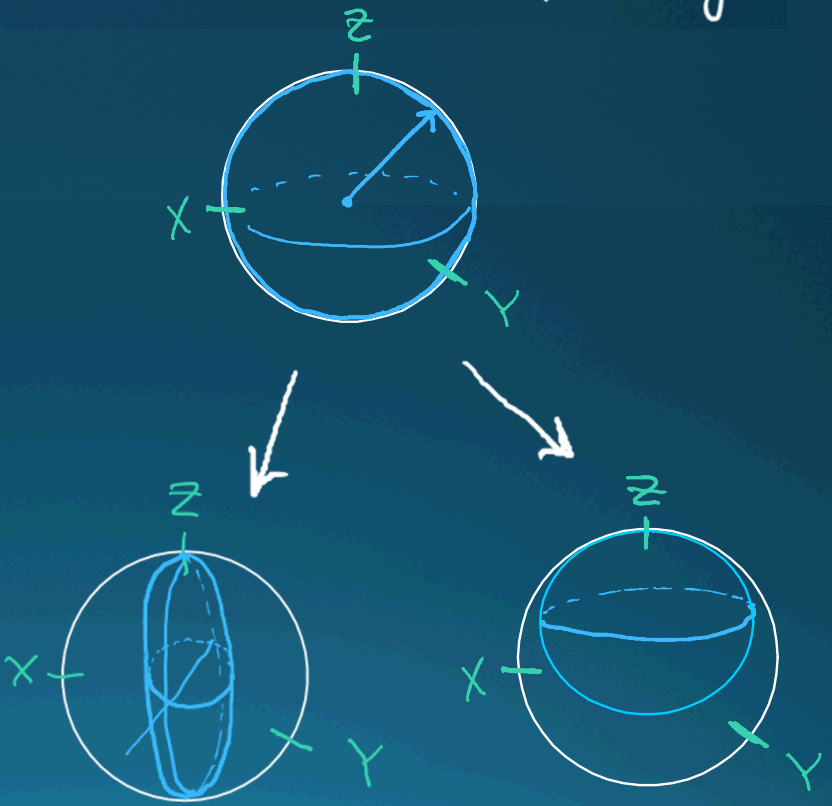
Per gate:

- insert error before
- insert error after
- replace the gate

Error generator/model

```
depolar = pc.error_gen.GateWise('symmetric_depolarizing')
```

- Paulis
- Cliffords
- Measurements
- Leakage
- much more



Error generator/model

```
surface = pc.qecc.Surface4444(distance=5)

L = pc.circuit.Logic(layout=surface.layout)
L.gate(surface_code, 'ideal init |psi>')
L.gate(surface_code, 'syndrome_extraction')

depolar = pc.error_gen.Gatewise(error_model='symmetric depolarizing')
```

Simulation model

```
sim_model = pc.sim_model.StandardCodeCapacity(logic = L,
                                              error_gen = depolar)
```

- initiates all qudits to $|0\rangle$
- couples the circuits in logic and generates errors

Stabilizer Simulation

- Realistic codes and error models →

- Normally much faster ($n^{0.7}$ vs n^2)

Levels of Modeling

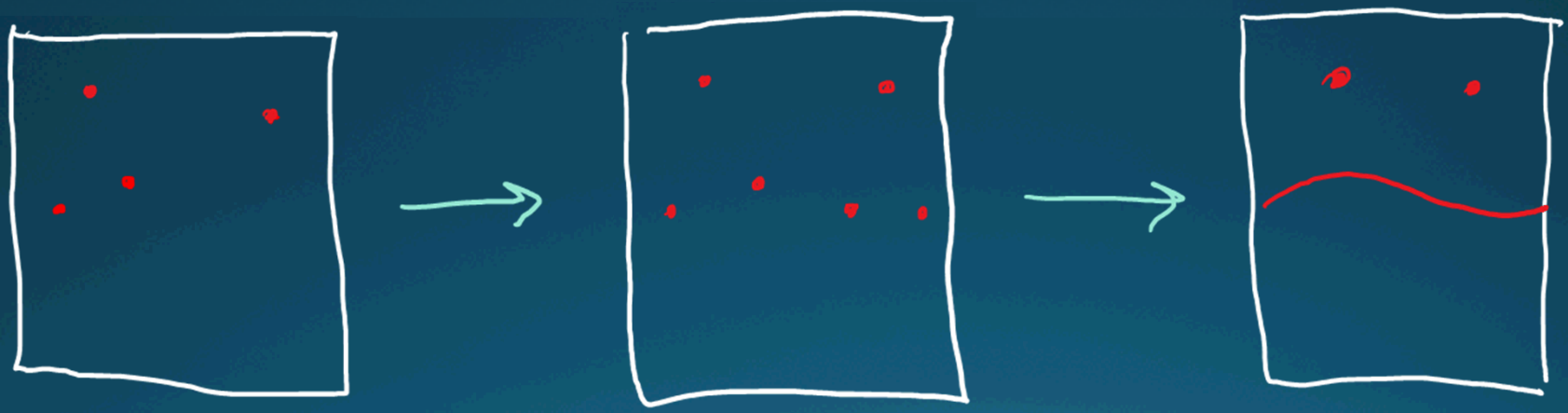
- Code Capacity
  - Error on data qubits
- Phenomenological
  - Error on each data qubits and measurement
- Circuit
  - Error on each unitary and measurement

Simulation Model

```
surface = pc.qecc.Surface4444(distance=5)

L = pc.circuit.Logic(layout=surface.layout)
L.gate(surface_code, 'ideal init |psi>')
L.gate(surface_code, 'syndrome_extraction')

depolar = pc.error_gen.Gatewise(error_model='symmetric depolarizing')

sim_model = pc.sim_model.StandardCodeCapacity(logic=L,
                                              error_generator=depolar)
```

Decoder

```
diamonds = pc.decoder.ExpandingDiamonds(logic=L)
```



Decoder

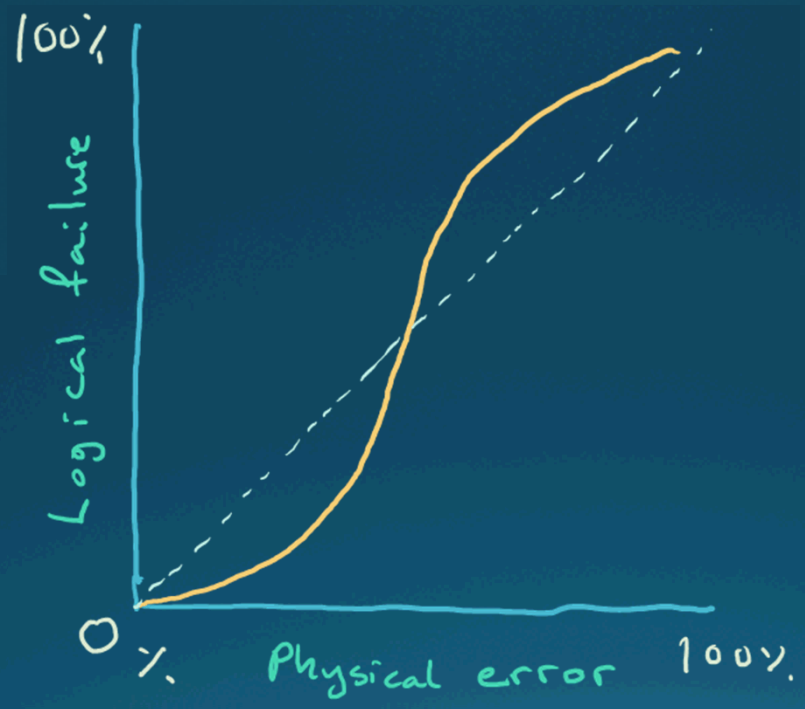
```
surface = pc.qecc.Surface4444(distance=5)

L = pc.circuit.Logic(layout=surface.layout)
L.gate(surface_code, 'ideal init |psi>')
L.gate(surface_code, 'syndrome_extraction')

depolar = pc.error_gen.Gatewise(error_model='symmetric depolarizing')

sim_model = pc.sim_model.StandardCodeCapacity(logic=L,
                                              error_generator=depolar)

diamonds = pc.decoder.ExpandingDiamonds(logic=L)
```

Analysis Tool

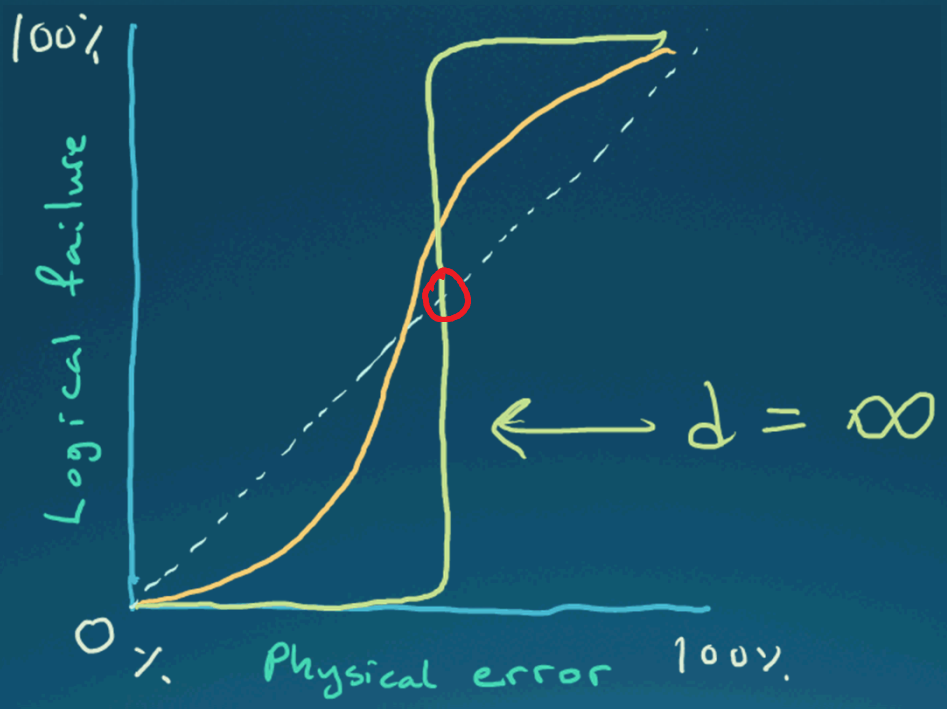
```
pc.tool.monte_carlo.Default(sim_model,
                            error_params=mm,
                            decoder=diamonds)
```

- Run the sim_model many times.
- Determine the rate of logical error to physical error.

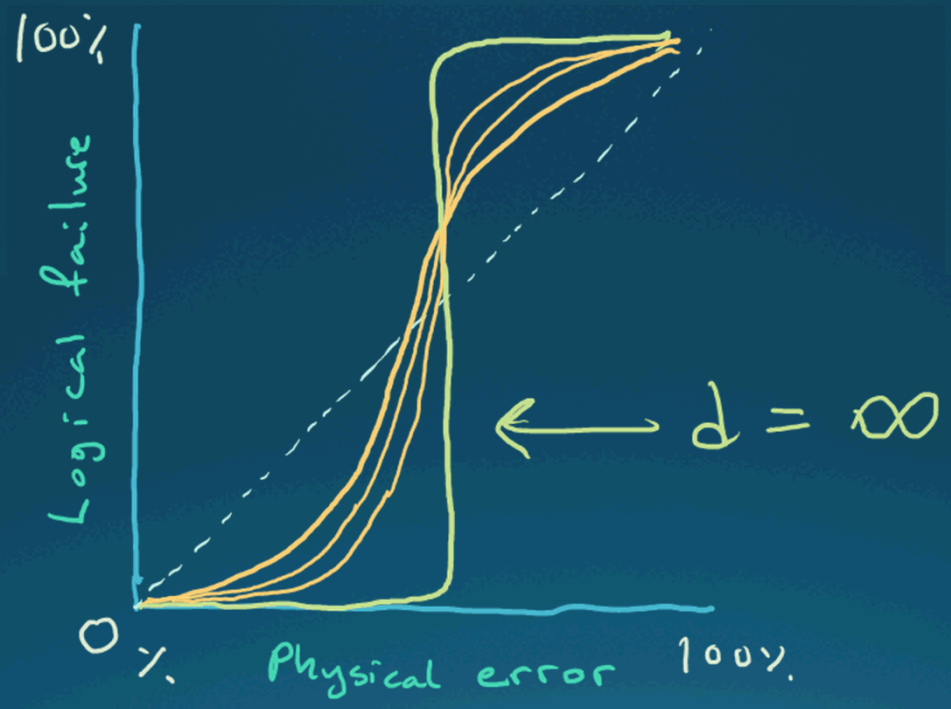
Logical Error Rate



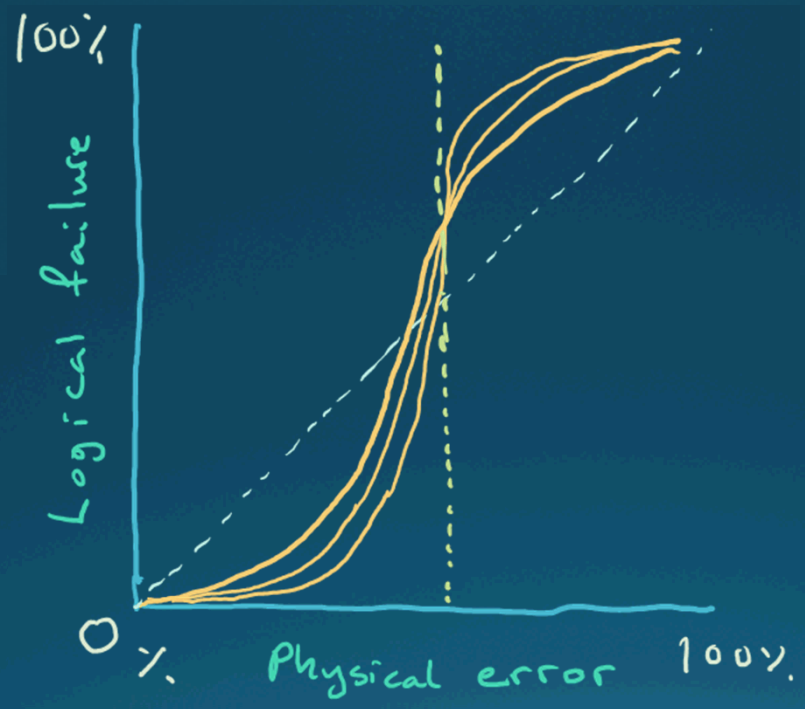
Logical Error Rate



Logical Error Rate



Logical Error Rate



Analysis Tool

```
surface = pc.qecc.Surface4444(distance=5)

L = pc.circuit.Logic(layout=surface.layout)
L.gate(surface_code, 'ideal init |psi>')
L.gate(surface_code, 'syndrome_extraction')

depolar = pc.error_gen.Gatewise(error_model='symmetric depolarizing')

sim_model = pc.sim_model.StandardCodeCapacity(logic=L,
                                              error_generator=depolar)

diamonds = pc.decoder.ExpandingDiamonds(logic=L)

monte = pc.tool.monte_carlo.Default(sim_model=sim_model,
                                    decoder=diamonds)
```

The interface

code →

layout →

Logical gates

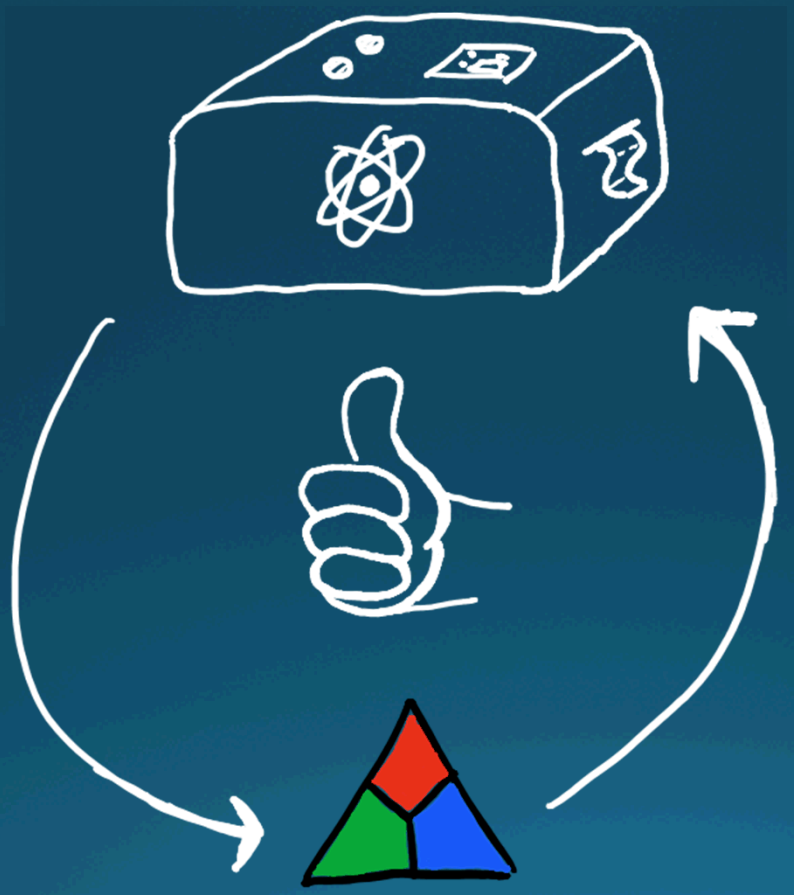
error →

How to model →

Find out something

```
import pecos as pc
Surface = pc.qecc.Surface4444(distance=5)
surface_layout = surface.layout
L = pc.circuit.Logic(surface_layout)
L.gate(surface, 'ideal init |psi>')
L.gate(surface, 'syndrome extraction')
depolar = pc.error_gen.GateWise('symmetric_depolarizing')
sim_model = pc.sim_model.StandardCodeCapacity(logic=L,
                                              error_gen=depolar)
pc.tool.monte_carlo.Default(sim_model,
                            error_params=...,
                            decoder=diamonds)
```

Theory and experiment



Future additions

- Add gate duration
- A few more tweaks to efficiency
- Qudit stabilizer simulation
- Add the decoders being worked on
- Machine learning decoder

Sandia National Laboratories

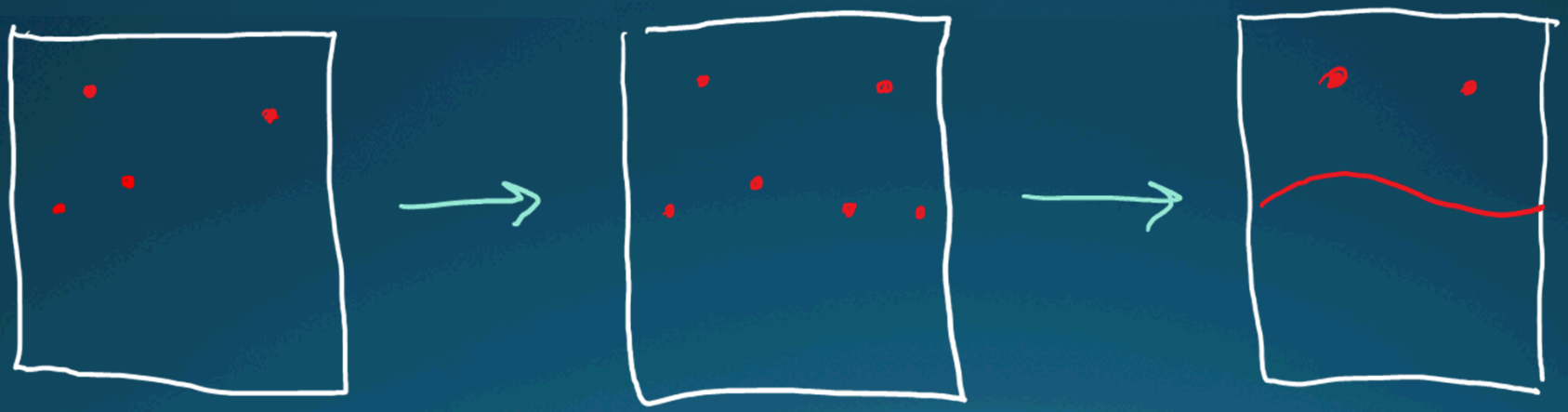
Thanks!

Sandia National Laboratories

Extras

Decoder

```
diamonds = pc.decoder.ExpandingDiamonds(logic=L)
```



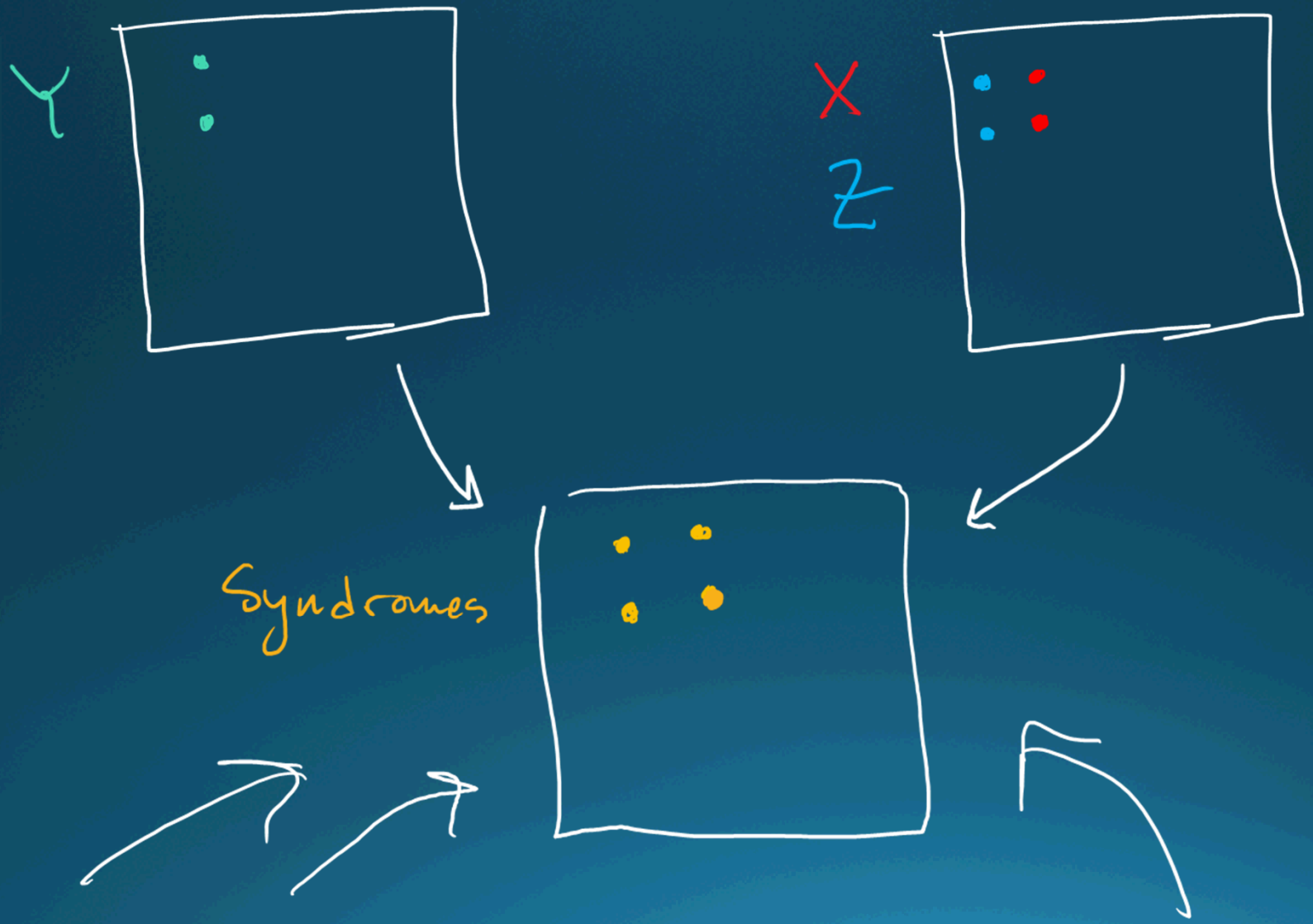
Decoder

```
diamonds = pc.decoder.ExpandingDiamonds(logic=L)
```

Decoder

```
diamonds = pc.decoder.ExpandingDiamonds(logic=L)
```

Decoder



Error Model

dictionary for replace, before, after

{'H': { set_of_errors: 0.5, ... },

'CNOT': { ... }, ... }