

Quantum Computing for Particle Physics: Simulation and Machine Learning

Callum Duffy

UCL · IOP Joint APP and HEPP Conference · 2026

Motivation: Quantum Simulation

Strongly correlated quantum systems (e.g. Fermi-Hubbard model, Z2 lattice Gauge theory) are exponentially hard to simulate classically.

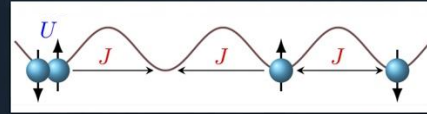
Quantum computers offer a natural platform since the Hilbert space is native.

Gate-based simulation: decompose time-evolution operator into sequences of elementary gates.

Key challenge: Noisy quantum computers mean limited coherence times restrict the depth of circuits we can run.

Goal: maximise physical simulation time within hardware coherence budget.

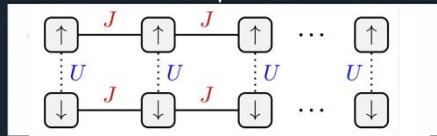
1D FERMION-HUBBARD MODEL



Fermion Hamiltonian

$$H = -J \sum_{j=1}^{L-1} \sum_{\sigma \in \{\uparrow, \downarrow\}} (c_{j,\sigma}^\dagger c_{j+1,\sigma} + \text{h.c.}) + U \sum_j n_{j,\uparrow} n_{j,\downarrow}$$

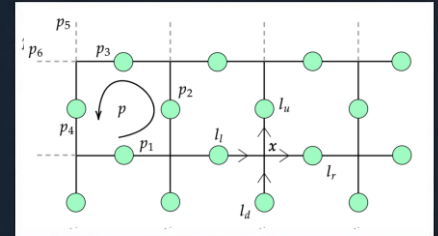
Jordan-Wigner Mapping:
Fermions \rightarrow Qubits



Qubit Hamiltonian (Jordan-Wigner)

$$\tilde{H} = -\frac{t}{2} \sum_{j=0}^{L-2} (X_{2j} X_{2j+2} + Y_{2j} Y_{2j+2} + X_{2j+1} X_{2j+3} + Y_{2j+1} Y_{2j+3}) + \frac{U}{4} \sum_{j=0}^{L-1} (I_{2j} I_{2j+1} + Z_{2j} Z_{2j+1} - Z_{2j} - Z_{2j+1})$$

Z2 LATTICE GAUGE THEORY



Kogut-Susskind Hamiltonian

$$H_{KS} = H_E + H_B = \lambda_E \sum_l [2 - (P_l + P_l^\dagger)] + \lambda_B \sum_n [2 - (U_{p1} U_{p2} U_{p3}^\dagger U_{p4}^\dagger + \text{H.c.})].$$

Electric field term H_E (links),
Magnetic field term H_B (plaquettes)

Pauli Operator Hamiltonian

$$H_{KS} = H_E + H_B = -2\lambda_E \sum_l Z_l - 2\lambda_B \sum_p X_{p1} X_{p2} X_{p3} X_{p4},$$

Background: Trotterised Evolution

Simulating time evolution e^{-iHt}

The full Hamiltonian H often contains non-commuting terms

Trotter-Suzuki: split evolution into small timesteps δt

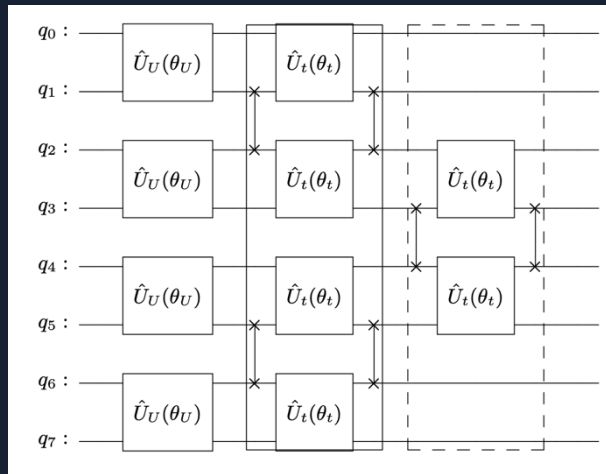
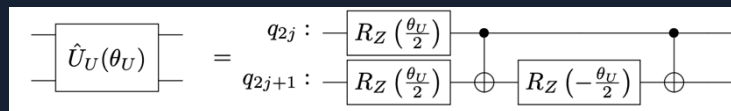
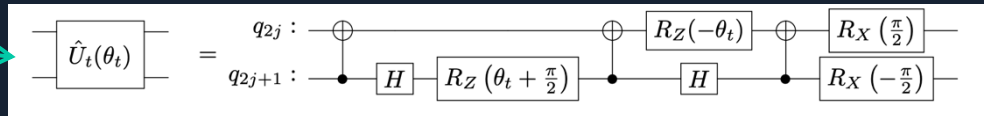
First order: $e^{-iH\delta t} \approx \prod_k e^{-iH_k\delta t}$

This introduces errors but allows us to simulate the theory with the standard gate set.

Total simulation time $T = N \times \delta t$, therefore, deeper circuits incur more gate errors

1D Fermi Hubbard Trotter Circuit

$$\tilde{H} = -\frac{t}{2} \sum_{j=0}^{L-2} (X_{2j}X_{2j+2} + Y_{2j}Y_{2j+2} + X_{2j+1}X_{2j+3} + Y_{2j+1}Y_{2j+3}) + \frac{U}{4} \sum_{j=0}^{L-1} (I_{2j}I_{2j+1} + Z_{2j}Z_{2j+1} - Z_{2j} - Z_{2j+1})$$



Background: Trotterised Evolution

Simulating time evolution e^{-iHt}

The full Hamiltonian H often contains non-commuting terms

Trotter-Suzuki: split evolution into small timesteps δt

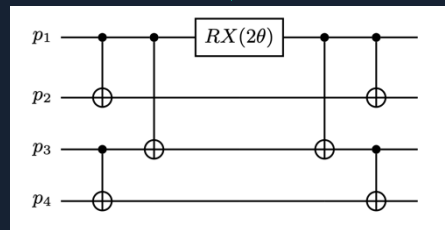
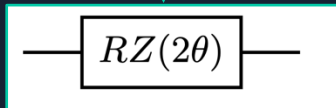
First order: $e^{-iH\delta t} \approx \prod_k e^{-iH_k\delta t}$

This introduces errors but allows us to simulate the theory with the standard gate set.

Total simulation time $T = N \times \delta t$, therefore, deeper circuits incur more gate errors

Z2 Lattice Gauge Theory

$$H_{KS} = H_E + H_B$$
$$= -2\lambda_E \sum_l Z_l - 2\lambda_B \sum_p X_{p1} X_{p2} X_{p3} X_{p4}$$



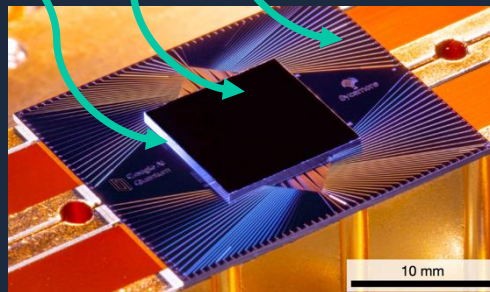
Hardware Platform: Transmon Qubits

General Hardware Hamiltonian

$$H(t) = H_0 + \sum u_j(t) H_j$$

H_0 : Drive Hamiltonian

H_j : Control Hamiltonian



The Transmon Hamiltonian

$$H_D = \underbrace{\sum_i \omega_i a_i^\dagger a_i}_{|0\rangle \rightarrow |1\rangle} - \underbrace{\sum_i \frac{\delta_i}{2} a_i^\dagger a_i^\dagger a_i a_i}_{\text{Separate higher-order states}} + \underbrace{\sum_{ij} g_{ij} a_i^\dagger a_j}_{\text{Qubit architecture}}$$

$|0\rangle \rightarrow |1\rangle$

Separate
higher-
order states

Qubit
architecture

$$H_c = \sum_i \Omega_i(t) (e^{i\nu t} a_i + e^{-i\nu t} a_i^\dagger)$$

$\Omega(t)$: Pulse amplitude
 ν : Phase

Background: Quantum Optimal Control & GRAPE

Goal: find a time-dependent control field $u(t)$ that steers the system to a target unitary V

Discretise Divide time T into N piecewise-constant pulses $\{u_k\}$

Propagate Compute $U_k = \exp(-iH(u_k)\delta t)$, $U = U_N \dots U_1$

Infidelity Minimise $I = 1 - |\text{Tr}(U^\dagger V)|/N$

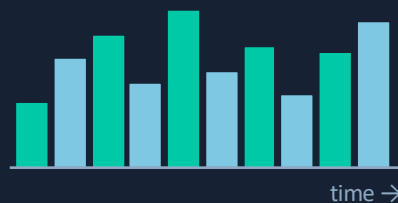
Gradient Compute $\frac{\partial I}{\partial u_k}$ (GRAPE gradient)

Update Gradient descent: $u_k \leftarrow u_k - \alpha \frac{\partial I}{\partial u_k}$

Iterate Repeat until convergence

GRAPE Pulse Profile

$u(t)$ Piecewise-constant control amplitudes



Input: $V, \Phi, R_{\mathcal{H}}, \varepsilon, \lambda, M$

Output: Φ

Define evolution: $U_G(\Phi) = U(\Phi_K) \dots U(\Phi_1)$

Define an infidelity function: $I(\Phi, V) = 1 - \text{Tr}\{U_G(\Phi)^\dagger V\}/N$

Initialise Adam moments per layer: $\mu_{l,k} = 0, \mathbf{v}_{l,k} = 0$

$\beta_1 \leftarrow 0.9$

$\beta_2 \leftarrow 0.999$

for $m \in \{1, 2, \dots, M\}$ **do**

 Compute the gradient at Φ :

$\mathbf{g}_{l,k} \leftarrow \partial_{x_{l,k}} I(\Phi, V)$

 Calculate Adam moments:

$\mu_{l,k} \leftarrow \beta_1 \cdot \mu_{l,k} + (1 - \beta_1) \cdot \mathbf{g}_{l,k}$

$\mathbf{v}_{l,k} \leftarrow \beta_2 \cdot \mathbf{v}_{l,k} + (1 - \beta_2) \cdot \mathbf{g}_{l,k}^2$

$\hat{\mu}_{l,k} \leftarrow \mu_{l,k} / (1 - \beta_1^m)$

$\hat{\mathbf{v}}_{l,k} \leftarrow \mathbf{v}_{l,k} / (1 - \beta_2^m)$

 Update parameters:

$\Phi \leftarrow \Phi - \lambda \cdot \hat{\mu} / (\sqrt{\hat{\mathbf{v}}} + \text{eps})$

if $I(\Phi, V) < \varepsilon$ **then**

 | break

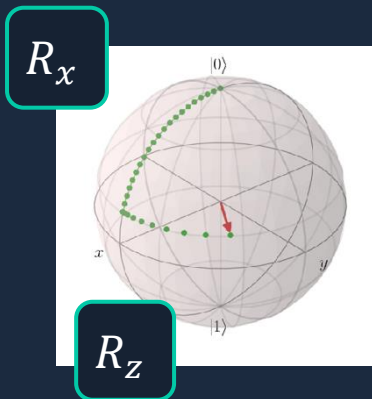
end

end

The Core Idea: Direct Pulse Synthesis

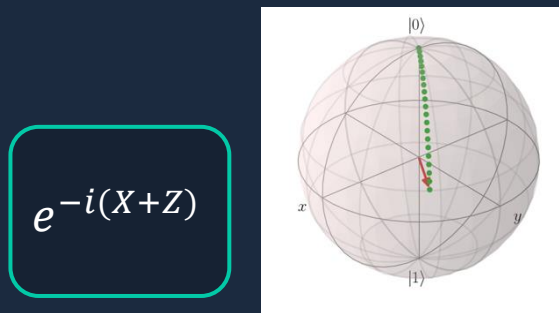
Why decompose into standard gates if we can synthesise the operation directly?

Using standard set of gates



Longer circuit \rightarrow more errors \rightarrow shorter simulation

Directly implement Operation



Shorter circuit \rightarrow fewer errors \rightarrow longer simulation

Our Approach: Pulse-Synthesised Trotter Steps

1 Target Unitary

Identify the Trotter step unitary

$U_{Trotter}$



2 GRAPE Optimisation

Run GRAPE on the Transmon Hamiltonian to find pulse $u^*(t)$ that implements $U_{Trotter}$



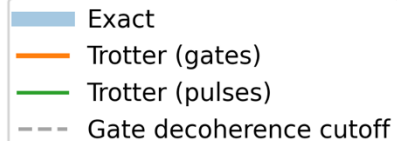
3 Pulse Execution

Execute $u^*(t)$ directly on Hardware, single short circuit element

Key Benefits

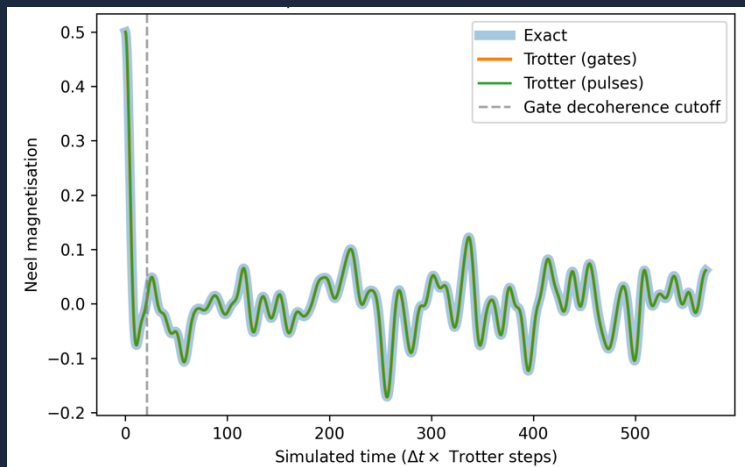
- Fewer gate layers → shorter circuit depth
- Reduced coherent error accumulation
- Lower susceptibility to decoherence
- Enables longer physical simulation times T

Results: Noiseless Simulation



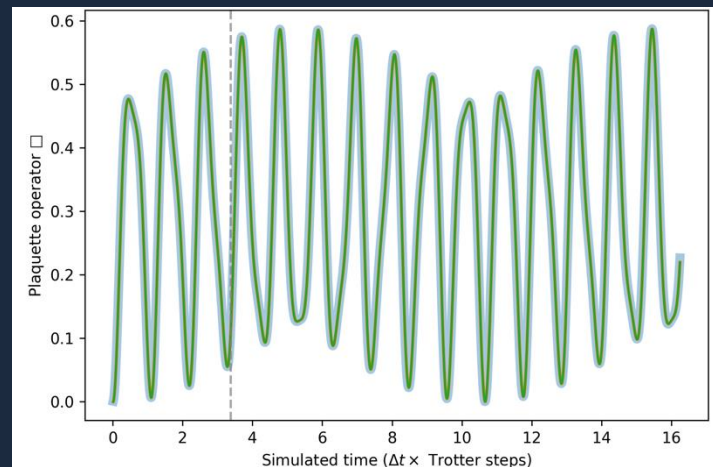
1D Fermi Hubbard

- **Simulation:** 6-site model
- **Single Trotter Step Execution:**
 - Standard Gates: 532ns
 - Pulse Approach: **136ns**
- **Global Budget:** 113ms



Z2 Lattice Gauge Theory

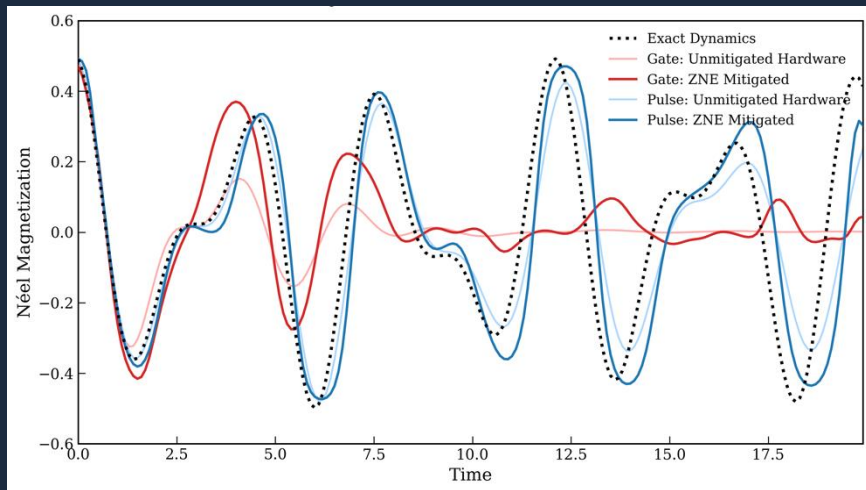
- **Simulation:** 2x2 lattice
- **Single Trotter Step Execution:**
 - Standard Gates: 336ns
 - Pulse Approach: **72ns**
- **Global Budget:** 113ms



Results: Noisy Simulations

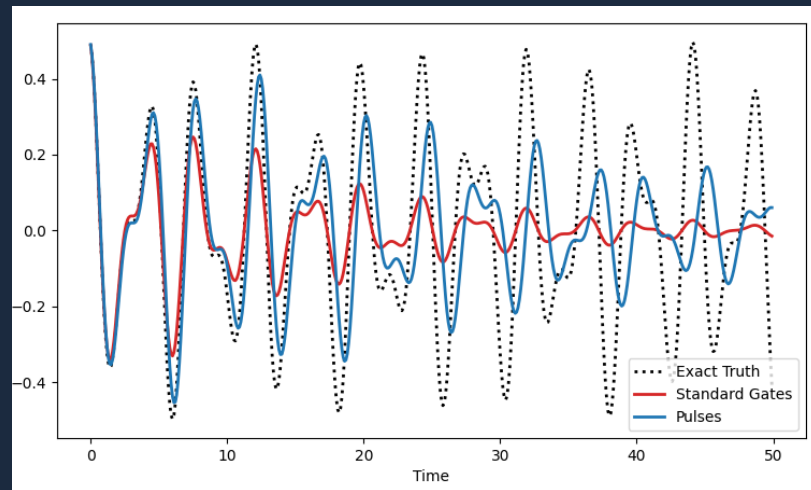
Coherent noise

Noise due to imperfections in our lasers



Decoherent noise

Noise due to interactions with environment



Noise model: T_1 relaxation 100ms & T_2 dephasing 50ms, pulse errors (multiplicative noise)

Summary

Quantum simulation

Trotterised evolution naturally maps onto quantum circuits, but gate depth limits simulation time

Quantum optimal control

GRAPE finds shaped pulses that implement target unitaries with minimal duration

Native pulse synthesis

By synthesising Trotter-step gates directly via GRAPE, circuit depth drops significantly

Key result

Shorter circuits → reduced coherent and incoherent errors → longer achievable simulation times