Obtaining continuum physics from dynamical simulations of Hamiltonian lattice gauge theories

Speaker: Christopher Kane[1,2]

Collaborators: Siddarth Hariprakash[3,4,5], Christian Bauer[4]

Based off work in: [arXiv:2506.16559]

QuantHEP 2025

Date: October 2, 2025

[1]Maryland Center for Fundamental Physics and Department of Physics
[2]Joint Center for Quantum Information and Computer Science
[3]Center for Theoretical Physics and Department of Physics, University of California, Berkeley
[4]Physics Division, Lawrence Berkeley National Laboratory
[5]National Energy Research Scientific Computing Center (NERSC)

Cost of a quantum simulation

- Calculating physical observables requires taking the continuum limit $a \to 0$
- Fair comparisons between methods must include this cost
- Error from approximate time evolution can spoil continuum limit
- Procedure to remove time evolution errors when using Trotter methods developed
  [Marcela Carena, Henry Lamm, Ying-Ying Li, Wanqiang Liu, PRD, arXiv:2107.01166]
  $\to$ not applicable to other algorithms
- **Our Goal:** Develop a general framework for controlling impact of time evolution errors on continuum limit applicable to any algorithm

Outline

1. Review continuum limit assuming exact time evolution

2. Review existing methods for treating Trotter errors using renormalization [Marcela Carena, Henry Lamm, Ying-Ying Li, Wanqiang Liu, PRD, arXiv:2107.01166]
→ Present simpler, alternative approach to treating Trotter errors

3. Present general procedure applicable to any time evolution algorithm
→ Statistically-Bounded Time Evolution Protocol

Hamiltonian LGT

Pure gauge Hamiltonian

$$H_{\mathrm{KS}} = \frac{1}{a}\left[g_t^2 \widetilde{H}_E - \frac{1}{g_s^2}\widetilde{H}_B\right]$$

Lorentz invariance broken $\longrightarrow$ gauge coupling for H_E and H_B different

Hamiltonian LGT

Pure gauge Hamiltonian

$$H_{\mathrm{KS}} = \frac{1}{a}\left[g_t^2\widetilde{H}_E - \frac{1}{g_s^2}\widetilde{H}_B\right]$$

Lorentz invariance broken $\longrightarrow$ gauge coupling for H_E and H_B different

- Speed of light: $c = \frac{g_t}{g_s}$
- Gauge coupling $g = \sqrt{g_s g_t}$

Hamiltonian LGT

Pure gauge Hamiltonian

$$H_{\mathrm{KS}} = \frac{1}{a}\left[g_t^2 \widetilde{H}_E - \frac{1}{g_s^2}\widetilde{H}_B\right]$$

Lorentz invariance broken $\longrightarrow$ gauge coupling for H_E and H_B different

- Speed of light: $c = \frac{g_t}{g_s}$
- Gauge coupling $g = \sqrt{g_s g_t}$

Speed of light $c(a) \neq 1$ changes overall scale of H:

$$H_{\mathrm{KS}} = \frac{c}{a}\left[g^2 \widetilde{H}_E - \frac{1}{g^2}\widetilde{H}_B\right]$$

Hamiltonian LGT: Two Different Scales

Hamiltonian

$$\hat{H}_{\mathsf{KS}} = a_t H_{\mathrm{KS}}$$

Temporal scale: $a_t = \frac{a}{c}$

(Notation: hatted quantities are dimensionless)

Hamiltonian LGT: Two Different Scales

Hamiltonian

$$\hat{H}_{\mathsf{KS}} = a_t H_{\mathrm{KS}}$$

Temporal scale: $a_t = \frac{a}{c}$

Momentum Operator

$$\hat{P}_{\mathrm{KS}} = a P_{\mathrm{KS}}$$

Spatial scale: a (lattice spacing)

(Notation: hatted quantities are dimensionless)

Hamiltonian LGT: Two Different Scales

Hamiltonian

$\hat{H}_{\mathrm{KS}} = a_t H_{\mathrm{KS}}$

Temporal scale: $a_t = \frac{a}{c}$

Momentum Operator

$\hat{P}_{\mathrm{KS}} = a P_{\mathrm{KS}}$

Spatial scale: a (lattice spacing)

Different "rulers" for temporal and spatial quantities

- **Speed of light c(a):** conversion factor between a and a_t

(Notation: hatted quantities are dimensionless)

Hamiltonian LGT: Two Different Scales

Hamiltonian

$$\hat{H}_{\mathrm{KS}} = a_t H_{\mathrm{KS}}$$

Temporal scale: $a_t = \frac{a}{c}$

Momentum Operator

$$\hat{P}_{\mathrm{KS}} = a P_{\mathrm{KS}}$$

Spatial scale: a (lattice spacing)

Different “rulers” for temporal and spatial quantities

- **Speed of light c(a):** conversion factor between a and a_t

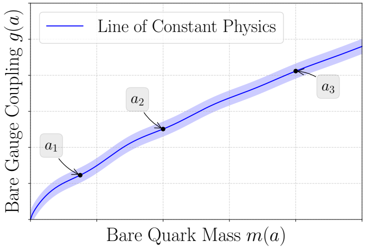
Dimensionless quantities measured on the lattice:

$$\hat{m} = a_t\, m\,, \qquad \hat{t} = \frac{1}{a_t} t\,, \qquad \hat{p} = ap, \qquad \hat{x} = \frac{1}{a} x$$

(Notation: hatted quantities are dimensionless)

Renormalization: taking the continuum limit

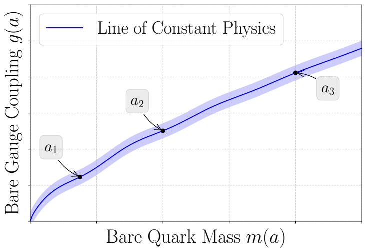
1. **Determine renormalization trajectory**



Renormalization: taking the continuum limit

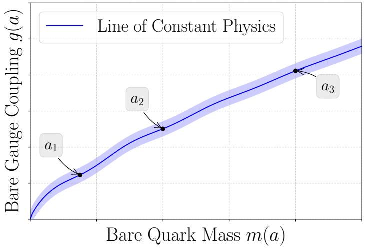
1. Determine renormalization trajectory

a Choose value of bare coupling $g(a)$



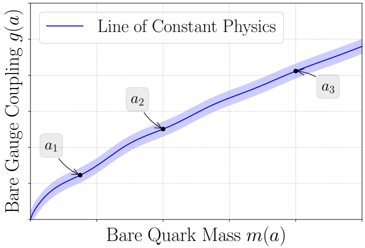
Renormalization: taking the continuum limit

1. **Determine renormalization trajectory**
   - **a** Choose value of bare coupling $g(a)$
   - **b** Tune bare parameters $c(a)$, $m(a)$ to reproduce known dimensionless quantities



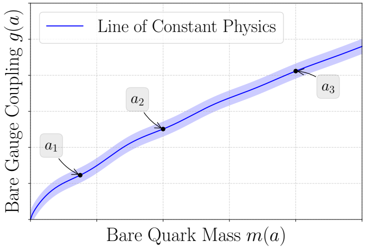
Renormalization: taking the continuum limit

1. **Determine renormalization trajectory**
   - **a** Choose value of bare coupling $g(a)$
   - **b** Tune bare parameters $c(a)$, $m(a)$ to reproduce known dimensionless quantities
   - **c** Determine lattice spacing a: compare to known dimensionful quantity



Renormalization: taking the continuum limit

1. **Determine renormalization trajectory**
   - **a** Choose value of bare coupling $g(a)$
   - **b** Tune bare parameters $c(a)$, $m(a)$ to reproduce known dimensionless quantities
   - **c** Determine lattice spacing a: compare to known dimensionful quantity

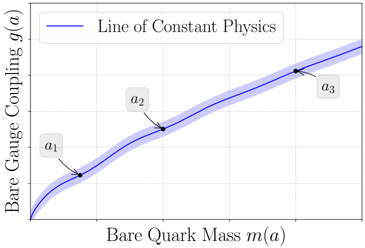


2. **Calculate desired physical observable**

Renormalization: taking the continuum limit

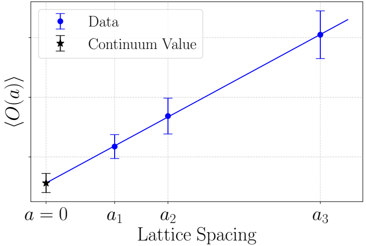
1. Determine renormalization trajectory

- **a** Choose value of bare coupling $g(a)$
- **b** Tune bare parameters $c(a)$, $m(a)$ to reproduce known dimensionless quantities
- **c** Determine lattice spacing a: compare to known dimensionful quantity



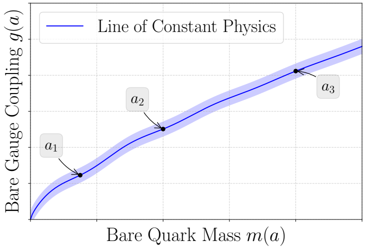
2. Calculate desired physical observable

- **a** Calculate dimensionless $\langle \hat{O}(\hat{t}, a) \rangle$ and rescale by a to convert to physical units

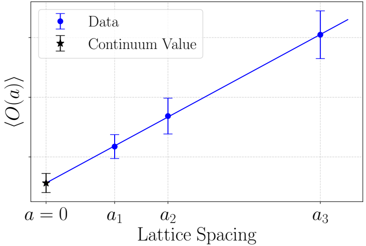


Renormalization: taking the continuum limit

1. **Determine renormalization trajectory**
   - **a** Choose value of bare coupling $g(a)$
   - **b** Tune bare parameters $c(a)$, $m(a)$ to reproduce known dimensionless quantities
   - **c** Determine lattice spacing a: compare to known dimensionful quantity



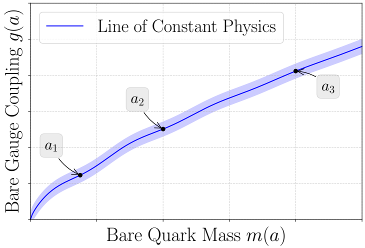
2. **Calculate desired physical observable**
   - **a** Calculate dimensionless $\langle \hat{O}(\hat{t}, a) \rangle$ and rescale by a to convert to physical units
   - **b** Renormalize operator $\hat{O}(t)$ (if necessary)



Renormalization: taking the continuum limit

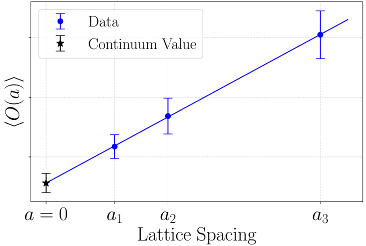
1. Determine renormalization trajectory

- **a** Choose value of bare coupling $g(a)$
- **b** Tune bare parameters $c(a)$, $m(a)$ to reproduce known dimensionless quantities
- **c** Determine lattice spacing a: compare to known dimensionful quantity



2. Calculate desired physical observable

- **a** Calculate dimensionless $\langle \hat{O}(\hat{t}, a) \rangle$ and rescale by a to convert to physical units
- **b** Renormalize operator $\hat{O}(t)$ (if necessary)
- **c** Extrapolate to continuum



Effective Hamiltonian with Trotter

*(hatted quantity $\leftrightarrow$ dimensionless)

Look at effective Hamiltonian simulated using Trotter [Marcela Carena, Henry Lamm, Ying-Ying Li, Wanqiang Liu, PRD, arXiv:2107.01166]

$$e^{-i\hat{\delta}_t(\hat{H}_E+\hat{H}_B)} \approx e^{-i\frac{\hat{\delta}_t}{2}\hat{H}_E} e^{-i\hat{\delta}_t\hat{H}_B} e^{-i\frac{\hat{\delta}_t}{2}\hat{H}_E}$$

Effective Hamiltonian with Trotter

*(hatted quantity $\leftrightarrow$ dimensionless)

Look at effective Hamiltonian simulated using Trotter [Marcela Carena, Henry Lamm, Ying-Ying Li, Wanqiang Liu, PRD, arXiv:2107.01166]

$$e^{-i\hat{\delta}_t(\hat{H}_E+\hat{H}_B)} \approx e^{-i\frac{\hat{\delta}_t}{2}\hat{H}_E} e^{-i\hat{\delta}_t\hat{H}_B} e^{-i\frac{\hat{\delta}_t}{2}\hat{H}_E} \equiv e^{-i\hat{\delta}_t\hat{H}_{\mathrm{eff}}}$$

Effective Hamiltonian with Trotter

*(hatted quantity $\leftrightarrow$ dimensionless)

Look at effective Hamiltonian simulated using Trotter [Marcela Carena, Henry Lamm, Ying-Ying Li, Wanqiang Liu, PRD, arXiv:2107.01166]

$$e^{-i\hat{\delta}_t(\hat{H}_E+\hat{H}_B)} \approx e^{-i\frac{\hat{\delta}_t}{2}\hat{H}_E} e^{-i\hat{\delta}_t\hat{H}_B} e^{-i\frac{\hat{\delta}_t}{2}\hat{H}_E} \equiv e^{-i\hat{\delta}_t\hat{H}_{\mathrm{eff}}}$$

where

$$\hat{H}_{\mathrm{eff}}(\delta_t) = \hat{H}_E + \hat{H}_B + \frac{\hat{\delta}_t^2}{24}\Big([\hat{H}_E,[\hat{H}_E,\hat{H}_B]] + 2[\hat{H}_B,[\hat{H}_E,\hat{H}_B]]\Big) + \dots$$

Effective Hamiltonian with Trotter

*(hatted quantity $\leftrightarrow$ dimensionless)

Look at effective Hamiltonian simulated using Trotter [Marcela Carena, Henry Lamm, Ying-Ying Li, Wanqiang Liu, PRD, arXiv:2107.01166]

$$e^{-i\hat{\delta}_t(\hat{H}_E+\hat{H}_B)} \approx e^{-i\frac{\hat{\delta}_t}{2}\hat{H}_E} e^{-i\hat{\delta}_t\hat{H}_B} e^{-i\frac{\hat{\delta}_t}{2}\hat{H}_E} \equiv e^{-i\hat{\delta}_t\hat{H}_{\text{eff}}}$$

where

$$\hat{H}_{\text{eff}}(\delta_t) = \hat{H}_E + \hat{H}_B + \frac{\hat{\delta}_t^2}{24}\Big([\hat{H}_E,[\hat{H}_E,\hat{H}_B]] + 2[\hat{H}_B,[\hat{H}_E,\hat{H}_B]]\Big) + \dots$$

- View as “temporal lattice”, treat $\hat{\delta}_t$ as parameter in the effective Hamiltonian
- $\hat{\delta}_t \neq 0$ changes physics which changes values used in tuning and scale setting

$$g(a) \to g(a,\delta_t), \qquad m(a) \to m(a,\delta_t), \qquad \dots$$

Continuum physics achieved taking simultaneous limit $\lim_{a\to 0}\lim_{\delta_t\to 0}$

(Or, work at fixed anisotropy $\xi = a/\delta_t$ and extrap $a \to 0$)

Euclidean simulations to assist renormalization

Main idea: relate H_{eff} to Euclidean transfer matrix T on anisotropic lattice

$$Z = \int \mathcal{D}[U] e^{-S_E} = \mathrm{Tr}\, T^N$$

Euclidean simulations to assist renormalization

Main idea: relate H_{eff} to Euclidean transfer matrix T on anisotropic lattice

$$Z = \int \mathcal{D}[U] e^{-S_E} = \mathrm{Tr}\, T^N$$

Tune bare parameters and scale set using Euclidean simulations by exploiting:

$$\exp(-i\hat{\delta}_t \hat{H}_{\mathrm{eff}}(\delta_t)) \underset{\hat{\delta}_t \to -ia_0 \frac{c}{a}}{\approx} T$$

Euclidean simulations to assist renormalization

Main idea: relate H_{eff} to Euclidean transfer matrix T on anisotropic lattice

$$Z = \int \mathcal{D}[U] e^{-S_E} = \mathrm{Tr}\, T^N$$

Tune bare parameters and scale set using Euclidean simulations by exploiting:

$$\exp(-i\hat{\delta}_t \hat{H}_{\mathrm{eff}}(\delta_t)) \underset{\hat{\delta}_t \to -ia_0 \frac{c}{a}}{\approx} T$$

Limitations:

- Relation exact for pure gauge only if one uses heat-kernel action

Euclidean simulations to assist renormalization

Main idea: relate H_{eff} to Euclidean transfer matrix T on anisotropic lattice

$$Z = \int \mathcal{D}[U] e^{-S_E} = \mathrm{Tr}\, T^N$$

Tune bare parameters and scale set using Euclidean simulations by exploiting:

$$\exp(-i\hat{\delta}_t \hat{H}_{\mathrm{eff}}(\delta_t)) \underset{\hat{\delta}_t \to -i a_0 \frac{c}{a}}{\approx} T$$

Limitations:

- Relation exact for pure gauge only if one uses heat-kernel action
- Only true for 2nd order PF with this precise Trotter splitting

Euclidean simulations to assist renormalization

Main idea: relate H_{eff} to Euclidean transfer matrix T on anisotropic lattice

$$Z = \int \mathcal{D}[U] e^{-S_E} = \mathrm{Tr}\, T^N$$

Tune bare parameters and scale set using Euclidean simulations by exploiting:

$$\exp(-i\hat{\delta}_t \hat{H}_{\text{eff}}(\delta_t)) \underset{\hat{\delta}_t \to -ia_0 \frac{c}{a}}{\approx} T$$

Limitations:

- Relation exact for pure gauge only if one uses heat-kernel action
- Only true for 2nd order PF with this precise Trotter splitting
- Including fermions introduces $\mathcal{O}(a_0)$ systematics

Euclidean simulations to assist renormalization

Main idea: relate H_{eff} to Euclidean transfer matrix T on anisotropic lattice

$$Z = \int \mathcal{D}[U] e^{-S_E} = \mathrm{Tr}\, T^N$$

Tune bare parameters and scale set using Euclidean simulations by exploiting:

$$\exp(-i\hat{\delta}_t \hat{H}_{\mathrm{eff}}(\delta_t)) \underset{\hat{\delta}_t \to -ia_0 \frac{c}{a}}{\approx} T$$

Limitations:

- Relation exact for pure gauge only if one uses heat-kernel action
- Only true for 2nd order PF with this precise Trotter splitting
- Including fermions introduces $\mathcal{O}(a_0)$ systematics
- Does not reduce size and quality of quantum device needed

Alternative Approach

Look more closely at H_{eff}:

$$\hat{H}_{\text{eff}}(\delta_t) = \hat{H}_E + \hat{H}_B + \frac{\hat{\delta}_t^2}{24}\left([\hat{H}_E, [\hat{H}_E, \hat{H}_B]] + 2[\hat{H}_B, [\hat{H}_E, \hat{H}_B]]\right) + \dots$$

Alternative Approach

Look more closely at H_{eff}:

$$\hat{H}_{\text{eff}}(\delta_t) = \hat{H}_E + \hat{H}_B + \underbrace{\frac{\hat{\delta}_t^2}{24}\left([\hat{H}_E, [\hat{H}_E, \hat{H}_B]] + 2[\hat{H}_B, [\hat{H}_E, \hat{H}_B]]\right) + \dots}_{\text{irrelevant operators}}$$

Alternative Approach

Look more closely at H_{eff}:

$$\hat{H}_{\text{eff}}(\delta_t) = \hat{H}_E + \hat{H}_B + \underbrace{\frac{\hat{\delta}_t^2}{24}\Big([\hat{H}_E,[\hat{H}_E,\hat{H}_B]] + 2[\hat{H}_B,[\hat{H}_E,\hat{H}_B]]\Big) + \dots}_{\text{irrelevant operators}}$$

Simple alternative renormalization trajectory:

Alternative Approach

Look more closely at H_{eff}:

$$\hat{H}_{\mathrm{eff}}(\delta_t) = \hat{H}_E + \hat{H}_B + \underbrace{\frac{\hat{\delta}_t^2}{24}\left([\hat{H}_E,[\hat{H}_E,\hat{H}_B]] + 2[\hat{H}_B,[\hat{H}_E,\hat{H}_B]]\right) + \dots}_{\text{irrelevant operators}}$$

Simple alternative renormalization trajectory:

- Fix bare Trotter step-size $\hat{\delta}_t$

Alternative Approach

Look more closely at H_{eff}:

$$\hat{H}_{\text{eff}}(\delta_t) = \hat{H}_E + \hat{H}_B + \frac{\hat{\delta}_t^2}{24} \underbrace{\left([\hat{H}_E, [\hat{H}_E, \hat{H}_B]] + 2[\hat{H}_B, [\hat{H}_E, \hat{H}_B]]\right) + \dots}_{\text{irrelevant operators}}$$

Simple alternative renormalization trajectory:

- Fix bare Trotter step-size $\hat{\delta}_t$
- Tune bare parameters assuming $\hat{\delta}_t = 0$, only introduces $\mathcal{O}(a^2)$ errors
  $\rightarrow$ simplifies renormalization to exact time evolution case

Alternative Approach

Look more closely at H_{eff}:

$$\hat{H}_{\text{eff}}(\delta_t) = \hat{H}_E + \hat{H}_B + \frac{\hat{\delta}_t^2}{24} \underbrace{\left([\hat{H}_E, [\hat{H}_E, \hat{H}_B]] + 2[\hat{H}_B, [\hat{H}_E, \hat{H}_B]]\right) + \ldots}_{\text{irrelevant operators}}$$

Simple alternative renormalization trajectory:

- Fix bare Trotter step-size $\hat{\delta}_t$
- Tune bare parameters assuming $\hat{\delta}_t = 0$, only introduces $\mathcal{O}(a^2)$ errors
  $\rightarrow$ simplifies renormalization to exact time evolution case
- Trotter errors go to zero as $\mathcal{O}(a^2)$ in continuum limit

Extension to other algorithms?

Can we apply a similar procedure to simulations done using other simulation algorithms?

Consider Quantum Signal Processing as test case

Quantum Signal Processing: high level review

Hamiltonian input model: given $|\psi\rangle \to H|\psi\rangle$, one can implement $f(H)|\psi\rangle$

Quantum Signal Processing: high level review

Hamiltonian input model: given $|\psi\rangle \to H\,|\psi\rangle$, one can implement $f(H)\,|\psi\rangle$

This is achieved via a **block encoding** of H/λ where $\lambda \geq \|H\|$

$$U_H = \begin{pmatrix} \frac{H}{\lambda} & * \\ * & * \end{pmatrix} \to \begin{pmatrix} \frac{H}{\lambda} & * \\ * & * \end{pmatrix} \begin{pmatrix} |\psi\rangle \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{H}{\lambda}\,|\psi\rangle \\ * \end{pmatrix}$$

Quantum Signal Processing: high level review

Hamiltonian input model: given $|\psi\rangle \to H|\psi\rangle$, one can implement $f(H)|\psi\rangle$

This is achieved via a **block encoding** of H/λ where $\lambda \geq \|H\|$

$$U_H = \begin{pmatrix} \frac{H}{\lambda} & * \\ * & * \end{pmatrix} \to \begin{pmatrix} \frac{H}{\lambda} & * \\ * & * \end{pmatrix} \begin{pmatrix} |\psi\rangle \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{H}{\lambda}|\psi\rangle \\ * \end{pmatrix}$$

Quantum Signal Processing $\to$ d calls to U_H can implement $f(H)|\psi\rangle$ where $\deg(f) \sim d$

[G. Low, I. Chuang, Quantum, arXiv:1606.02685], [G. Low, I. Chuang, Quantum, arXiv:1610.06546]

Quantum Signal Processing: high level review

Hamiltonian input model: given $|\psi\rangle \to H|\psi\rangle$, one can implement $f(H)|\psi\rangle$

This is achieved via a **block encoding** of H/λ where $\lambda \geq \|H\|$

$$U_H = \begin{pmatrix} \frac{H}{\lambda} & * \\ * & * \end{pmatrix} \to \begin{pmatrix} \frac{H}{\lambda} & * \\ * & * \end{pmatrix} \begin{pmatrix} |\psi\rangle \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{H}{\lambda}|\psi\rangle \\ * \end{pmatrix}$$

Quantum Signal Processing $\to$ d calls to U_H can implement $f(H)|\psi\rangle$ where $\deg(f) \sim d$

[G. Low, I. Chuang, Quantum, arXiv:1606.02685], [G. Low, I. Chuang, Quantum, arXiv:1610.06546]

Using Jacobi-Anger expansion for time evolution:

$$e^{-iHt} \approx J_0(\lambda t) + 2\sum_{k>0 \text{ even}}^{d} (-i)^{k/2} J_k(\lambda t) T_k(H/\lambda) - 2i \sum_{k \text{ odd}}^{d} (-i)^{(k-1)/2} J_k(\lambda t) T_k(H/\lambda)$$

Quantum Signal Processing: high level review

Hamiltonian input model: given $|\psi\rangle \to H|\psi\rangle$, one can implement $f(H)|\psi\rangle$

This is achieved via a **block encoding** of H/λ where $\lambda \geq \|H\|$

$$U_H = \begin{pmatrix} \frac{H}{\lambda} & * \\ * & * \end{pmatrix} \to \begin{pmatrix} \frac{H}{\lambda} & * \\ * & * \end{pmatrix} \begin{pmatrix} |\psi\rangle \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{H}{\lambda}|\psi\rangle \\ * \end{pmatrix}$$

Quantum Signal Processing $\to$ d calls to U_H can implement $f(H)|\psi\rangle$ where $\deg(f) \sim d$

[G. Low, I. Chuang, Quantum, arXiv:1606.02685], [G. Low, I. Chuang, Quantum, arXiv:1610.06546]

Using Jacobi-Anger expansion for time evolution:

$$e^{-iHt} \approx J_0(\lambda t) + 2\sum_{k>0 \text{ even}}^{d} (-i)^{k/2} J_k(\lambda t) T_k(H/\lambda) - 2i \sum_{k \text{ odd}}^{d} (-i)^{(k-1)/2} J_k(\lambda t) T_k(H/\lambda)$$

Provably optimal scaling with t and ϵ [G. Low, I. Chuang, Quantum, arXiv:1606.02685]

$$\text{Calls to } U_H = \mathcal{O}\big(\lambda t + \log\frac{1}{\epsilon}\big)$$

Breakdown of previous approach to QSP

Can we apply previous approaches to control time evolution errors as $a \to 0$ for QSP?

Approximate time evolution operator

$$e^{-iHt} \approx J_0(\lambda t) + 2 \sum_{k>0 \text{ even}}^{d} (-i)^{k/2} J_k(\lambda t) T_k(H/\lambda) - 2i \sum_{k \text{ odd}}^{d} (-i)^{(k-1)/2} J_k(\lambda t) T_k(H/\lambda)$$

Breakdown of previous approach to QSP

Can we apply previous approaches to control time evolution errors as $a \to 0$ for QSP?

Approximate time evolution operator

$$e^{-iHt} \approx J_0(\lambda t) + 2 \sum_{k>0 \text{ even}}^{d} (-i)^{k/2} J_k(\lambda t) T_k(H/\lambda) - 2i \sum_{k \text{ odd}}^{d} (-i)^{(k-1)/2} J_k(\lambda t) T_k(H/\lambda)$$

Previous picture relies on effective Hamiltonian formalism

Unclear how to proceed:
$\to$ approximate evolution is not unitary, not clear what H_{eff} is

We need an alternative more general approach

Exact time evolution introduces no UV divergences

Exact time evolution introduces no UV divergences

Define δ as parameter that controls time evolution error

- For Trotter δ is the timestep δ_t
- For QSP δ is $\sim 1/d$, with d degree of the polynomial approx to e^{-iHt}

Exact time evolution introduces no UV divergences

Define δ as parameter that controls time evolution error

- For Trotter δ is the timestep δ_t
- For QSP δ is $\sim 1/d$, with d degree of the polynomial approx to e^{-iHt}

Key insight: unlike taking $a \to 0$, taking $\delta \to 0$ does not introduce divergences

$$\lim_{\delta \to 0} g(a, \delta) = g(a), \qquad \lim_{\delta \to 0} m(a, \delta) = m(a)$$

Exact time evolution introduces no UV divergences

Define δ as parameter that controls time evolution error

- For Trotter δ is the timestep δ_t
- For QSP δ is $\sim 1/d$, with d degree of the polynomial approx to e^{-iHt}

Key insight: unlike taking $a \to 0$, taking $\delta \to 0$ does not introduce divergences

$$\lim_{\delta \to 0} g(a, \delta) = g(a), \qquad \lim_{\delta \to 0} m(a, \delta) = m(a)$$

Implication: instead of treating $\delta \neq 0$ with machinery of renormalization, simply treat as source of systematic uncertainty

Exact time evolution introduces no UV divergences

Define δ as parameter that controls time evolution error

- For Trotter δ is the timestep δ_t
- For QSP δ is $\sim 1/d$, with d degree of the polynomial approx to e^{-iHt}

Key insight: unlike taking $a \to 0$, taking $\delta \to 0$ does not introduce divergences

$$\lim_{\delta \to 0} g(a, \delta) = g(a), \qquad \lim_{\delta \to 0} m(a, \delta) = m(a)$$

Implication: instead of treating $\delta \neq 0$ with machinery of renormalization, simply treat as source of systematic uncertainty
$\to$ driving time evolution error much below statistical precision effectively takes $\delta \to 0$

Exact time evolution introduces no UV divergences

Define δ as parameter that controls time evolution error

- For Trotter δ is the timestep δ_t
- For QSP δ is $\sim 1/d$, with d degree of the polynomial approx to e^{-iHt}

Key insight: unlike taking $a \to 0$, taking $\delta \to 0$ does not introduce divergences

$$\lim_{\delta \to 0} g(a, \delta) = g(a), \qquad \lim_{\delta \to 0} m(a, \delta) = m(a)$$

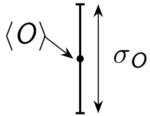
Implication: instead of treating $\delta \neq 0$ with machinery of renormalization, simply treat as source of systematic uncertainty

$\to$ driving time evolution error much below statistical precision effectively takes $\delta \to 0$

$\to$ renormalization procedure simplifies to exact time evolution case

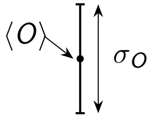
Statistically-Bounded Time Evolution (SBTE) Protocol

Calculate time-dependent observable $\langle \hat{O}(t, a) \rangle$ to uncertainty σ_O
$\rightarrow$ sources of uncertainty from shot noise, device noise, etc.



Statistically-Bounded Time Evolution (SBTE) Protocol

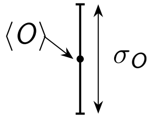
Calculate time-dependent observable $\langle \hat{O}(t, a) \rangle$ to uncertainty σ_O
$\rightarrow$ sources of uncertainty from shot noise, device noise, etc.



If approximate time evolution error $\epsilon_O \ll \sigma_O$, we can neglect it

Statistically-Bounded Time Evolution (SBTE) Protocol

Calculate time-dependent observable $\langle \hat{O}(t, a) \rangle$ to uncertainty σ_O
$\rightarrow$ sources of uncertainty from shot noise, device noise, etc.



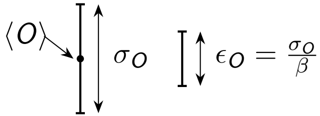
If approximate time evolution error $\epsilon_O \ll \sigma_O$, we can neglect it

For some $\beta > 1$, we require

$$\epsilon_O \equiv \| e^{iHt} \hat{O}(0, a) e^{-iHt} - U_\delta^\dagger(t) \hat{O}(0, a) U_\delta(t) \| \leq \frac{\sigma_O}{\beta}$$

Statistically-Bounded Time Evolution (SBTE) Protocol

Calculate time-dependent observable $\langle \hat{O}(t,a) \rangle$ to uncertainty σ_O
$\rightarrow$ sources of uncertainty from shot noise, device noise, etc.



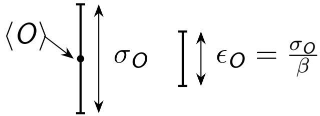
If approximate time evolution error $\epsilon_O \ll \sigma_O$, we can neglect it

For some $\beta > 1$, we require

$$\epsilon_O \equiv \| e^{iHt} \hat{O}(0,a) e^{-iHt} - U_\delta^\dagger(t) \hat{O}(0,a) U_\delta(t) \| \leq \frac{\sigma_O}{\beta}$$

Statistically-Bounded Time Evolution (SBTE) Protocol

Calculate time-dependent observable $\langle \hat{O}(t, a) \rangle$ to uncertainty σ_O
$\rightarrow$ sources of uncertainty from shot noise, device noise, etc.



If approximate time evolution error $\epsilon_O \ll \sigma_O$, we can neglect it

For some $\beta > 1$, we require

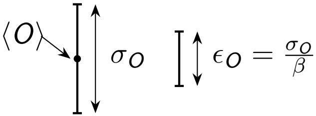
$$\epsilon_O \equiv \| e^{iHt} \hat{O}(0, a) e^{-iHt} - U_\delta^\dagger(t) \hat{O}(0, a) U_\delta(t) \| \leq \frac{\sigma_O}{\beta}$$

This is guaranteed by choosing time evolution operator error ϵ_{sim} as:

$$\epsilon_{\text{sim}} = \| e^{-iHt} - U_\delta(t) \| \leq \left(\frac{2\beta \| \hat{O}(0, a) \|}{\sigma_O} \right)^{-1}$$

Statistically-Bounded Time Evolution (SBTE) Protocol

Calculate time-dependent observable $\langle \hat{O}(t,a) \rangle$ to uncertainty σ_O
$\rightarrow$ sources of uncertainty from shot noise, device noise, etc.



If approximate time evolution error $\epsilon_O \ll \sigma_O$, we can neglect it

For some $\beta > 1$, we require

$$\epsilon_O \equiv \| e^{iHt} \hat{O}(0,a) e^{-iHt} - U_\delta^\dagger(t) \hat{O}(0,a) U_\delta(t) \| \leq \frac{\sigma_O}{\beta}$$

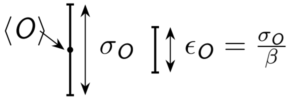
This is guaranteed by choosing time evolution operator error ϵ_{sim} as:

$$\epsilon_{\mathrm{sim}} = \| e^{-iHt} - U_\delta(t) \| \leq \left(\frac{2\beta \| \hat{O}(0,a) \|}{\sigma_O} \right)^{-1}$$

Key question: what is the additional computational cost to ensure this?

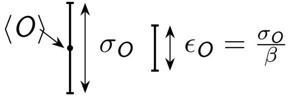
Examples: Product formula and Quantum Signal Processing

Set simulation error $\epsilon_{\text{sim}} = \left(\frac{2\beta\|O(0,a)\|}{\sigma_O}\right)^{-1}$



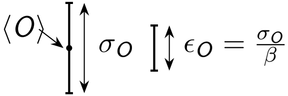
Examples: Product formula and Quantum Signal Processing

Set simulation error $\epsilon_{\text{sim}} = \left(\frac{2\beta\|O(0,a)\|}{\sigma_O}\right)^{-1}$



SBTE applied to Product Formulas:

Examples: Product formula and Quantum Signal Processing



Set simulation error $\epsilon_{\rm sim} = \left(\frac{2\beta\|O(0,a)\|}{\sigma_O}\right)^{-1}$

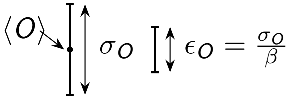
SBTE applied to Product Formulas:

$$\text{Trotter number}: \quad N_{\rm PF} \geq \left(\frac{1}{\epsilon_{\rm sim}}\right)^{1/p} \widetilde{\alpha}^{1/p} t^{1+1/p}$$

[A. Childs, Y. Su, M. Tran, S. Zhu, PRX arXiv:1912.08854]

Examples: Product formula and Quantum Signal Processing

Set simulation error $\epsilon_{\rm sim} = \left(\frac{2\beta\|O(0,a)\|}{\sigma_O}\right)^{-1}$



SBTE applied to Product Formulas:

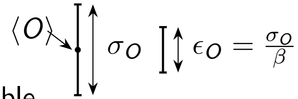
$$\text{Trotter number :} \quad N_{\rm PF} \geq \left(\frac{1}{\epsilon_{\rm sim}}\right)^{1/p} \widetilde{\alpha}^{1/p} t^{1+1/p}$$

$$\geq \left(\frac{2\beta\|O(0,a)\|}{\sigma_O}\right)^{1/p} \widetilde{\alpha}^{1/p} t^{1+1/p}$$

[A. Childs, Y. Su, M. Tran, S. Zhu, PRX arXiv:1912.08854]

Examples: Product formula and Quantum Signal Processing

Set simulation error $\epsilon_{\text{sim}} = \left(\frac{2\beta\|O(0,a)\|}{\sigma_O}\right)^{-1}$



SBTE applied to Product Formulas: extra cost not obviously negligible

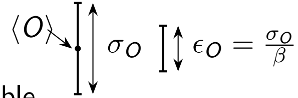
$$\text{Trotter number}: \quad N_{\text{PF}} \geq \left(\frac{1}{\epsilon_{\text{sim}}}\right)^{1/p} \widetilde{\alpha}^{1/p} t^{1+1/p}$$

[A. Childs, Y. Su, M. Tran, S. Zhu, PRX arXiv:1912.08854]

$$\geq \left(\frac{2\beta\|O(0,a)\|}{\sigma_O}\right)^{1/p} \widetilde{\alpha}^{1/p} t^{1+1/p}$$

Examples: Product formula and Quantum Signal Processing

Set simulation error $\epsilon_{\mathrm{sim}} = \left(\frac{2\beta\|O(0,a)\|}{\sigma_O}\right)^{-1}$



SBTE applied to Product Formulas: extra cost not obviously negligible

$$\text{Trotter number}: \quad N_{\mathrm{PF}} \geq \left(\frac{1}{\epsilon_{\mathrm{sim}}}\right)^{1/p} \widetilde{\alpha}^{1/p} t^{1+1/p}$$

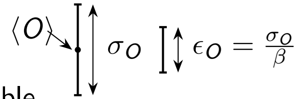
[A. Childs, Y. Su, M. Tran, S. Zhu, PRX arXiv:1912.08854]

$$\geq \left(\frac{2\beta\|O(0,a)\|}{\sigma_O}\right)^{1/p} \widetilde{\alpha}^{1/p} t^{1+1/p}$$

SBTE applied to QSP:

Examples: Product formula and Quantum Signal Processing

Set simulation error $\epsilon_{\text{sim}} = \left(\frac{2\beta\|O(0,a)\|}{\sigma_O}\right)^{-1}$



SBTE applied to Product Formulas: extra cost not obviously negligible

$$\text{Trotter number}: \quad N_{\text{PF}} \geq \left(\frac{1}{\epsilon_{\text{sim}}}\right)^{1/p} \widetilde{\alpha}^{1/p} t^{1+1/p}$$

$$\geq \left(\frac{2\beta\|O(0,a)\|}{\sigma_O}\right)^{1/p} \widetilde{\alpha}^{1/p} t^{1+1/p}$$

[A. Childs, Y. Su, M. Tran, S. Zhu, PRX arXiv:1912.08854]

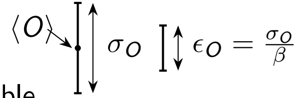
SBTE applied to QSP:

$$\text{Calls to Block Encoding}: \quad N_{\text{QSP}} \geq \frac{e\lambda t}{2} + \log\frac{1}{\epsilon_{\text{sim}}}, \qquad \lambda > \|H\|$$

[G. Low, I. Chuang, Quantum, arXiv:1606.02685]

Examples: Product formula and Quantum Signal Processing

Set simulation error $\epsilon_{\rm sim} = \left(\frac{2\beta\|O(0,a)\|}{\sigma_O}\right)^{-1}$



SBTE applied to Product Formulas: extra cost not obviously negligible

$$\text{Trotter number}: \quad N_{\rm PF} \geq \left(\frac{1}{\epsilon_{\rm sim}}\right)^{1/p} \widetilde{\alpha}^{1/p} t^{1+1/p}$$

$$\geq \left(\frac{2\beta\|O(0,a)\|}{\sigma_O}\right)^{1/p} \widetilde{\alpha}^{1/p} t^{1+1/p}$$

[A. Childs, Y. Su, M. Tran, S. Zhu, PRX arXiv:1912.08854]

SBTE applied to QSP:

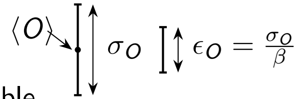
$$\text{Calls to Block Encoding}: \quad N_{\rm QSP} \geq \frac{e\lambda t}{2} + \log\frac{1}{\epsilon_{\rm sim}}, \qquad \lambda > \|H\|$$

$$\geq \frac{e\lambda t}{2} + \log\left(\frac{2\beta\|O(0,a)\|}{\sigma_O}\right)$$

[G. Low, I. Chuang, Quantum, arXiv:1606.02685]

Examples: Product formula and Quantum Signal Processing

Set simulation error $\epsilon_{\rm sim} = \left(\frac{2\beta\|O(0,a)\|}{\sigma_O}\right)^{-1}$



SBTE applied to Product Formulas: extra cost not obviously negligible

$$\text{Trotter number}: \quad N_{\rm PF} \geq \left(\frac{1}{\epsilon_{\rm sim}}\right)^{1/p} \widetilde{\alpha}^{1/p} t^{1+1/p}$$

$$\geq \left(\frac{2\beta\|O(0,a)\|}{\sigma_O}\right)^{1/p} \widetilde{\alpha}^{1/p} t^{1+1/p}$$

[A. Childs, Y. Su, M. Tran, S. Zhu, PRX arXiv:1912.08854]

SBTE applied to QSP: extra cost negligible *a priori*

$$\text{Calls to Block Encoding}: \quad N_{\rm QSP} \geq \frac{e\lambda t}{2} + \log\frac{1}{\epsilon_{\rm sim}}, \qquad \lambda > \|H\|$$

$$\geq \frac{e\lambda t}{2} + \log\left(\frac{2\beta\|O(0,a)\|}{\sigma_O}\right)$$

[G. Low, I. Chuang, Quantum, arXiv:1606.02685]

Summary and conclusions

- Full cost of a quantum simulation requires cost of $a \to 0$ limit
- Previously, controlling impact of approximate time evolution errors only understood for Trotter methods
- We introduced a simple, general approach applicable to any quantum algorithm
  $\to$ Statistically-Bounded Time Evolution Protocol
- Opens the door to performing end-to-end cost comparisons between different algorithms

Backup slides

Hamiltonian LGT

Kogut-Susskind Hamiltonian

$$H_{\text{KS}} = \left[\frac{g_t^2}{a}\widetilde{H}_E - \frac{1}{ag_s^2}\widetilde{H}_B + \frac{\kappa}{a}\widetilde{H}_{\text{hop}} + m\widetilde{H}_M\right]$$

Lorentz invariance broken $\longrightarrow$ gauge coupling for H_E and H_B different

- Speed of light: $c = \frac{g_t}{g_s}$
- Gauge coupling $g = \sqrt{g_s g_t}$
- Hopping coefficient κ
  $\rightarrow$ Euclidean simulations on anisotropic lattices have bare fermionic anisotropy factor γ_f
  $\rightarrow$ need γ_f so physical anisotropy $\xi = a/a_0$ "seen" by gluons and fermions is the same
  $\rightarrow$ keeping track of γ_f in Transfer matrix implies need for $\kappa(a) \neq 1$

Speed of light $c(a) \neq 1$ changes overall scale of H:

$$H_{\text{KS}} = \frac{c}{a}\left[g^2\widetilde{H}_E - \frac{1}{g^2}\widetilde{H}_B + \frac{\kappa}{c}\widetilde{H}_{\text{hop}} + (\frac{a}{c}m)\widetilde{H}_M\right] \equiv \frac{1}{a_t}\hat{H}_{\text{KS}}$$