LQCD simulations with GPT

Tutorials

Jong-Wan Lee (IBS)

LQCDW1 @ Sejong University

Jan. 4, 2024

A workflow of lattice simulations

Gauge theory (e.g QCD) discretized on the lattice of a 4-dim. Euclidean space-time (UV theory)

Finite lattice spacing

Generate gauge field configurations

Finite volume

Finite number of configurations (statistics)

Sequential, computationally expensive

Measurements of physical observables (correlation functions)

Statistical analysis

Scale setting

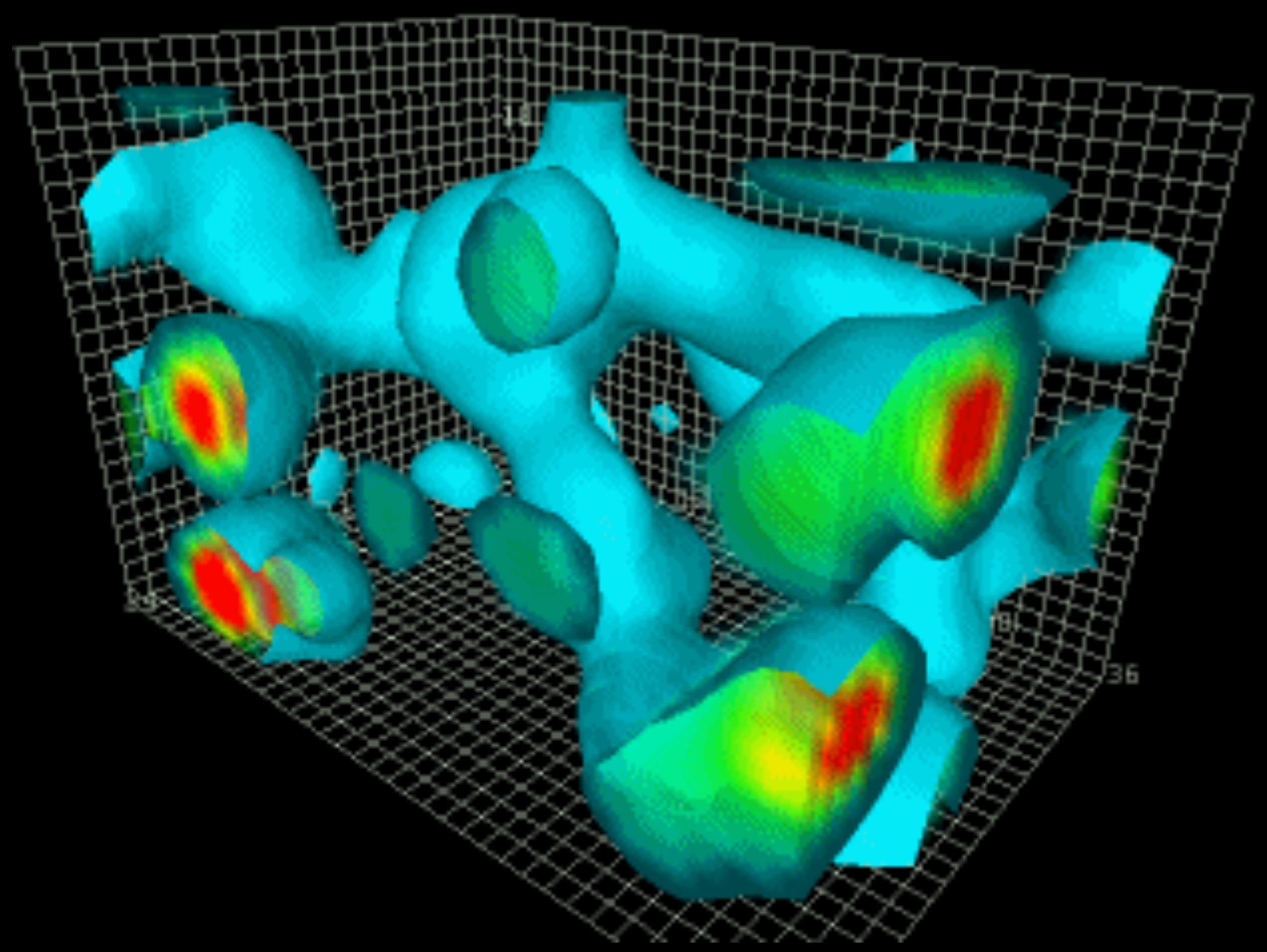
Data analysis

Continuum, infinite volume (thermodynamic) & massless or physical-mass

Goal:

$$S_{QCD} = \int d^4x \frac{1}{4} F^a_{\mu\nu} F^a_{\mu\nu} + \sum_f \int d^4x \bar{\psi}_f(x)(\gamma_\mu D_\mu + m_f)\psi_f(x)$$

First-principal lattice calculations



π, ρ meson masses

Goal:

$$S_{QCD} = \int d^4x \frac{1}{4} F^a_{\mu\nu} F^a_{\mu\nu} + \sum_f \int d^4x \bar{\psi}_f(x)(\gamma_\mu D_\mu + m_f)\psi_f(x)$$

First-principal lattice calculations

Grid **P**ython **T**oolkit (GPT)

GRID

π, ρ meson masses

GRID

✓ A C++ data parallel interface for cartesian grid problems, especially lattice QCD

Performance portable across many CPU's $\quad$ SIMD $\otimes$ OpenMP $\otimes$ MPI

Performance portable to GPU's $\quad$ SIMT $\otimes$ offload $\otimes$ MPI

✓ Give performance portability between many Exascale architectures

✓ Developed & maintained by P. Boyle, G. Cossu, A. Potelli & A. Yamaguchi @ Edinburgh, UK

github link https://github.com/paboyle/Grid

FRONTIER: AMD CPU, AMD GPU - HIP

Perlmutter: AMD CPU, Nvidia GPU - CUDA

Aurora: Intel CPU, Intel GPU - SYCL

Tesseract: Intel Skylake - OpenMP, SIMD

Grid Python Toolkit (GPT)

- ✓ A toolkit for lattice QCD and related theories
- ✓ Python frontend and C++ backend - modularity & composability

  Build up from modular high-performance components, several layers of composability
- ✓ Built on GRID - performance portability
- ✓ Developed by Christoph Lehner (Regenburg) et al

Python script / Jupyter notebook

gpt (Python)

- Defines data types and objects (group structures etc.)
- Expression engine (linear algebra)
- Algorithms (Solver, Eigensystem, . . .)
- File formats
- Stencils / global data transfers
- QCD, QIS, ML subsystems

cgpt (Python library written in C++)

- Global data transfer system (gpt creates pattern, cgpt optimizes data movement plan)
- Virtual lattices (tensors built from multiple Grid tensors)
- Optimized blocking, linear algebra, and Dirac operators
- Vectorized ranlux-like pRNG (parallel seed through 3xSHA256)

Grid

Eigen

FFTW

Talk by C. Lehner @ Lattice Practice (2023)

Grid Python Toolkit (GPT)

Thought that GPT would be good for LQCD beginners thanks to its accessibility & modularity while keeping its high performance on various architectures.

github link

https://github.com/lehner/gpt

Lectures by C. Lehner

https://homepages.uni-regensburg.de/~lec17310/teaching/wise2324/lqft.html

Let’s get started

1. Simulate QCD on the lattice !!!

(Generating gauge field configurations)

$$\int \mathrm{d}[U]\,\det D(U) \exp[-S_G]$$

Lattice and Fields

- Discrete lattice in 4-dimensional Euclidean space-time $T \times L \times L \times L$

- Field variables of various types, scalar-, vector- and matrix-valued, can be introduced at each site $[x, y, z, t]$. Particularly important ones are vector- and matrix-valued variables for gluons, pseudo-fermions, and operators with spin and color indices.

$$\phi(x, y, z, t) : \quad \phi[x, y, z, t]$$

$$U_\mu(x, y, z, t)^a{}_b : \quad U[\mu][x, y, z, t][a, b]$$

Lattice Gauge Action

- Wilson gauge action

$$S_W = \frac{1}{2}\beta \sum_x \sum_{\mu,\nu=0}^{d-1} (1 - P_{\mu\nu}(x)) = \beta \sum_x \sum_{\mu<\nu} (1 - P_{\mu\nu}(x))$$

- The plaquette is given by

$$P_{\mu\nu}(x) = \frac{1}{N_c} \mathrm{Re}\,\mathrm{Tr}\, U_{\mu\nu}(x)$$ with $$\beta = \frac{2N_c}{g^2}$$ and $$U_{\mu\nu}(x) = U_\mu(x) U_\nu(x + a\hat{\mu}) U_\mu^\dagger(x + a\hat{\nu}) U_\nu^\dagger(x)$$

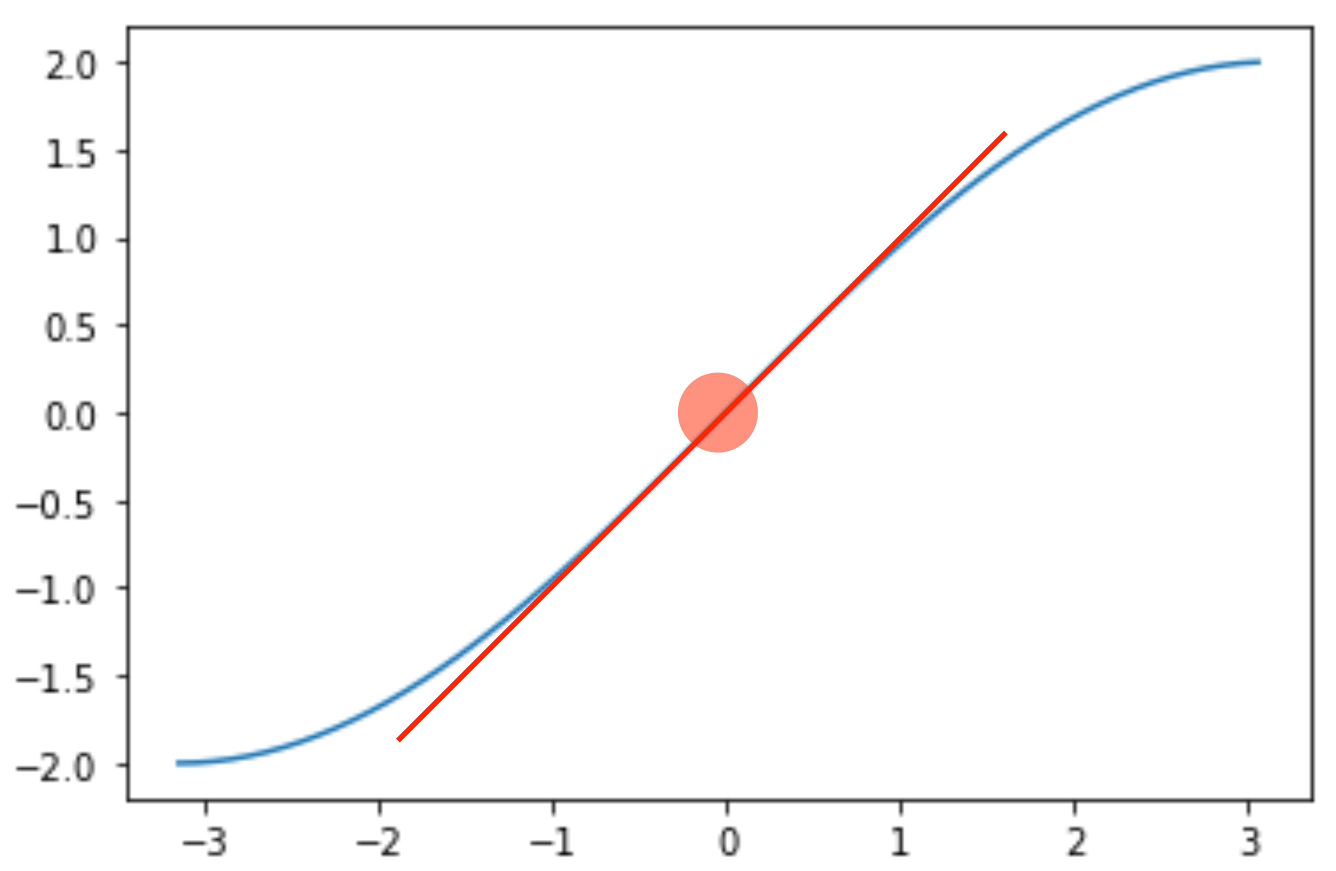
- Expected discretization errors are in the order of a.

- Can be improved by taking appropriate combinations of other Wilson loops

Lattice Fermion Action

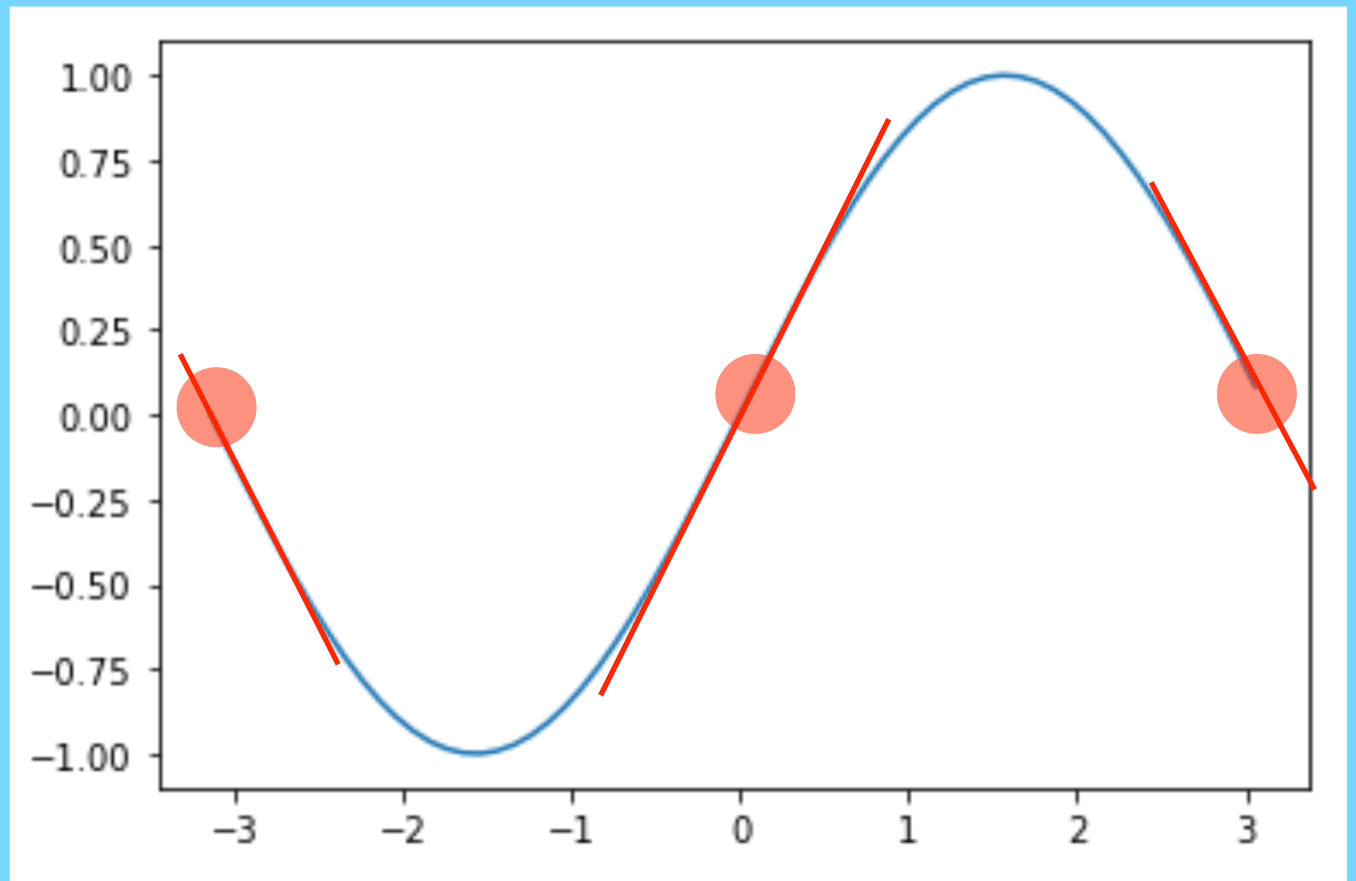
- Naive lattice fermion action suffers from fermion-doubling problem.

Discrete dispersion relation of non-interacting particles



Scalar

$$\sum_\mu \sin\left(\frac{p_\mu}{2}\right)^2 = 0$$



Fermion

$$\sum_\mu \sin\left(p_\mu\right)^2 = 0$$

Lattice Fermion Action

- Naive lattice fermion action suffers from fermion-doubling problem.

- Fermion doublers can be removed at the cost of giving up some properties of the continuum theory (Nielson-Ninomiya theorem), which can systematically be restored later.

- Wilson(-clover) fermions both isotropic and anisotropic - explicit breaking of chiral. sym. by the Wilson term, computational cost - moderate

$$\boxed{S_f = \sum_x \bar{\psi}(x)\left(\sum_\mu \gamma_\mu D_\mu + m - \frac{1}{2}\sum_\mu D_\mu^2\right)\psi(x)} \xrightarrow[\text{Dispersion relation}]{\text{Discretized}} \sum_\mu \sin\left(p_\mu\right)^2 \psi(x) + \left(4 - \sum_\mu \cos\left(p_\mu\right)\right)^2 = 0$$

$$S_f = a^4 \sum_n \bar{\psi}_n \psi_n - \kappa\left(\bar{\psi}_n(1-\gamma_\mu)U_{n,\hat{\mu}}\psi_{n+\hat{\mu}} + \bar{\psi}_n(1+\gamma_\mu)U^\dagger_{n-\hat{\mu},\hat{\mu}}\psi_{n-\hat{\mu}}\right) \quad \text{with} \quad \kappa = \frac{1}{8+2m}$$

(hopping parameter)

Lattice Fermion Action

- Naive lattice fermion action suffers from fermion-doubling problem.

- Fermion doublers can be removed at the cost of giving up some properties of the continuum theory (Nielson-Ninomiya theorem), which can systematically be restored later.

- Wilson(-clover) fermions both isotropic and anisotropic - explicit breaking of chiral. sym. by the Wilson term, computational cost - moderate

- Domain-wall fermions ((z-)Mobius) - chiral fermions, computational cost - expensive

Markov chain Monte Carlo

- Markov chain Monte Carlo algorithms are used to update link variables (gauge fields).

- In the case of pure gauge action, heatbath algorithms are typically used, while other algorithms including local-metropolis, Langevin, hybrid Monte Carlo (HMC) are also available.

$$\int d[U] f(U) = \int d[U] f(VU) = \int d[U] f(UV)$$

$$d[V(x) U_\mu(x) V(x + a\hat{\mu})^\dagger] = d[U_\mu(x)]$$

- Heatbath algorithms - explicit parameterization of Haar measure for $SU(2)$ in which $SU(2)$ matrices can be constructed with correct probability distribution. For $SU(N_c)$ we can use the heatbath algorithms for the $SU(2)$ subgroups to update $SU(N_c)$ matrices.

Markov chain Monte Carlo

- In the cases of a gauge theory coupled to dynamical fermions, we introduce pseudo-fermions ϕ (bosonic fields having the same quantum numbers).

$$Z = \int d[U] d\bar{\psi} d\psi e^{-S_G[U] - \sum_f \bar{\psi}_f D[U, m_f] \psi_f} = \int d[U] \left(\prod_f \det(D[U, m_f]) \right) e^{-S_G[U]}$$

$$\det(M) = \int d\phi d\phi^\dagger e^{-\phi^\dagger M^{-1} \phi}$$

positive definite

Markov chain Monte Carlo

- In the cases of a gauge theory coupled to dynamical fermions, we introduce pseudo-fermions ϕ (bosonic fields having the same quantum numbers).

For mass-degenerate two-flavors, $\det(D[U,m])^2 = \det(D[U,m]D[U,m]^\dagger)$

$$Z = \int d[U] \det(D[U,m])^2 e^{-S_G[U]} = \int d[U]d\phi d\phi^\dagger e^{-S_G[U]-\phi^\dagger(D[U,m]D[U,m]^\dagger)^{-1}\phi}$$

- We then update the gauge links U_μ using Molecular-Dynamics (MD) evolution for $H(\pi, U) = \frac{1}{2}\sum_{x,\mu} \pi(x,\mu)^2 + S_G(U) + S_{pf}(U)$ after introducing conjugate momentum $\pi(x,\mu) = i\pi^a(x,\mu)T^a$.

2. Measure physical quantities !!!

(Averaging over configurations)

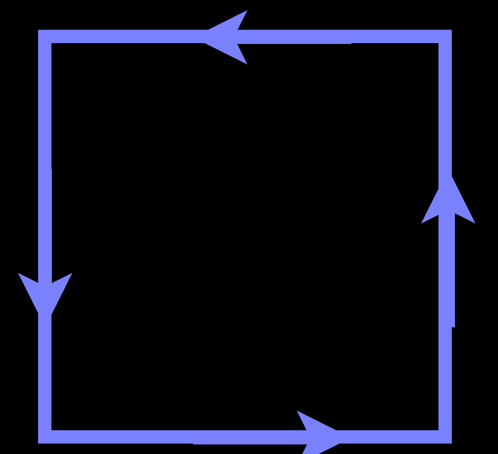
$$\langle \mathcal{O} \rangle = \frac{1}{N} \sum_{i}^{N} \mathcal{O}(U_i)$$

Measurements - Plaquettes and Polyakov loops

- Plaquette: the simplest gauge invariant object, the action density

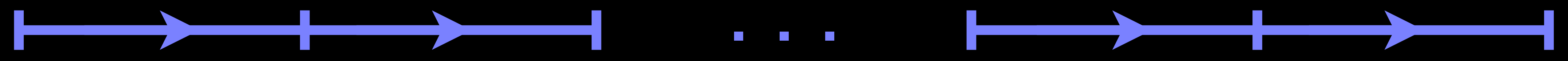
$$P_{\mu\nu}(x) = \frac{1}{N_c}\mathrm{Re}\,\mathrm{Tr}\,U_{\mu\nu}(x)$$

$$U_{\mu\nu}(x) = U_\mu(x)U_\nu(x+a\hat{\mu})U^\dagger_\mu(x+a\hat{\nu})U^\dagger_\nu(x)$$



- Polyakov loop: a trace of path-ordered products of link variables along the paths that wind around the lattice in a compactified direction, e.g. temporal, periodic b.c.

$$\Phi(T,\vec{x}) = \mathrm{Tr}\;\Pi_{t=0}^{T-1}\,U_0(t,\vec{x})$$



Measurements - interpolating operators of mesons

Interpolating operator ($\mathcal{O}_M$)	Meson	J^P
$\overline{Q^i}\gamma_5 Q^j$	π	0^-
$\overline{Q^i} Q^j$	a_0	0^+
$\overline{Q^i}\gamma_\mu Q^j$	ρ	1^-
$\overline{Q^i}\gamma_0\gamma_\mu Q^j$	ρ	1^-
$\overline{Q^i}\gamma_5\gamma_\mu Q^j$	a_1	1^+
$\overline{Q^i}\gamma_5\gamma_0\gamma_\mu Q^j$	b_1	1^+

Measurements - meson 2-point correlation functions

- Quark propagator

$$\langle \psi(x)_i \bar{\psi}(y)_j \rangle = \langle D^{-1}(U)_{i,x;j,y} \rangle$$

- Meson propagator

Performe Wick contraction

$$\langle \psi(x_1)_{i_1} \bar{\psi}(y_1)_{j_1} \psi(x_2)_{i_2} \bar{\psi}(y_2)_{j_2} \rangle = \langle D^{-1}(U)_{i_1,x_1;j_1,y_1} D^{-1}(U)_{i_2,x_2;j_2,y_2} \rangle - \langle D^{-1}(U)_{i_1,x_1;j_2,y_2} D^{-1}(U)_{i_2,x_2;j_1,y_1} \rangle$$

Consider an interpolating operator of $\pi^{\pm}$ $\quad \hat{O}_\pi(t) = i \sum_{\vec{x}} \hat{\bar{u}}(\vec{x},t)\gamma_5 \hat{d}\,(\vec{x},t)$

Then, the zero-momentum 2-point correlation function becomes

$$\begin{aligned} C(t) &= \langle \hat{O}_\pi(t) \hat{O}^\dagger_\pi(0) \rangle \\ &= \sum_{\vec{x},\vec{y}} \langle \mathrm{Tr}[D^{-1}(U)_{\vec{x},t;\vec{y},0} \gamma_5 D^{-1}(U)_{\vec{y},0;\vec{x},t} \gamma_5] \rangle \end{aligned}$$

Analysis - extraction of the ground state energy (mass)

- At large Euclidean time, the correlation function behaves as

$$C_{\mathcal{O}_M}(t) \xrightarrow{t\to\infty} \langle 0|\mathcal{O}_M|M\rangle\langle 0|\mathcal{O}_M|M\rangle^* \frac{1}{2m_M}\left[e^{-m_M t} + e^{-m_M(T-t)}\right]$$

- One can extract the mass by fitting the data to the single exponential function over the range of late time, showing a plateau in the effective mass defined by

$$m_{\text{eff}}(t) = \operatorname{arccos}\left(\frac{C(t) + C(t+2)}{2C(t+1)}\right)$$

- Other physical observables can be measured in a similar way.

3. Analyze lattice results with EFT

(continuum & physical-mass extrapolation)

Have fun!