Lecture 2 Homework:
Simulating 1+1D Schwinger Model in LQCD

Aaron S. Meyer (asmeyer.physics@gmail.com)

Lawrence Livermore National Laboratory

June 6, 2024

NuSTEC 2024 Summer School — CERN

This work is supported in part by:
Lawrence Livermore National Security, LLC #DE-AC52-07NA27344,
Neutrino Theory Network Program Grant #DE-AC02-07CHI11359,
U.S. Department of Energy Award #DE-SC0020250.

Exercise 1

Time to try a LQCD computation! We'll use the 1+1D Schwinger Model (see [Mod.Phys.Lett.A 16 (2001)])

I'm assuming you're using a unix system (Mac/Linux).
If you need help with Windows, let me know and I'll attempt to help.

1. Download and unpack the LQCD physics package from Indico.
   The command to unpack the repository is:
   `tar -xzvf lqcd_physics.tar.gz`
2. Take a look at the file `physics_workflow`. We'll examine some of these steps in a bit more detail along the way.
3. Try running `physics_workflow` with the command:
   `source physics_workflow`
   This takes about 3 minutes to run on my system.
4. The script produces "pion" correlation functions, computes an error, and prints.
   - What do you notice about the time dependence of the correlator?
   - What is the physics reason for that behavior?
5. Try decreasing the number of correlators that are loaded in `analyze_correlators.py`.
   - Does the error scale with the usual $1/\sqrt{N_{\text{sample}}}$ statistical uncertainty?
   - Why or why not?

Exercise 2

Save your correlators along the way and keep track of the parameters for each ensemble (so you don't have to recompute later!).

1. Try changing some of the parameters at the top of the file in **`physics_workflow`**.
   (The algorithm is not optimized, so it will scale quite poorly with increasing the computation intensity! Watch out!)
   - What kinds of effects do you notice?
     Pay attention to computing time and to uncertainties on the final correlators.
2. Pick one of your favorite ensembles and try to extract the pion mass.
   Feel free to use the ensemble provided on the Indico page, **`gauge1016.tar.gz`** ($L \times T = 10 \times 16$)

   The fit function to use is

   $C(t) = \sum_n |z_n|^2 \left(e^{-E_n t} + e^{-E_n (T-t)}\right)$, where T is the **`NT`** parameter for that ensemble.

   If your time range is long enough, you could try extracting the effective mass

   $E_{\text{eff}} = -\log\left[\frac{C(t+1)}{C(t)}\right]$.
   - How do you derive the effective mass relation?
3. If you can, try extracting the pion mass for multiple ensembles.
   - How does your pion mass change with the ensemble parameters?

Exercise 3

We've been working with a "pion" so far. In the Schwinger model, this is really a bound state of e^+e^-. We can also try a "nucleon" operator, a bound state of e^-e^-, where α and β index the spin of the electron,

$$\begin{aligned}\mathcal{O}(x) &= \frac{1}{2}\epsilon^{\alpha\beta}\psi_\alpha(x)\psi_\beta(x) \\ &= \frac{1}{2}\Big(\psi_0(x)\psi_1(x) - \psi_1(x)\psi_0(x)\Big) = \psi_0(x)\psi_1(x)\end{aligned}$$

1. Look at the function `compute_pion_correlator` in `compute_correlators.py` and try to code up a nucleon 2-point function.

   Pro tip: there are two combinations of propagators that need to be computed. Be careful about relative signs!

2. Compute your nucleon correlation functions.
   - What does the uncertainty look like for your nucleon correlators?
   - Can you extract a mass for your nucleon?

3. How would you go about computing a 3-point correlation function, for example the pseudoscalar current $\mathcal{O}(x) = \bar{\psi}(x)\gamma_5\psi(x)$?

   Use what you've learned about computing correlation functions to design the computation.