Key Performance Indicator for Topological Charge Diffusion on Lattice

Jiqun Tu

1 Introduction

This is a manual of the package for measuring topological charge diffusion in the way described in [1]. We assume that the potential users have basic understanding of `bash` system and lattice QCD calculation. Statistical techniques used are discussed in the appendix since they are only briefly mentioned in the paper.

2 Package

The package is available at `git@github.com:hummingtree/DBW2-KPI.git`. In order to run wilson flow and measure topological charge `CPS`(Columbia Physics System) is needed: `git@github.com:hummingtree/cps-public.git`.

3 Install

3.1 CPS

We will need `gcc` and `MPI` to compile `CPS`. Make sure your `$PATH` includes `MPI` before compling.

1. Clone `CPS` from the git repository. Through out this document we will call this directory `CPS_REPO`.
2. Go to the top of the repository, `CPS_REPO`. `CPS` will be installed and placed under a directory specified as a variable `$cps`(line 4) in `CPS_REPO/conf.sh` so change that if you want. The default directory is `../cps-build/public/`. Through out this document we will call this directory `CPS_BUILD`.
3. Execute the `build.sh` script.

3.2 Analysis

We will need `g++`, `python`(version `2.7.5` tested) and `matlab`(version `R2013a (8.1.0.604)64-bit (glnxa64)` tested), later versions should be fine) to perform the whole analysis. Make sure these softwares are included in your `$PATH`.

1. Clone DBW2-KPI from the git repository. Through out this document we will call this directory DBW2-KPI_REPO.

2. Copy the file CPS_BUILD/build-qmp/Makefile.users to both DBW2-KPI_REPO/ens_gen/ and DBW2-KPI_REPO/wflow_tcharge_cps/ as makefile, i.e.

```
$ cp CPS_BUILD/build-qmp/Makefile.users DBW2-KPI_REPO/ens_gen/makefile
$ cp CPS_BUILD/build-qmp/Makefile.users DBW2-KPI_REPO/wflow_tcharge_cps/
    makefile
```

4 Usage

There are two stages when using this package. In the first stage CPS reads in the lattice configurations in standard *nersc* format, run wilson flow and measure the topological charge density. The second stage process the raw data produced by CPS and perform the fit.

4.1 First Stage

1. Go to DBW2-KPI_REPO/wflow_tcharge_cps/binaries.

2. Information of the incoming configurations are supposed to be given from line 100 to line 119. You might want to change these information to match your configurations. The comment there should be quite self-explaining.

3. make.

4. Use mpirun -np $num_node NOARCH.x -qmp-geom $x $y $z $t to run the job. You should change $num_node to match your environment. $x, $y, $z and $t are number of nodes to be allocated in each dimensions. The product of these four should be equal to $num_node. Examples are

```
mpirun -np 1 NOARCH.x -qmp-geom 1 1 1 1
mpirun -np 256 NOARCH.x -qmp-geom 4 4 4 4
```

5. The results will be placed under DBW2-KPI_REPO/wflow_tcharge_cps/results/alg_wflow/. The files would look like:

```
   2.131295e+00
   1.700994e+00    2.087967e+00    2.171048e+00    2.162269e+00    ...
AlgTcharge:
nleaf : 5
   0 : 1x1
   1 : 1x2
   2 : 2x2
   3 : 3x3
   4 : 1x3
0 0 :    6.235056e-01
```

```
11 1 1 :    4.098351e-01
12 2 2 :    1.044206e-01
13 3 3 :    7.561517e-03
14 4 4 :    2.090085e-01
15 0 :    2.519808e-01   -1.512013e-01   -2.445761e-01    1.517083e-02    ...
16 1 :    8.713487e-02   -9.518725e-03   -2.284069e-02    3.067389e-03    ...
17 2 :    4.226348e-03   -4.861770e-03    9.862970e-03   -1.161517e-04    ...
18 3 :   -1.795864e-03    2.317363e-03   -9.817474e-04    1.049183e-03    ...
19 4 :    2.337879e-02    1.922419e-02    2.056840e-02   -6.668813e-03    ...
20 AlgWilsonFlow: dt = 5.000000e-02
```

The "5 Li" method can be found in [2]. Here in the outputfile line 5 shows number of leaves we use to calculate the topological charge density, line 5 through 9 show the size of the leaves, line 10 through 14 show the values of the different pieces of the total topological charge. line 15 through 19 show the same number but local on different time slices.

4.2 Second Stage

1. Go to `DBW2-KPI_REPO`.
2. The input information of this stage is specified in `DBW2-KPI_REPO/conf.sh`. Specify all the information in that file.
3. execute `DBW2-KPI_REPO/do.sh`.
4. The final result will be placed under `DBW2-KPI_REPO/correlation_to_fit/results/`.

4.3 Ensemble Generation

`DBW2-KPI_REPO/ens_gen/binaries` contains the code to generate open/periodic boundary condition pure gauge lattice with DBW2 action. Before compiling you should specify the information of the lattice you want to generate:

1. Lattice size is specified as `int x_sites`, `int y_sites`, `int z_sites` and `int t_sites` in `DBW2-KPI_REPO/ens_gen/vmls/do_arg.vml`.
2. t-direction boundary condition is specsified at line 94 of `DBW2-KPI_REPO/ens_gen/binaries/main.C`.
3. Number of steps in one trajectory and step size are specified as `int steps_per_traj` and `double step_size` in `DBW2-KPI_REPO/ens_gen/vmls/hmc_arg.vml`.
4. β and c_1 of the gauge action are specified as `double beta` and `c_1` in `DBW2-KPI_REPO/ens_gen/vmls/do_arg.vml`. The default value is $c_1 = -1.4088$ for DBW2 action.
5. You are probably safe to use all the other default values.

Now just go to `DBW2-KPI_REPO/ens_gen/binaries`, do `make` and execute the `NOARCH.x` in the same way mentioned in the first stage.

The topological charge density data is automatically generated and placed under `DBW2-KPI_REPO/ens_gen/results/alg_wflow/`. From here you can directly go to the second stage.

Appendices

A Statistics

A.1 Jackknife Method[3]

Let $[X_1, \cdots, X_n]$ be n (*independent and identically distributed*) random variables. We split this sample into g groups of size h each, $n = gh$[1]. Let T be an estimator of some parameter θ based on sample size n, T_{-i} be the corresponding estimator based on the sample of size $(g-1)h$, where the i-th group of size h has been excluded:

$$T = f[X_1, \cdots, X_n],$$
$$T_{-i} = f[X_1, \cdots, X_{(i-1)h}, X_{ih+1}, \cdots, X_n].$$

Write $\bar{T}_\bullet$ as the mean of the T_{-i}'s, the jackknife estimator is defined as

$$T_j = gT - (g-1)\bar{T}_\bullet,\ \bar{T}_\bullet = \langle T_{-i}\rangle_i = \frac{1}{g}\sum_{i=1}^{g} T_{-i},$$

and the estimator of the variance of the T_j is

$$\mathrm{var}[T_j] = \frac{g-1}{g}\sum_{i=1}^{g}(T_{-i} - \bar{T}_\bullet)^2.$$

Usually we use T instead of T_j, since they have the same expectation value.

A.2 Variance of the Correlation Function

A.2.1 Open Boundary Condition

In [1] the correlation function on lattices with open boundary condition is defined as

$$C(t, t_0, \tau) \equiv \langle Q(t, \tau_0 + \tau)Q(t_0, \tau_0)\rangle.$$

A natural estimator is:

$$\hat{C}(t, t_0, \tau) = \langle Q(t, \tau_0 + \tau)Q(t_0, \tau_0)\rangle_{\tau_0}.$$

Variance of $\hat{C}(t, t_0, \tau)$ is roughly[2]

$$\mathrm{var}[\hat{C}(t, t_0, \tau)] = \langle C(t, t, \tau_0)C(t_0, t_0, \tau_0) + C(t, t_0, \tau_0 + \tau)C(t_0, t, \tau_0 - \tau)\rangle_{\tau_0}.$$

[1] If h is longer than the autocorrelation time of the random variables it seems the requirement that the random variables be *independent and identically distributed* can be skipped.

[2] This is true if mean value of $Q(t, t_0, \tau)$ is zero and its the fourth joint cumulant κ_4 is also zero.[4] The mean value of the topological charge density is zero while we are assuming κ_4 to be zero.

A.2.2 Periodic Boundary Condition

Correlation function on lattices with periodic boundary condition is simplified as

$$C(\Delta t, \tau) \equiv \langle Q(t_0, \tau_0 + \tau) Q(t_0 + \Delta t, \tau_0) \rangle.$$

A natural estimator is:

$$\hat{C}(\Delta t, \tau) \equiv \langle Q(t_0, \tau_0 + \tau) Q(t_0 + \Delta t, \tau_0) \rangle_{\tau_0, t_0}.$$

The corresponding variance is roughly

$$\mathrm{var}[\hat{C}(\Delta t, \tau)] = \left\langle C(t_0, \tau_0)^2 + C(t_0 + t, \tau_0 + \tau) C(t_0 - t, \tau_0 - \tau) \right\rangle_{\tau_0, t_0}.$$

References

[1] G. McGlynn and R. D. Mawhinney, Physical Review D - Particles, Fields, Gravitation and Cosmology **90**, 1 (2014), arXiv:1406.4551 .

[2] P. de Forcrand, M. G. Perez, and I.-O. Stamatescu, (1997), 10.1016/S0550-3213(97)00275-7, arXiv:9701012 [hep-lat] .

[3] R. G. Miller, Biometrika **61**, 1 (1974).

[4] N. Madras and a. D. Sokal, J. Stat. Phys. **50**, 109 (1988).