FT-HMC code for 2D $U(1)$

Nobuyuki Matsumoto*

Abstract

Brief description of the code. Many features mimic the qlat [1].

*E-mail address: nobuyuki.matsumoto@riken.jp

Contents

1. Physical system

Path integral

$U(1)$ lattice gauge theory on T^2:

$$Z_{\beta,\theta} \equiv \int (dU)\, e^{-S(U)+i\theta Q(U)}, \tag{1.1}$$

where $U_{x,\mu}$ are $U(1)$-valued link variables (see figure 1):

$$U_{x,\mu} = e^{i\phi_{x,\mu}} \tag{1.2}$$

and (dU) the Haar measure:

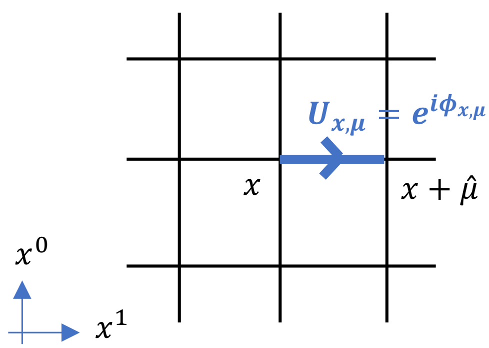

Figure 1: 2D $U(1)$ lattice.

$$(dU) = \prod_{x,\mu} \left(\frac{d\phi_{x,\mu}}{2\pi}\right). \tag{1.3}$$

We choose as $S(U)$ the Wilson action [2]:

$$S(U) \equiv -\beta \sum_x \cos \kappa_x \tag{1.4}$$

and $Q(U)$ the integer-valued topological charge [3,4]:

$$Q(U) \equiv \frac{1}{2\pi} \sum_x \kappa_x, \tag{1.5}$$

where κ_x is the plaquette angle:

$$\begin{aligned}\kappa_x &\equiv \frac{1}{i}\ln\left(U_{x,0}U_{x+0,1}U^\dagger_{x+1,0}U^\dagger_{x,1}\right)\\ &= \phi_{x,0}+\phi_{x+0,1}-\phi_{x+1,0}-\phi_{x,1} \pmod{2\pi},\end{aligned} \tag{1.6}$$

where it is assumed that we take the principal branch of the logarithm:

$$\kappa_x \in [-\pi,\pi). \tag{1.7}$$

Classically, the gauge potential $A_{x,\mu}$ in the continuum is defined as:

$$\phi_{x,\mu} = aA_{x,\mu}, \tag{1.8}$$

where a is the lattice spacing. By scaling β as:

$$\beta = \frac{1}{(ag)^2}, \tag{1.9}$$

we have

$$S(U) \sim \frac{a^2}{2g^2}\sum_x F_{01}^2 \sim \frac{1}{4g^2}\int d^2x\, F_{\mu\nu}^2 \tag{1.10}$$

and

$$Q(U) \sim \frac{1}{2\pi}a^2\sum_x F_{01} \sim \frac{1}{2\pi}\int d^2x\, F_{01}, \tag{1.11}$$

where

$$F_{\mu\nu} \equiv \partial_\mu A_\nu - \partial_\nu A_\mu. \tag{1.12}$$

Exact solutions

We compute the partition function:

$$Z_{\beta,\theta} = \int (dU)\prod_x e^{\beta\cos\kappa_x + \frac{i\theta}{2\pi}\kappa_x}. \tag{1.13}$$

We can use the Fourier expansion:

$$e^{\beta\cos\kappa + \frac{i\theta}{2\pi}\kappa} = \sum_{n\in\mathbb{Z}}\lambda_n(\beta,\theta)e^{in\kappa}, \tag{1.14}$$

where

$$\lambda_n(\beta,\theta) \equiv \int_{-\pi}^{\pi}\frac{d\kappa}{2\pi}\, e^{-i\nu\kappa}e^{\beta\cos\kappa}\Big|_{\nu=n-\theta/2\pi} \tag{1.15}$$

$$\sim \frac{e^\beta}{\sqrt{2\pi\beta}}\sum_{k\geq 0}(\nu,k)\left(\frac{-1}{2\beta}\right)^k\Big|_{\nu=n-\theta/2\pi} \tag{1.16}$$

with

$$(\nu, k) \equiv \frac{\Gamma(\nu + k + 1/2)}{k!\,\Gamma(\nu - k + 1/2)} = \frac{\{4\nu^2 - 1^2\}\{4\nu^2 - 3^2\}\cdots\{4\nu^2 - (2k-1)^2\}}{2^{2k}\cdot k!}. \tag{1.17}$$

For $\theta = 0$, $\lambda_n(\beta, \theta)$ is identical to the modified Bessel function $I_\nu(\beta)$, but they are different generically.[1] We now have

$$Z_{\beta,\theta} = \int (dU) \prod_x \sum_{n_x} \lambda_{n_x}(\beta,\theta) e^{i n_x \kappa_x} = \sum_n \lambda_n(\beta,\theta)^{\hat{V}}, \tag{1.18}$$

where $\hat{V}$ is the lattice volume (in the lattice units).

We compute some expectation values for the case $\theta = 0$. The average plaquette can be calculated as

$$\begin{aligned} \langle \cos\kappa_x \rangle &= \frac{1}{\hat{V}} \frac{d}{d\beta} \ln Z_{\beta,\theta=0} \\ &= \frac{\sum_n \left[(I_{n+1}(\beta) + I_{n-1}(\beta))/2 \right] \cdot I_n(\beta)^{\hat{V}-1}}{\sum_n I_n(\beta)^{\hat{V}}} \\ &\to \frac{I_1(\beta)}{I_0(\beta)} \sim \left(1 - \frac{1}{2\beta}\right) \quad (\beta \to \infty), \end{aligned} \tag{1.19}$$

where we used that $(0,0) = (1,0) = 1$, $(0,1) = -1/4$ and $(1,1) = 3/4$. We define the average energy density by:

$$e \equiv \frac{1}{a^4 \hat{V}} \sum_x (1 - \cos\kappa_x) \to \frac{1}{2V} \int d^2x\, F_{01}^2 \quad (a \to 0), \tag{1.20}$$

where $V \equiv a^2 \hat{V}$ is the physical volume of the system. This is actually a divergent quantity as can be seen from that:

$$\langle e \rangle \sim \frac{g^2}{2a^2}. \tag{1.21}$$

It is also notable that there is no correlation length in this system, which can be seen from the above calculation by modifying β to β_x and by taking the derivatives with respect to β_x:

$$\langle \cos^2\kappa_x \rangle = \frac{\sum_n \left[(I_{n+2}(\beta) + I_n(\beta)) + I_{n-2}(\beta))/4 \right] \cdot I_n(\beta)^{\hat{V}-1}}{\sum_n I_n(\beta)^{\hat{V}}}, \tag{1.22}$$

$$\langle \cos\kappa_x \cos\kappa_y \rangle = \frac{\sum_n \left[(I_{n+1}(\beta) + I_{n-1}(\beta))^2/4 \right] \cdot I_n(\beta)^{\hat{V}-2}}{\sum_n I_n(\beta)^{\hat{V}}} \quad (x \neq y). \tag{1.23}$$

[1] The integral (1.15) is an even function of ν, but $I_\nu(\beta)$ is not.

We also calculate the susceptibility:

$$
\begin{aligned}
\chi_Q &\equiv \frac{1}{V}\langle Q^2\rangle \\
&= -\frac{1}{V}\frac{d^2}{d\theta^2}\ln Z_{\beta,\theta}\Big|_{\theta=0} \\
&= -\frac{1}{V}\frac{1}{Z_{\beta,\theta}}\hat{V}\sum_n \lambda_n^{\hat{V}-1}\cdot\frac{d^2\lambda_n}{d\theta^2}\Big|_{\theta=0} \\
&\sim -\frac{1}{a^2}\frac{1}{\lambda_0(\beta,\theta)}\frac{d^2\lambda_0}{d\theta^2}\Big|_{\theta=0},
\end{aligned}
\tag{1.24}
$$

where we used the fact $(d\lambda_n/d\theta)|_{\theta=0} = 0$. To get the continuum limit, we take the derivative of eq. (1.16) with respect to ν. Using the polygamma function:

$$
\psi^{(m)}(z) \equiv \frac{d^m}{dz^m}\ln\Gamma(z),
\tag{1.25}
$$

we have:

$$
(2\pi)\frac{d}{d\theta}\lambda_n(\beta,\theta) \sim \frac{e^\beta}{\sqrt{2\pi\beta}}\sum_{k\geq 0}(\nu,k)A(\nu,k)\left(\frac{-1}{2\beta}\right)^k\Big|_{\nu=n-\theta/2\pi},
\tag{1.26}
$$

$$
(2\pi)^2\frac{d^2}{d\theta^2}\lambda_n(\beta,\theta) \sim \frac{e^\beta}{\sqrt{2\pi\beta}}\sum_{k\geq 0}(\nu,k)\left\{A(\nu,k)^2 + B(\nu,k)\right\}\left(\frac{-1}{2\beta}\right)^k\Big|_{\nu=n-\theta/2\pi}
\tag{1.27}
$$

with

$$
A(\nu,k) \equiv \psi^{(0)}(\nu+k+1/2) - \psi^{(0)}(\nu+k-1/2),
\tag{1.28}
$$

$$
B(\nu,k) \equiv \psi^{(1)}(\nu+k+1/2) - \psi^{(1)}(\nu+k-1/2).
\tag{1.29}
$$

Therefore,

$$
\begin{aligned}
\chi_Q &\sim -\frac{1}{a^2}\frac{1}{I_0(\beta)}\frac{d^2\lambda_0}{d\theta^2}\Big|_{\theta=0} \\
&\sim -\frac{1}{a^2}\cdot 1\cdot\left(\frac{-1}{4}\right)\cdot(-8)\cdot\frac{1}{(2\pi)^2}\cdot\left(\frac{-1}{2\beta}\right) = \frac{g^2}{(2\pi)^2},
\end{aligned}
\tag{1.30}
$$

where we used that $A(0,0) = A(0,1) = 0$, $B(0,0) = 0$ and $B(0,1) = -8$. The susceptibility has a finite value in the continuum limit.

2. Field-transformed HMC

The contents expressed here was first developed by [5] for $SU(3)$ on T^4.

<u>Transformation</u>

$$
U_{x,\mu} \to U_{x,\mu}^{(\epsilon)} \equiv \mathcal{F}_\epsilon(U)_{x,\mu} \equiv e^{\epsilon\partial_{x,\mu}K(U)}U_{x,\mu}.
\tag{2.1}
$$

We choose the kernel $K(U)$ to be the plaquettes:

$$K(U) = \sum_x \cos \kappa_x. \tag{2.2}$$

We recursively define for $\ell \geq 1$:

$$S_{\ell\epsilon}(U) \equiv S_{(\ell-1)\epsilon}(\mathcal{F}_\epsilon(U)) - \ln \det \mathcal{F}_{\epsilon *}(U), \tag{2.3}$$

where $S_0(U) = S(U)$ and $\mathcal{F}_{\epsilon *}(U)$ is the Jacobian of the transformation (2.1). By acting $d \equiv \sum_{x,\mu} d\phi_{x,\mu}\partial_{x,\mu}$ to eq. (2.1), we find

$$\mathcal{F}_{\epsilon *}(U)_{x,\mu|y,\nu} = \delta_{xy}\delta_{\mu\nu} + \epsilon\partial_{y,\nu}\partial_{x,\mu}K(U). \tag{2.4}$$

To the leading order in ϵ,

$$S_{\ell\epsilon}(U) = S_{(\ell-1)\epsilon}(U) + \epsilon \sum_{x,\mu} \partial_{x,\mu}K(U)\partial_{x,\mu}S(U) - \epsilon \sum_{x,\mu} \partial^2_{x,\mu}K(U). \tag{2.5}$$

Explicitly,

$$S_{\ell\epsilon}(U) = S_{(\ell-1)\epsilon}(U) + \epsilon \sum_x \Big[4\beta \sin^2 \kappa_x - \beta \sum_{\mu=\pm 0,\pm 1} \sin \kappa_{x+\mu} \sin \kappa_x + 4 \cos \kappa_x \Big]. \tag{2.6}$$

Note that, from eq. (2.4),

$$\partial_{y,\nu} = \partial^{(\epsilon)}_{x,\mu} \mathcal{F}_{\epsilon *}(U)_{x,\mu|y,\nu}. \tag{2.7}$$

Therefore, from eq. (2.3),

$$\partial_{y,\nu}S_{\ell\epsilon}(U) = \sum_{x,\mu} \big[\delta_{xy}\delta_{\mu\nu} + \epsilon\partial_{y,\nu}\partial_{x,\mu}K(U)\big] \partial^{(\epsilon)}_{x,\mu} S_{(\ell-1)\epsilon}(U^{(\epsilon)}) - \operatorname{tr}\big[\mathcal{F}_{\epsilon *}(U)^{-1}\partial_{y,\nu}\mathcal{F}_{\epsilon *}(U)\big]. \tag{2.8}$$

In practice, we deal with finite ϵ. To ensure that the map is invertible, we perform the even/odd partitioning. In terms of the angle variables, the transformation can be rewritten as:

$$\phi^{(\epsilon)}_{x,\mu} = \phi_{x,\mu} - \epsilon(\sin \kappa_{x'} - \sin \kappa_{x''}), \tag{2.9}$$

where

$$x' = x,\ x'' = x - 1 \quad (\mu = 0) \tag{2.10}$$

$$x' = x - 0,\ x'' = x \quad (\mu = 1). \tag{2.11}$$

To invert eq. (2.9), we rewrite the equation as:

$$f(X) = Z_{x,\mu}(\phi)|_{\phi=\phi^{(\epsilon)}-\epsilon X}, \quad X = f(X), \tag{2.12}$$

where

$$Z_{x,\mu}(\phi) \equiv \partial_{x,\mu} K(U) = -\sin\kappa_{x'} + \sin\kappa_{x''}. \tag{2.13}$$

Then, for $\epsilon < 1/2$ the map f defines the contraction mapping because:

$$\begin{aligned} |f(X) - f(Y)| &= |-\sin(\varphi + \epsilon X) + \sin(\varphi' - \epsilon X) + \sin(\varphi + \epsilon Y) - \sin(\varphi' - \epsilon Y)| \\ &= \Big| -2\cos\Big(\varphi + \frac{\epsilon}{2}(X-Y)\Big)\sin\Big(\frac{\epsilon}{2}(X-Y)\Big) \\ &\qquad + 2\cos\Big(\varphi' + \frac{\epsilon}{2}(X-Y)\Big)\sin\Big(\frac{\epsilon}{2}(X-Y)\Big)\Big| \qquad (2.14) \\ &\le 2\epsilon|X-Y|. \qquad (2.15) \end{aligned}$$

3. Code

The code consists of the `main.c` and the following header files:

util.h some utility functions (print functions for vectors, projection from $\mathbb{R}$ to $U(1)$).

lattice.h defines the `Lattice` class, which describes the geometry of the lattice.

coord.h defines the `Coord` class, which describes a point x on the lattice.

scalar_field.h defines the `ScalarField` class, which describes the field $\phi_{x,\mu}$ on the lattice. We use it for the angular variables.

action.h defines the `WilsonAction` class, which describes the action function S.

corr.h defines the `Corr` class, which gives the correlator functions $\cos\kappa_x \cdot \cos\kappa_y$.

rnd.h defines the `Rnd` class, which gives a set of mt19937 generators on each site or link.

hmc.h defines the `HMC` class, which gives the *ordinary* HMC algorithm.

kernel.h defines the `Kernel` class, which describes the kernel K for the field transformation.

field_trsf.h defines the `FieldTrsf` class, which describes the infinitesimal map $\mathcal{F}_\epsilon$.

ft_hmc.h defines the `FT_HMC` class, which gives the FT-HMC algorithm.

The present code calculates and records the

- average plaquette
- two-point functions

- topological charge.

The plaquette and the topological charge are recorded for both the physical field U and the auxiliary field V.

The code is aimed to be run on laptops and uses `c++17` for `filesystem`. The remaining part only uses `c++11`. It can be run with openmp. No MPI calls. It generates `500 configurations` with one-step Wilson-flowed HMC in `30 sec` for 8×8 lattice (without OpenMP). It creates `./result/` directory, in which the observable data are stored, and a log file. The contents written in this paragraph are mostly described in `main.cpp`.

One can compile and run the ordinary HMC code with `make; make run`. To switch to the FT-HMC, one can use `make IS_FTHMC=1; make run`. The included `analysis.ipynb` can be used to analyze observable data. The exact solutions (or their approximated functions) are written in the python notebook for checking.

References

[1] L. Jin, "Qlattice," https://github.com/jinluchang/Qlattice

[2] K. G. Wilson, "Confinement of Quarks," Phys. Rev. D **10**, 2445-2459 (1974) doi:10.1103/PhysRevD.10.2445

[3] M. Luscher, "Topology of Lattice Gauge Fields," Commun. Math. Phys. **85**, 39 (1982) doi:10.1007/BF02029132

[4] A. Phillips, "Characteristic Numbers of U(1) Valued Lattice Gauge Fields," Annals Phys. **161**, 399-422 (1985) doi:10.1016/0003-4916(85)90086-7

[5] M. Luscher, "Trivializing maps, the Wilson flow and the HMC algorithm," Commun. Math. Phys. **293**, 899-919 (2010) doi:10.1007/s00220-009-0953-7 [arXiv:0907.5491 [hep-lat]].