U(1) Fourier acceleration

Agostino Patella, Nazario Tantalo, RC* collaboration

December 2017

Contents

1 Introduction

The openQ*D code supports the Fourier acceleration for the U(1) molecular-dynamics equations [1]. This feature can be typically enabled by setting to 1 the value of the facc parameter in the [HMC parameters] section in the input file.

2 Molecular dynamics

The molecular-dynamics Hamiltonian with Fourier acceleration is defined to be

$$H = \frac{1}{2}(\pi, D^{-1}\pi)_{\mathrm{U}(1)} + \frac{1}{2}(\Pi, \Pi)_{\mathrm{SU}(3)} + S(U, A) \ , \tag{1}$$

where π and Π are the momenta conjugated to A and U respectively. The scalar products are defined as

$$(\Pi, \Pi)_{\mathrm{SU}(3)} = \omega_{\mathrm{C}^\star} \sum_{x,\mu,a} [\Pi^a(x, \mu)]^2 \ , \tag{2}$$

$$(\phi, \phi)_{\mathrm{U}(1)} = \omega_{\mathrm{C}^\star} \sum_{x,\mu} [\phi(x, \mu)]^2 \ . \tag{3}$$

The overall weight $\omega_{\mathrm{C}^\star}$ is 1 if no $\mathrm{C}^\star$ boundary conditions are used, and 1/2 otherwise because of the orbifold construction (see [2] for more details).

The operator $(-D)$ is a discretization of the Laplacian defined with appropriate boundary conditions. If the boundary conditions allow for zero modes, the action of the operator D is set to the identity on the zero mode by hand. The operator D is strictly positive, real and symmetric. The explicit definition of D will be given in sec. 3.

In order to generate the momentum field π with the correct distribution, one can generate a field ϕ with distribution $\propto \exp\{-\frac{1}{2}(\phi, \phi)_{\mathrm{U}(1)}\}$ first, and then define

$$\pi = D^{1/2}\phi \ . \tag{4}$$

The evolution equation for the U(1) gauge field with Fourier acceleration becomes

$$\partial_t A(x, \mu) = D^{-1}\pi(x, \mu) \ , \tag{5}$$

while the other fields are unaffected. The other equations are discussed in detail in [3].

3 Laplacian

The U(1) momentum is generally represented in momentum space as

$$\pi(x,\mu) = \frac{1}{L^3} \sum_{p_0 \in E_\mu} \sum_{\mathbf{p} \in \mathcal{P}} e_\mu(p_0,x_0) e^{i\mathbf{px}} \tilde{\pi}(p,\mu) \ . \tag{6}$$

The basis functions $e_\mu(p_0,x_0)$ (for fixed μ) are orthogonal with respect to a weighted scalar product

$$\sum_{x_0} w_\mu(x_0) e^*_\mu(p_0,x_0) e_\mu(q_0,x_0) = \delta_{p_0,q_0} \ , \tag{7}$$

where the weight $w_\mu(x)$ is taken to be 1/2 if x belongs to an open boundary (i.e. $x_0 = 0$ for open and open-SF b.c.s, and $x_0 = T-1$ for open b.c.s) and $\mu = 1,2,3$. In all other cases $w_\mu(x)$ is taken to be 1. The relation between π and $\tilde{\pi}$ is easily inverted

$$\tilde{\pi}(p,\mu) = \sum_x w_\mu(x_0) e^*_\mu(p_0,x_0) e^{-i\mathbf{px}} \pi(x,\mu) \ . \tag{8}$$

The set $\mathcal{P}$ is given by all spatial momenta $\mathbf{p} = (p_1,p_2,p_3)$ of the form

$$p_k = \tfrac{\pi}{L_k}(2n_k + c_k) \quad \text{with } n_k = 0,\ldots,L_k - 1 \ , \tag{9}$$

where $c_k = 0$ if k is a periodic direction and $c_k = 1$ if k is a C* direction. The sets E_μ and the eigenfunctions $e_\mu(p_0,x_0)$ depend on the boundary conditions in time. In the following $k = 1,2,3$.

- Open boundary conditions:

$$E_0 = \frac{\pi}{N_0 - 1}\{1,\ldots,N_0-1\} \ , \qquad E_k = \frac{\pi}{N_0-1}\{0,\ldots,N_0-1\} \ , \tag{10}$$

$$e_0(p_0,x_0) = \frac{i}{(1+\delta_{p_0,\pi})(N_0-1)} \sin\left[p_0\left(x_0 + \frac{1}{2}\right)\right] \ , \tag{11}$$

$$e_k(p_0,x_0) = \frac{1}{(1+\delta_{p_0,0}+\delta_{p_0,\pi})(N_0-1)} \cos(p_0 x_0) \ . \tag{12}$$

- SF boundary conditions:

$$E_0 = \frac{\pi}{N_0}\{0,\dots,N_0-1\} \ , \qquad E_k = \frac{\pi}{N_0}\{1,\dots,N_0-1\} \ , \tag{13}$$

$$e_0(p_0,x_0) = \frac{1}{(1+\delta_{p_0,\pi})N_0}\cos\left[p_0\left(x_0+\frac{1}{2}\right)\right] \ , \tag{14}$$

$$e_k(p_0,x_0) = \frac{i}{N_0}\sin(p_0x_0) \ . \tag{15}$$

- Open-SF boundary conditions:

$$E_0 = E_k = \frac{\pi}{N_0}\left(\{0,\dots,N_0-1\}+\frac{1}{2}\right) \ , \tag{16}$$

$$e_0(p_0,x_0) = \frac{i}{N_0}\sin\left[p_0\left(x_0+\frac{1}{2}\right)\right] \ , \tag{17}$$

$$e_k(p_0,x_0) = \frac{1}{N_0}\cos\left[p_0\left(x_0+\frac{1}{2}\right)\right] \ . \tag{18}$$

- Periodic boundary conditions:

$$E_0 = E_k = \frac{2\pi}{N_0}\{0,\dots,N_0-1\} \ , \tag{19}$$

$$e_0(p_0,x_0) = e_k(p_0,x_0) = \frac{1}{N_0}\exp(ip_0x_0) \ . \tag{20}$$

We use the Fourier decomposition to define the intermediate operator D_N

$$[D_\mathrm{N}\pi](x,\mu) = \frac{1}{L^3}\sum_{p_0\in E_\mu}\sum_{\mathbf{p}\in\mathcal{P}} e_\mu(p_0,x_0)e^{i\mathbf{px}}\tilde{D}_\mathrm{N}(p)\tilde{\pi}(p,\mu) \ , \tag{21}$$

$$\tilde{D}_\mathrm{N}(p) = \begin{cases} 1 & \text{if } p=0 \\ 4\sum_\mu \sin^2\frac{p_\mu}{2} & \text{otherwise} \end{cases} \ . \tag{22}$$

Explicity

$$D_\mathrm{N}(x,\mu;y,\nu) = \frac{1}{L^3}\sum_{p_0\in E_\mu}\sum_{\mathbf{p}\in\mathcal{P}} \tilde{D}_\mathrm{N}(p)\delta_{\mu\nu}e^{i\mathbf{p}(\mathbf{x}-\mathbf{y})}e_\mu(p_0,x_0)e_\nu^*(p_0,y_0)w_\nu(y_0) \ . \tag{23}$$

With respect to the scalar product defined by

$$(\phi, \phi)_G = (\phi, G\phi) \ , \tag{24}$$

$$[G\phi](x, \mu) = w_\mu(x_0)\phi(x, \mu) \ . \tag{25}$$

the operator $D_{\rm N}$ is symmetric and strictly positive, i.e.

$$(\phi', D_{\rm N}\phi)_G = (D_{\rm N}\phi', \phi)_G \ , \tag{26}$$

$$(\phi, D_{\rm N}\phi)_G \geq 0 \ , \tag{27}$$

$$(\phi, D_{\rm N}\phi)_G = 0 \ \Leftrightarrow \ \phi = 0 \ . \tag{28}$$

The desired operator is defined as

$$D = G^{1/2} D_{\rm N} G^{-1/2} \ . \tag{29}$$

Symmetry and strict positivity of D with respect to the canonical scalar product of D follow from the corresponding properties of $D_{\rm N}$. Notice that

$$D^\alpha = G^{1/2} D_{\rm N}^\alpha G^{-1/2} \ . \tag{30}$$

The `openQ*D` code uses the fast-Fourier-transform (FFT) algorithm to contruct $\tilde{\pi}(p, \mu)$ from $\pi(x, \mu)$ and vice versa. The FFT is implemented in the module `dft` which is an adaptation of the correponding module in the `NSPT-1.4` code written by Mattia Dalla Brida and Martin Lüscher [4].

References

[1] S. Duane, R. Kenway, B. J. Pendleton and D. Roweth, *Acceleration of Gauge Field Dynamics*, Phys. Lett. B **176**, 143 (1986).

[2] A. Patella, *C* boundary conditions*, code documentation, `doc/cstar.pdf`.

[3] A. Patella, *RHMC algorithm in `openQ*D`*, code documentation, `doc/rhmc.pdf`.

[4] `NSPT-1.4` simulation package, `http://luscher.web.cern.ch/luscher/NSPT`.