Lattice EFT note 3: BCC lattice

Young-Ho Song

October 24, 2018

Contents

Chapter 1

Introduction

The object of this note is to develop a formalism of lattice EFT calculation for non-cubic lattice(BCC or FCC). There may be two motivation. (1) To test the dependence(or independence) of physical observables on the lattice artifacts(lattice size, shape, etc.) (2) To compare the numerical efficiency between different lattice formulation. (Other motivation?)

1.1 Discrete Fourier Transform on a lattice

General introduction for DFT.

1.1.1 1-d case

- For 1D DFT of size N, a sampling(image) is a function, $a(j)$, defined at the points, $j \in [N]$, where a subset $[N] := 0, 1, 2, \ldots N-1$.
- DFT is a mapping

$$\hat{a}(k) = \frac{1}{\sqrt{N}} \sum_{j \in [N]} a(j) e^{-2\pi i \frac{jk}{N}}, k \in [N] \tag{1.1}$$

- If the $a(j)$ are purely real, we have a relation

$$\hat{a}(k) = \hat{a}(-k)^* = \hat{a}(N-k)^*. \tag{1.2}$$

  Thus, one can obtain $\hat{a}(k > N/2+1)$ from $\hat{a}(k \leq N/2)$. (Also, $\hat{a}(0)$ and $\hat{a}(N/2)$ are real.)

- FFTW use this symmetry for r2c or c2r transformation and thus the dimension of input and output is different. Real array have dimension N and complex array have dimension $N/2+1$ in r2c or c2r of FFTW.

1.1.2 2-d and 3-d case

- In a similar way, 2-d DFT of size $N_1 \times N_2$ can be written as

$$\hat{a}(k_1, k_2) = \frac{1}{\sqrt{N_1 N_2}} \sum_{j_1 \in [N_1], j_2 \in [N_2]} a(j_1, j_2) e^{-2\pi i \frac{j_1 k_1}{N_1} - 2\pi i \frac{j_2 k_2}{N_2}} \tag{1.3}$$

  where $k_1 \in [N_1]$ and $k_2 \in [N_2]$.

- In case of real function values a, we get relation

$$\hat{a}(k_1, k_2) = \hat{a}^*(-k_1, -k_2) = \hat{a}^*(N - k_1, N - k_2) \tag{1.4}$$

- In a similar way, 3-d DFT of size $N_1 \times N_2 \times N_3$ can be written as

$$\hat{a}(k_1, k_2, k_3) = \frac{1}{\sqrt{N_1 N_2 N_3}} \sum_{j_1 \in [N_1], j_2 \in [N_2], j_3 \in [N_3]} a(j_1, j_2, j_3) e^{-2\pi i \frac{j_1 k_1}{N_1} - 2\pi i \frac{j_2 k_2}{N_2} - 2\pi i \frac{j_3 k_3}{N_3}} \tag{1.5}$$

  where $k_1 \in [N_1]$, $k_2 \in [N_2]$ and $k_3 \in [N_3]$.

- A similar relation for real function values a, we get

$$\hat{a}(k_1, k_2, k_3) = \hat{a}^*(-k_1, -k_2, -k_3) = \hat{a}^*(N - k_1, N - k_2, N - k_3). \tag{1.6}$$

1.2 FFT on lattice

This is a summary of [1] with additional comments.

1.2.1 d-dimensional lattice

- Lattice L is set of points generated by basis vectors,

$$L := \left\{ \sum_{i=1}^{d} k_i \boldsymbol{u}_i, k_i \in Z \right\}. \tag{1.7}$$

- sampling matrix of generator matrix $\boldsymbol{U}$ is a matrix with basis vectors as columns.

- Sublattice L_0 is a d-dimensional lattice which is also a subset of lattice L. The generator of sublattice L_0 is denoted as $\boldsymbol{V}$. A periodicity matrix $\boldsymbol{M}$ relates $\boldsymbol{V}$ (the generator of sublattice L_0) with $\boldsymbol{U}$(generator of L) , $\boldsymbol{V} = \boldsymbol{U}\boldsymbol{M}$. The periodicity matrix can be written as $\boldsymbol{M} = \boldsymbol{E}\boldsymbol{D}\boldsymbol{F}$ with uni-modular matrix $\boldsymbol{E}$, $\boldsymbol{F}$ and diagonal matrix $\boldsymbol{D}$.

- Quotient group $L/L_0 := \{\boldsymbol{x} + L_0 : \boldsymbol{x} \in L\}$. the coset $\boldsymbol{x} + L_0$ is denoted $\bar{\boldsymbol{x}}$. If $S \subset L$ such that $\{\bar{\boldsymbol{s}} : \boldsymbol{s} \in S\} = L/L_0$ and $\bar{\boldsymbol{a}} \neq \bar{\boldsymbol{b}}$ whenever $\boldsymbol{a}, \boldsymbol{b} \neq S$ with $\boldsymbol{a} \neq \boldsymbol{b}$, $\mathbf{S}$ is called a set of coset representatives of the quotient group L/L_0.

- dual lattice(reciprocal lattice) of L is

$$L^* := \{\boldsymbol{s} \in R^d : \langle \boldsymbol{r}, \boldsymbol{s} \rangle \in Z \text{ for each } \boldsymbol{r} \in L\} \tag{1.8}$$

  where $\langle \boldsymbol{r}, \boldsymbol{s} \rangle$ denotes the inner product.

  If L_0 is a sublattice of L and L^* is a sublattice of L_0^*, the two quotient groups $G = L/L_0$ and $\widehat{G} = L_0^*/L_0^*$ are congruent.

- If $\boldsymbol{U}$ is a generator of the lattice L, $\boldsymbol{U}^{-T}$ is a generator of the dual lattice L^*. Dual lattice (in frequency space or reciprocal lattice) $L^*(L_0^*)$ is generated from generator matrix $\tilde{U}(\tilde{V})$,

$$\tilde{\boldsymbol{U}} = (\boldsymbol{U}^{-1})^T, \quad \tilde{\boldsymbol{V}} = (\boldsymbol{V}^{-1})^T. \tag{1.9}$$

  (short proof) For any lattice point $\boldsymbol{r} \in L$, there is an integer set $[\boldsymbol{r}]$ such that $\boldsymbol{r} = U[\boldsymbol{r}]$. If $\boldsymbol{\sigma}$ can be written as $\boldsymbol{s} = (U^{-1})^T[\boldsymbol{\sigma}]$ with an integer set $[\boldsymbol{s}]$, we can see $\langle \boldsymbol{r}, \boldsymbol{s} \rangle = \boldsymbol{r}^T \cdot \boldsymbol{s} = [\boldsymbol{r}]^T \cdot [\boldsymbol{s}]$ =(integer). Thus, the reciprocal lattice can be generated with $(U^{-1})^T$

- Let quotient groups $G = L/L_0$ in position space and $\hat{G} = L_0^*/L^*$ in frequency. Let $P(\hat{P})$ be a set of coset representatives of the quotient group $G(\hat{G})$. Then for any $\boldsymbol{r} \in P$, there exists $[\boldsymbol{r}] \in Z^d$ such that $\boldsymbol{r} = \boldsymbol{U} \cdot [\boldsymbol{r}]$. Similarly, for any $\boldsymbol{s} \in \hat{P}$, there exists $[\boldsymbol{s}]_{\tilde{V}} \in Z^d$ such that $\boldsymbol{s} = \tilde{\boldsymbol{V}}[\boldsymbol{s}]_{\tilde{V}}$. (In other words, when $\boldsymbol{r}$ is a Cartesian coordinates of a lattice point, $[\boldsymbol{r}]$ is its coordinates in generator basis of the lattice L. Similarly, $\boldsymbol{s}$ is a frequency in Cartesian coordinates and $[\boldsymbol{s}]$ in generator basis of L^*.) Thus,

$$\begin{aligned}\langle \boldsymbol{r}, \boldsymbol{s} \rangle &= \boldsymbol{r}^T \cdot \boldsymbol{s} = (\boldsymbol{U} \cdot [\boldsymbol{r}])^T \cdot \tilde{\boldsymbol{V}} \cdot [\boldsymbol{s}]_{\tilde{\boldsymbol{V}}} = (\boldsymbol{U} \cdot [\boldsymbol{r}])^T \cdot (\boldsymbol{V}^{-1})^T \cdot [\boldsymbol{s}]_{\tilde{\boldsymbol{V}}} \\ &= [\boldsymbol{r}]^T \cdot \boldsymbol{U}^T (\boldsymbol{U} \cdot \boldsymbol{M})^{-1,T} \cdot [\boldsymbol{s}]_{\tilde{\boldsymbol{V}}} \\ &= [\boldsymbol{r}]^T \cdot (\boldsymbol{M}^{-1})^T \cdot [\boldsymbol{s}]_{\tilde{V}} \\ &= \left(\boldsymbol{E}^{-1} \cdot [\boldsymbol{r}]\right)^T \cdot (\boldsymbol{D}^{-1})^T \cdot \left((\boldsymbol{F}^{-1})^T [\boldsymbol{s}]_{\tilde{V}}\right) \end{aligned} \tag{1.10}$$

1.2.2 DFT on d-dimensional lattice

- DFT on the quotient group G is the mapping $\mathcal{F} : a \to \hat{a}$ from C^G to $C^{\hat{G}}$ defined by

$$\hat{a}(\bar{s}) = \frac{1}{\sqrt{|G|}} \sum_{\bar{r} \in G} a(\bar{r}) \cdot e^{-2\pi i \langle \boldsymbol{r}, \boldsymbol{s} \rangle} \tag{1.11}$$

  for each $\bar{s} := s + L^* \in \hat{G}$, where $\boldsymbol{s}$ is an element in L_0^* such that $s + L^* = \bar{s}$.

- Inverse FT is

$$a(\bar{r}) = \frac{1}{\sqrt{|G|}} \sum_{\bar{s} \in \hat{G}} \hat{a}(\bar{s}) \cdot e^{2\pi i \langle \boldsymbol{r}, \boldsymbol{s} \rangle} \tag{1.12}$$

  for each $\bar{r} \in G$, where r is an element in L such that $r + L_0 = \bar{r}$.

- Reduction to standard FFT: As noted in the previous section, we can express

$$\begin{aligned}\langle \boldsymbol{r}, \boldsymbol{s} \rangle = \boldsymbol{r}^T \cdot \boldsymbol{s} &= [\boldsymbol{r}]^T \cdot (\boldsymbol{M}^{-1})^T \cdot [\boldsymbol{s}]_{\tilde{V}} \\ &= \left(\boldsymbol{E}^{-1}[\boldsymbol{r}]\right)^T \boldsymbol{D}^{-1} \left(\boldsymbol{F}^{-T}[\boldsymbol{s}]_{\tilde{V}}\right). \end{aligned} \tag{1.13}$$

  Thus, because $\boldsymbol{D}$ is diagonal, the DFT $\hat{a}$ can be computed by way of one-dimensional FFTs whose sizes are diagonal entries of $\boldsymbol{D}$. ($\sqrt{|G|} = \sqrt{\det \boldsymbol{D}}$.)

- Let $[P] = \{[\boldsymbol{r}] : \boldsymbol{r} \in P\}$ and $[\hat{P}] = \{[\boldsymbol{s}]_{\tilde{V}} : \boldsymbol{s} \in \hat{P}\}$, DFT in d-dimension is

$$\begin{aligned}\hat{a}(\tilde{V} \cdot [\boldsymbol{s}]_{\tilde{V}}) &= \sum_{[\boldsymbol{r}] \in [P]} a(\boldsymbol{U} \cdot [\boldsymbol{r}]) \cdot e^{-2\pi i [\boldsymbol{r}] \cdot (\boldsymbol{M}^{-1})^T \cdot [\boldsymbol{s}]_{\tilde{V}}} \\ &= \sum_{[\boldsymbol{r}] \in [P]} a(\boldsymbol{U} \cdot [\boldsymbol{r}]) \cdot e^{-2\pi i \left(\boldsymbol{E}^{-1}[\boldsymbol{r}]\right)^T \boldsymbol{D}^{-1} \left(\boldsymbol{F}^{-T}[\boldsymbol{s}]_{\tilde{V}}\right)} \end{aligned} \tag{1.14}$$

- inverse DFT is

$$a(\boldsymbol{U} \cdot [\boldsymbol{r}]) = \frac{1}{|P|} \sum_{[\boldsymbol{s}]_{\tilde{V}} \in [\hat{P}]} a(\tilde{\boldsymbol{V}} \cdot [\boldsymbol{s}]_{\tilde{V}}) \cdot e^{2\pi i \left(\boldsymbol{E}^{-1}[\boldsymbol{r}]\right)^T \boldsymbol{D}^{-1} \left(\boldsymbol{F}^{-T}[\boldsymbol{s}]_{\tilde{V}}\right)} \tag{1.15}$$

- **Caution:** We have relations with d-dimensional deltas,

$$\sum_{[\boldsymbol{r}]} e^{2\pi i \langle \boldsymbol{r}, (\boldsymbol{s} - \boldsymbol{s}') \rangle} = |P| \delta^{(d)}_{[\boldsymbol{s}],[\boldsymbol{s}']}, \quad \sum_{[\boldsymbol{s}]} e^{2\pi i \langle (\boldsymbol{r} - \boldsymbol{r}'), \boldsymbol{s} \rangle} = |P| \delta^{(d)}_{[\boldsymbol{r}],[\boldsymbol{r}']}. \tag{1.16}$$

  However, the relation does not hold for individual components,

$$\sum_{r_1} e^{2\pi i r_1 (s_1 - s_1')} \neq D_{11} \delta_{s_1 s_1'} \tag{1.17}$$

1.3 Cubic/BCC/FCC Lattice

- From now on, everything will be written in lattice units a. And every lattice points will be labeled with an integer number **lp**.

- **U**: generator matrix for cubic(c), FCC(f) and BCC(b) lattice L are

$$\boldsymbol{U}_c = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \boldsymbol{U}_f = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}, \boldsymbol{U}_b = \begin{pmatrix} 2 & 0 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix} \tag{1.18}$$

  For an inverse transformation,(from Cartesian to lattice coordinate)

$$\boldsymbol{U}_c^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \boldsymbol{U}_f^{-1} = \begin{pmatrix} 1/2 & -1/2 & 1/2 \\ 1/2 & 1/2 & -1/2 \\ -1/2 & 1/2 & 1/2 \end{pmatrix}, \boldsymbol{U}_b^{-1} = \begin{pmatrix} 1/2 & 0 & -1/2 \\ 0 & 1/2 & -1/2 \\ 0 & 0 & 1 \end{pmatrix} \tag{1.19}$$

- **V** and **M**: generator matrix of a sublattice L_0 of L and periodicity matrix $\boldsymbol{M}$ is related as

$$\boldsymbol{V} = \boldsymbol{UM}. \tag{1.20}$$

  Sublattice(s) of cubic(FCC,BCC) lattice is cubic(BCC,FCC). Generator matrix are

$$\begin{aligned} \boldsymbol{V}_{cs} &= k \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{M}_{cs} = k \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \\ \boldsymbol{V}_{fs} &= 2k \begin{pmatrix} 2 & 0 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{M}_{fs} = k \begin{pmatrix} 2 & -2 & 1 \\ 2 & 2 & 1 \\ -2 & 2 & 1 \end{pmatrix}, \\ \boldsymbol{V}_{bs} &= 2k \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}, \quad \boldsymbol{M}_{bs} = k \begin{pmatrix} 0 & -1 & 1 \\ -1 & 0 & 1 \\ 2 & 2 & 0 \end{pmatrix}. \end{aligned} \tag{1.21}$$

  where k is a measure of periodicity. (It is convenient to use k as an even number.)

  There can be other choices of sublattice and $\boldsymbol{V}$ depending on the periodic condition. See later section for choices for cubic periodic boundary condition.

- We can think a sampling of a function which is periodic,

$$f(\boldsymbol{r}) = f(\boldsymbol{r} + \boldsymbol{Vn}), \quad \boldsymbol{n} \in Z^3. \tag{1.22}$$

  The sublattice generator $\boldsymbol{V}$ (periodicity matrix $\boldsymbol{M}$) also determines the periodicity in frequency domain.[1]

[1] dual lattice point $\boldsymbol{k}$ have to satisfy for any integers $\boldsymbol{n}$,

$$\boldsymbol{k}^T \boldsymbol{Vn} = (integer).$$

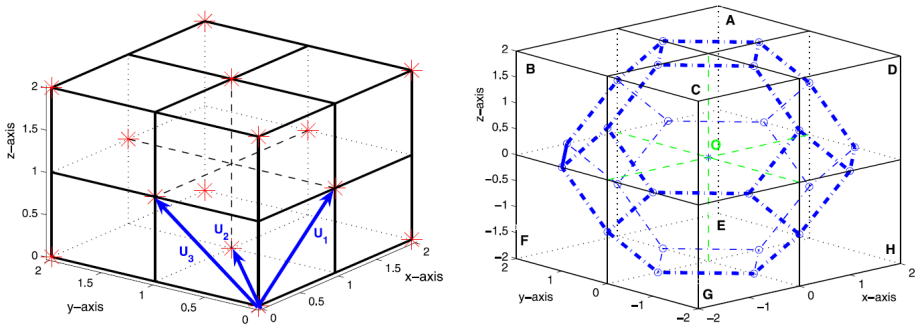

Figure 1.1: FCC lattice and its basis vectors and Voronoi cell(truncated octahedron).

- Each case, $\boldsymbol{M} = \boldsymbol{EDF}$ where $|\det \boldsymbol{E}| = |\det \boldsymbol{F}| = 1$ and $\boldsymbol{D}$ is a diagonal matrix.

$$\boldsymbol{E}_{cs} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \boldsymbol{D}_{cs} = k \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \boldsymbol{F}_{cs} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$
$$\boldsymbol{E}_{fs} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}, \boldsymbol{D}_{fs} = k \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{pmatrix}, \boldsymbol{F}_{fs} = \begin{pmatrix} 2 & -2 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix},$$
$$\boldsymbol{E}_{bs} = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 2 & 2 & 1 \end{pmatrix}, \boldsymbol{D}_{bs} = k \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 4 \end{pmatrix}, \boldsymbol{F}_{bs} = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \quad (1.23)$$

  For later convenience,

$$\boldsymbol{E}_{fs}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}, \quad \boldsymbol{F}_{fs}^{-T} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 2 \\ -1 & 0 & 2 \end{pmatrix},$$
$$\boldsymbol{E}_{bs}^{-1} = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 2 & 2 & 1 \end{pmatrix}, \quad \boldsymbol{F}_{bs}^{-T} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}. \quad (1.24)$$

- **Voronoi cell $\boldsymbol{\Omega}$**: See Fig. 1.1 and 1.2 of basis vectors and Voronoi cells for FCC and BCC lattice.

  Edge length L of a Voronoi cell is $L = ka$ for cubic, $L = \sqrt{2}ka$ for FCC lattice, and $L = \sqrt{3}ka$ for BCC lattice. The volume V of a Voronoi cell with a edge length L are

$$\begin{aligned} V_c &= L^3 \quad \text{cube}, \\ V_{fs} &= 8\sqrt{2}L^3 \quad \text{truncated octahedron}, \\ V_{bs} &= \frac{16\sqrt{3}}{9}L^3 \quad \text{rhombic dodecahedron}. \end{aligned} \quad (1.25)$$

- $\boldsymbol{S}_\Omega$ and $\boldsymbol{S}_{\Omega,coef}$: There is no unique choice of coset representatives. One choice is to find non-redundant lattice points (w.r.t. periodicity) in a Voronoi cell centered at the origin.

  Let us denote a set of coset representatives of the quotient group L/L_0 as S_Ω.

$$S_\Omega = \{\boldsymbol{r} | \boldsymbol{r} \in L/L_0, \boldsymbol{r} \in \boldsymbol{\Omega}\} \quad (1.26)$$

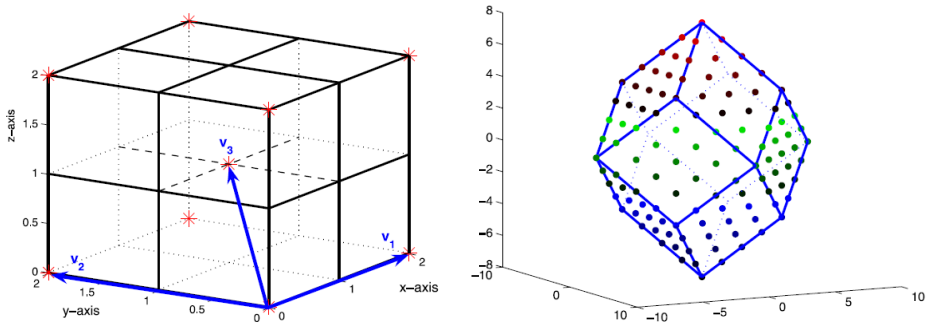

Figure 1.2: BCC lattice and its basis vectors and Voronoi cell(rhombic dodecahedron). Example corresponds to edge length $4\sqrt{3}a$ ($k = 4$).

In other words, S_Ω is a set of Cartesian coordinates of lattice points in coset representatives within a Voronoi cell. Details on the construction of S_Ω is explained later.

For each points in S_Ω, there exists a set of corresponding coefficients(lattice coordinates) of basis vectors. The set of coefficients corresponding to S_Ω are denoted as $S_{\Omega,coef}$,

$$S_{\Omega,coef} = \{[\boldsymbol{r}] : \boldsymbol{U} \cdot [\boldsymbol{r}] \in S_\Omega\}. \tag{1.27}$$

- **Nearest Neighbors**: The number of nearest neighbor points in cubic(FCC,BCC) lattice are 6(12,8) in distance $a(\sqrt{2}a, \sqrt{3}a)$. One can find their location from a given lattice point by displacements in lattice coordinates $[\boldsymbol{r}]$,

$$\begin{aligned}
\Delta\boldsymbol{n}_{cubic} &= (+1,0,0),(-1,0,0),(0,1,0),(0,-1,0),(0,0,1),(0,0,-1),\\
\Delta\boldsymbol{n}_{FCC} &= (-1,0,0),(-1,0,1),(-1,1,0),(0,-1,0),(0,-1,1),(0,0,-1),\\
&\quad (0,0,1),(0,1,-1),(0,1,0),(1,-1,0),(1,0,-1),(1,0,0),\\
\Delta\boldsymbol{n}_{BCC} &= (-1,0,1),(0,0,1),(0,-1,1),(-1,-1,1),\\
&\quad (0,1,-1),(1,1,-1),(1,0,-1),(0,0,-1).
\end{aligned} \tag{1.28}$$

Next nearest neighbors of cubic(BCC) is 12(6) points. Next-next nearest neighbors of cubic(BCC) are 8(12) points.

- $\boldsymbol{S}^A_\Omega$ and $\boldsymbol{S}^A_{\Omega,coef}$: Frequency set S^A_Ω is a set of coset representatives of L^*_0/L^*.(For example, $[N]/N$ in case of 1-D case.)

$$S^A_\Omega = \{s | s \in L^*_0/L^*\}. \tag{1.29}$$

One can define $S^A_{\Omega,coef}$ such that

$$S^A_{\Omega,coef} = \{[\boldsymbol{s}] \in Z^d : (V^T)^{-1}[\boldsymbol{s}] \in S^A_\Omega\}. \tag{1.30}$$

- (According to the proposition given in [1]), one can obtain $S^A_{\Omega,coef}$ from $S_{\Omega,coef}$ as

$$S^A_{\Omega,coef} = \{A^{-1}c : c \in S_{\Omega,coef}\}. \tag{1.31}$$

According to [1], unimodular matrix A is defined such that

$$\begin{aligned}
(\boldsymbol{V}^T_{cs})^{-1} &= \frac{1}{k}\boldsymbol{U}_c \cdot \boldsymbol{A}_{cs}, \quad (\boldsymbol{U}^T_c)^{-1} = \frac{1}{k}\boldsymbol{V}_{cs} \cdot \boldsymbol{A}^T_{cs},\\
(\boldsymbol{V}^T_{fs})^{-1} &= \frac{1}{4k}\boldsymbol{U}_f \cdot \boldsymbol{A}_{fs}, \quad (\boldsymbol{U}^T_f)^{-1} = \frac{1}{4k}\boldsymbol{V}_{fs} \cdot \boldsymbol{A}^T_{fs},\\
(\boldsymbol{V}^T_{bs})^{-1} &= \frac{1}{4k}\boldsymbol{U}_b \cdot \boldsymbol{A}_{bs}, \quad (\boldsymbol{U}^T_b)^{-1} = \frac{1}{4k}\boldsymbol{V}_{bs} \cdot \boldsymbol{A}^T_{bs}.
\end{aligned} \tag{1.32}$$

Then,

$$\boldsymbol{A}_{cs} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \boldsymbol{A}_{fs} = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 1 & -1 \\ -1 & 0 & 1 \end{pmatrix}, \boldsymbol{A}_{bs} = \begin{pmatrix} 0 & -1 & 1 \\ -1 & 0 & 1 \\ 1 & 1 & -1 \end{pmatrix}, \quad (1.33)$$

It may be defined in a compact way,

$$\boldsymbol{A} = (|\det \boldsymbol{U} \det \boldsymbol{V}|)^{\frac{1}{3}} \boldsymbol{U}^{-1} \boldsymbol{V}^{-T}, \quad \boldsymbol{V}^{-T} = (|\det \boldsymbol{U} \det \boldsymbol{V}|)^{-\frac{1}{3}} \boldsymbol{U} \boldsymbol{A} \quad (1.34)$$

However, the proposition about $S^A_{\Omega,coef}$ seems to be only applicable for above cases. (not for other lattice or other periodic boundary condition.)

- **size of** S_Ω: The number of lattice points of S_Ω, $S_{\Omega,coef}$, S^A_Ω, $S^A_{\Omega,coef}$ for given size parameter k are

$$N = \det \boldsymbol{D} = \frac{|\det \boldsymbol{V}|}{|\det \boldsymbol{U}|} \quad \Rightarrow \quad N_{cs} = k^3, \quad N_{fs} = 16k^3, \quad N_{bs} = 4k^3. \quad (1.35)$$

- $\boldsymbol{S}_{\Omega,cartes}$: (this is not in the [1]). we may define a set $S_{\Omega,cartes}$ such that for a $\boldsymbol{p} \in S_{\Omega,cartes}$,

$$\boldsymbol{p}_i = mod((\boldsymbol{E}^{-1} \cdot [\boldsymbol{r}])_i, D_{ii}), \quad [\boldsymbol{r}] \in S_{\Omega,coef} \quad (1.36)$$

Now, the $\boldsymbol{p}_i$ is in range $[0, D_{ii} - 1]$ as like 1-D periodic lattice in i-th direction. [2] Also, in frequency domain, one may define $\boldsymbol{w} \in S^A_{\Omega,cartes}$,

$$\boldsymbol{w}_i = mod((\boldsymbol{F}^{-T}[\boldsymbol{s}])_i, D_{ii}), \quad [\boldsymbol{s}] \in S^A_{\Omega,coef}. \quad (1.37)$$

This is similar to the cubic lattice case.

Also, because of modular operation, it is easy to find a $\boldsymbol{p} \in S_{\Omega,cartes}$ for any lattice point $\boldsymbol{r} \notin S_\Omega$. In other words, any two lattice points give the same $\boldsymbol{p}$ are congruent points.

- **In summary** : For given size parameter k of a lattice and periodicity , N number of lattice points are defined. For $lp - th$ lattice point, there are several representations we can use,

$$lp \in [1, N] \quad \leftrightarrow \quad \boldsymbol{r} \in S_\Omega \quad \leftrightarrow \quad [\boldsymbol{r}] \in S_{\Omega,coef} \quad \leftrightarrow \quad \boldsymbol{p} \in S_{\Omega,cartes} \quad (1.38)$$

1.4 A Periodic condition in lattice coordinates

We may express a lattice points in three coordinates.

- In a Cartesian basis, $\boldsymbol{r} = (n_x, n_y, n_z) \in S_\Omega$.
- In a Generator basis, $\boldsymbol{q} = [\boldsymbol{r}] = (q_1, q_2, q_3) \in S_{\Omega,coef}$. [3]
- In a new Cartesian basis, $\boldsymbol{p} = (p_1, p_2, p_3) \in S_{\Omega,cartes}$.

Any lattice point not in S_Ω is congruent with a point in S_Ω. The congruency(periodicity) is easiest to see in $\boldsymbol{p}$, we just have

$$\psi(p_1, p_2, p_3) = \psi(p_1 + D_{11}, p_2 + D_{22}, p_3 + D_{33}) \quad (1.39)$$

[2] Be careful that the action of **mod** is different depending on Fortran or MATLAB. Fortran **modulo** and MATLAB **mod** gives modulo(A,P)= $A - floor(A/P) * P$ which have the same sign of P. On the other hand Fortran **mod** gives mod(A,P)= $A - int(A/P) * P$ which have the same sign of A. In both case magnitude are smaller than $|P|$.

[3] (Actually, this should be q^1, q^2, q^3 in curvilinear coordinates. But I will ignore the difference from now on.)

However, the periodicity condition is complicate in other coordinates, $f(\boldsymbol{r}) = f(\boldsymbol{r}')$ for $\boldsymbol{r}$ if there exists integers $\boldsymbol{n}$ such that

$$\boldsymbol{r} - \boldsymbol{r}' = \boldsymbol{V}\boldsymbol{n} \tag{1.40}$$

or $\phi(\boldsymbol{q}) = \phi(\boldsymbol{q}')$ for $\boldsymbol{q}$ if there exists integers $\boldsymbol{n}$ such that

$$\boldsymbol{q} - \boldsymbol{q}' = \boldsymbol{U}^{-1}\boldsymbol{V}\boldsymbol{n} \tag{1.41}$$

Thus, probably the easiest way to treat the lattice is to always convert between lattice coordinates $\boldsymbol{q}$ and $\boldsymbol{p}$.

1.4.1 Construction of S_Ω

Thus, one way to construct S_Ω lattice is as follows:

- Construct $\boldsymbol{p} = (p_1, p_2, p_3) \in S_{\Omega,cartes}$ with integers $0 \le p_1 \le D_{11} - 1$, $0 \le p_2 \le D_{22} - 1$, $0 \le p_3 \le D_{33} - 1$.
- Convert $\boldsymbol{p}$ to $[\boldsymbol{r}]$ and $\boldsymbol{r}$ with appropriate integers $\boldsymbol{n} = (n_1, n_2, n_3)$,

$$\begin{aligned} [\boldsymbol{r}] &= \boldsymbol{E} \cdot (\boldsymbol{p} - \boldsymbol{D} \cdot \boldsymbol{n}), \\ \boldsymbol{r} &= \boldsymbol{U} \cdot [\boldsymbol{r}] = \boldsymbol{U} \cdot \boldsymbol{E} \cdot (\boldsymbol{p} - \boldsymbol{D} \cdot \boldsymbol{n}). \end{aligned} \tag{1.42}$$

  However, this conversion is not unique. One way to choose $\boldsymbol{n}$ is to minimize the distance $|\boldsymbol{r}|$ from the origin. Once $\boldsymbol{n}$ is chosen for each $\boldsymbol{p}$ points, S_Ω and $S_{\Omega,coef}$ can be constructed.

- Or, if S_Ω is easy to construct like cubic lattice. One can obtain $S_{\Omega,coef}$ and $S_{\Omega,coef}$ by

$$\begin{aligned} [\boldsymbol{r}] &= \boldsymbol{U}^{-1} \cdot \boldsymbol{r}, \quad \boldsymbol{r} \in S_\Omega, \\ \boldsymbol{p}_i &= \mod((\boldsymbol{E}^{-1}[\boldsymbol{r}])_i, D_{ii}), \quad i = 1, 2, 3. \end{aligned} \tag{1.43}$$

1.4.2 Distance between two lattice points

The distance between two points, $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$, on the lattice have to be computed by taking into account periodicity. Let us define a function

$$d(\boldsymbol{r}, \boldsymbol{n}) = \sqrt{(\boldsymbol{r} + \boldsymbol{V} \cdot \boldsymbol{n})^2}. \tag{1.44}$$

The distance of two lattice points is the minimum of $d(\boldsymbol{r}_1 - \boldsymbol{r}_2, \boldsymbol{n})$ for given $\boldsymbol{r} = \boldsymbol{r}_1 - \boldsymbol{r}_2$ vector. Question is how to find an integer vector $\boldsymbol{n}$ which gives the minimum.

Cubic periodic boundary

For example, let us consider 1-d periodic lattice of size N. Let $-\frac{N}{2} \le x_0 < \frac{N}{2}$, $x = x_0 + Nn$. Then for given x, we can find x_0 and n by

$$n = \mathrm{round}(\frac{x}{N}), \quad x_0 = x - Nn. \tag{1.45}$$

The same can be applied for cubic periodic boundary condition, where $\boldsymbol{V}$ is diagonal,

$$\boldsymbol{n} = \mathrm{round}(\boldsymbol{V}^{-1}\boldsymbol{r}), \quad \boldsymbol{r}_0 = \boldsymbol{r} - \boldsymbol{V}\boldsymbol{n}. \tag{1.46}$$

where round is implied for each Cartesian component. Thus, the distance can be computed simply by $\boldsymbol{r}_0^2$.

BCC/FCC periodic boundary

Suppose $\boldsymbol{r} \in S_\Omega$ and $\boldsymbol{r}' = \boldsymbol{r} + \boldsymbol{V} \cdot \boldsymbol{n}$. The problem is to find $\boldsymbol{r}$ or $\boldsymbol{n}$ from given $\boldsymbol{r}'$. Let us define $\boldsymbol{p}' = \boldsymbol{E}^{-1}\boldsymbol{U}^{-1}\boldsymbol{r}'$,

$$\begin{aligned} & \boldsymbol{r}' = \boldsymbol{U}\boldsymbol{E}\boldsymbol{p}' = \boldsymbol{U}\boldsymbol{E}\boldsymbol{p} + \boldsymbol{V} \cdot \boldsymbol{n}, \\ \Rightarrow \quad & \boldsymbol{p}' = \boldsymbol{p} + \boldsymbol{D} \cdot \boldsymbol{m}, \quad \boldsymbol{m} = \boldsymbol{F} \cdot \boldsymbol{n}. \end{aligned} \tag{1.47}$$

Since $\boldsymbol{D}$ is diagonal, we can consider $\boldsymbol{p}$ in each component separately. However, unlike the cubic lattice, the corresponding range of $\boldsymbol{p}$ depends on the choice of S_Ω. (In other words, $\boldsymbol{p}$ is not necessarily in range $0 \le p_i < D_{ii}$.) For given $\boldsymbol{r}'$ or $\boldsymbol{p}'$, there is a unique $\boldsymbol{p}_{0,i} = \mod(\boldsymbol{p}'_i, \boldsymbol{D}_{ii})$ for $0 \le p_{0,i} < D_{ii}$, $\boldsymbol{p}_0 \in S_{\Omega,cartes}$. However, the exact relation between $\boldsymbol{p}_0$ and $\boldsymbol{p}$ depends on the particular choice of S_Ω. Thus, it seems to be difficult to make a simple relation like cubic periodic boundary case.

If we have already tabulated S_Ω, we can make a table of correspondence $\boldsymbol{r} \leftrightarrow \boldsymbol{p}_0$ where $\boldsymbol{r} \in S_\Omega$. However, when the lattice size parameter is large, this can be slow.

One may shift the $\boldsymbol{r}'$ or $\boldsymbol{p}'$ by $\boldsymbol{n} = \mathrm{round}(\boldsymbol{V}^{-1}\boldsymbol{r}')$ or $\boldsymbol{m} = \mathrm{round}(\boldsymbol{D}^{-1}\boldsymbol{p}')$ first and compare with additional nearby shifts. This may reduce number of trials.

1.5 Cartesian Periodic boundary condition

Sublattice and Periodicity: One can choose different Sublattice L_0 (this $\boldsymbol{V}$ and $\boldsymbol{M}$) other than shown in previous section. (This corresponds to divide space into different blocks. Previous cubic lattice corresponds to divide space into large periodic cubes. On the other hand, previous BCC lattice corresponds to divide space into periodic rhombic dodecahedrons.)

Imposing Cartesian periodic boundary condition may be easier to use. Ref. [2] gives very simple implementation of FFT,which is redundant, for cubic periodic boundary condition with BCC/FCC lattice. Ref. [1] gives non-redundant implementation of FFT for cubic periodic boundary condition with BCC/FCC lattice. See the appendix for explanation of Ref. [2].

Following is the implementation according to Ref. [1].

1.5.1 BCC lattice case

For BCC(b) matrix with Cubic(c) sublattice(s) with even integer length $N = 2k$, we have

$$\boldsymbol{V}_{bcs} := N \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \boldsymbol{U}_b \boldsymbol{M}_{bcs}, \quad \boldsymbol{M}_{bcs} := \frac{N}{2} \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 2 \end{pmatrix}. \tag{1.48}$$

$$\boldsymbol{E}_{bcs} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{D}_{bcs} = k \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \quad \boldsymbol{F}_{bcs} = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}. \tag{1.49}$$

One choice of S_Ω is for integers $r, s, t \in Z$,[4]

$$\begin{aligned} S_\Omega &= \left\{ \begin{pmatrix} 2r \\ 2s \\ 2t \end{pmatrix}, \begin{pmatrix} 2r+1 \\ 2s+1 \\ 2t+1 \end{pmatrix} : r, s, t \in [0, \frac{N}{2} - 1] \right\}, \\ S_{\Omega,coef} &= \{\boldsymbol{U}^{-1}\boldsymbol{r} : \boldsymbol{r} \in S_\Omega\} \end{aligned} \tag{1.50}$$

[4] To make center of Voronoi cell at the origin, for symmetric form, one can use $r, s, t \in [-\frac{N}{4}, \frac{N}{4} - 1]$ for S_Ω.

For lattice point $\boldsymbol{r} = \boldsymbol{U}[\boldsymbol{r}]$, we have[5]

$$\boldsymbol{E}^{-1}[\boldsymbol{r}] = \begin{pmatrix} r-t \\ s-t \\ 2t \end{pmatrix}, \begin{pmatrix} r-t \\ s-t \\ 2t+1 \end{pmatrix}. \tag{1.51}$$

One choice of S^A_Ω is for integers $u, v, w \in Z$,

$$\begin{aligned} S^A_\Omega &= \{(\frac{u}{N}, \frac{v}{N}, \frac{w}{N}) : u, v \in [0, \frac{N}{2} - 1], w \in [0, N-1]\}, \\ S^A_{\Omega,coef} &= \{[\boldsymbol{s}] : \boldsymbol{s} = \boldsymbol{V}^{-T}[\boldsymbol{s}], \boldsymbol{s} \in S^A_\Omega\} \end{aligned} \tag{1.52}$$

for dual lattice points, $\boldsymbol{s}$, $[\boldsymbol{s}]$,

$$\boldsymbol{F}^{-T}[\boldsymbol{s}] = \begin{pmatrix} u \\ v \\ u+v+w \end{pmatrix}. \tag{1.53}$$

By taking modular operation, we have $S_{\Omega,cartes}$ and $S^A_{\Omega,cartes}$ and these can be used for DFT.

Note: Construction of S_Ω which is symmetric w.r.t. the origin is easy. However, it is not trivial for me to construct S^A_Ω if one wants to make it symmetric to the origin such that $-\frac{N}{2} \le u, v, w < \frac{N}{2} - 1$ and close to the $(0, 0, 0)$. We may follow following algorithm to construct S^A_Ω which is symmetric to the origin.

- (1) Any combination of (u, v, w), $u, v \in [0, \frac{N}{2} - 1], w \in [0, N-1]$ is unique.
- (2) test following cases for a (u, v, w),

$$\begin{aligned} &(u, v, w-N), (u, v-N, w), (u-N, v, w), (u, v-N, w-N), (u-N, v, w-N) \\ &(u - \frac{N}{2}, v - \frac{N}{2}, w), (u - \frac{N}{2}, v, w - \frac{N}{2}), (u, v - \frac{N}{2}, w - \frac{N}{2}), \\ &(u - \frac{N}{2}, v - \frac{N}{2}, w - N) \end{aligned} \tag{1.54}$$

- (3) Choose one of them which is closest to the origin.

1.5.2 FCC lattice case

For FCC(f) matrix with Cubic(c) sublattice(s) with even integer length $N = 2k$, we have

$$\boldsymbol{V}_{fcs} := N \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \boldsymbol{U}_b \boldsymbol{M}_{fcs}, \quad \boldsymbol{M}_{fcs} := \frac{N}{2} \begin{pmatrix} 1 & -1 & 1 \\ 1 & 1 & -1 \\ -1 & 1 & 1 \end{pmatrix}. \tag{1.55}$$

$$\boldsymbol{E}_{fcs} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & -1 \\ -1 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{D}_{fcs} = k \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \quad \boldsymbol{F}_{fcs} = \begin{pmatrix} 1 & -1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{1.56}$$

[5] $$\boldsymbol{E}^{-1}_{bcs} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \boldsymbol{F}^{-T}_{bcs} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}.$$

One choice of S_Ω is for integers $r, s, t \in Z$,

$$S_\Omega = \left\{ \boldsymbol{r} : \boldsymbol{r} = \begin{pmatrix} 2r \\ 2s \\ 2t \end{pmatrix}, \begin{pmatrix} 2r \\ 2s+1 \\ 2t+1 \end{pmatrix}, \begin{pmatrix} 2r+1 \\ 2s \\ 2t+1 \end{pmatrix}, \begin{pmatrix} 2r+1 \\ 2s+1 \\ 2t \end{pmatrix}, \quad r, s, t \in [0, \frac{N}{2} - 1] \right\},$$

$$S_{\Omega,coef} = \{ \boldsymbol{U}^{-1}\boldsymbol{r} : \boldsymbol{r} \in S_\Omega \} \tag{1.57}$$

For a lattice point $\boldsymbol{U}[\boldsymbol{r}]$, we have[6]

$$\boldsymbol{E}^{-1}[\boldsymbol{r}] = \begin{pmatrix} r-s+t \\ 2s \\ 2t \end{pmatrix}, \begin{pmatrix} r-s+t \\ 2s+1 \\ 2t+1 \end{pmatrix}, \begin{pmatrix} r-s+t+1 \\ 2s \\ 2t+1 \end{pmatrix}, \begin{pmatrix} r-s+t \\ 2s+1 \\ 2t \end{pmatrix}. \tag{1.58}$$

One choice of S_Ω^A is for integers $u, v, w \in Z$,

$$\begin{aligned} S_\Omega^A &= \{ (\frac{u}{N}, \frac{v}{N}, \frac{w}{N}) : u \in [0, \frac{N}{2} - 1], v, w \in [0, N-1] \}, \\ S_{\Omega,coef}^A &= \{ [\boldsymbol{s}] : \boldsymbol{s} = \boldsymbol{V}^{-T}[\boldsymbol{s}], \boldsymbol{s} \in S_\Omega^A \} \end{aligned} \tag{1.59}$$

$$\boldsymbol{F}^{-T}[\boldsymbol{s}] = \begin{pmatrix} u \\ v+u \\ w-u \end{pmatrix}. \tag{1.60}$$

Note that above relations will be satisfied for any other choice of S_Ω or S_Ω^A as long as they are coset representatives of P and $\hat{P}$. Note that [1] have errors/typo in the algorithm.

Note: Construction of S_Ω which is symmetric w.r.t. the origin is easy. However, it is not trivial to construct S_Ω^A if one wants to make it symmetric to the origin such that $-\frac{N}{2} \leq u, v, w < \frac{N}{2} - 1$ and close to the $(0, 0, 0)$. We may follow following algorithm.

(1) Any combination of (u, v, w), $u \in [0, \frac{N}{2} - 1], v, w \in [0, N-1]$ is unique.

(2) Test following cases for given (u, v, w).

$$\begin{aligned} &(u - N, v, w), (u, v - N, w), (u, v, w - N), \\ &(u - \frac{N}{2}, v - \frac{N}{2}, w - \frac{N}{2}) \end{aligned} \tag{1.61}$$

(3) Choose one of them which is closest to the origin.

[6]

$$\boldsymbol{E}_{fcs}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{F}_{fcs}^{-T} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}.$$

.

Chapter 2

BCC lattice

2.1 BCC lattice

BCC(Body-centered-cubic) lattice can be generated from the three column vectors of $\boldsymbol{U}_b$, and size of Voronoi cell is determined by k in FCC sublattice generators $\boldsymbol{V}_{bs}$.

$$\boldsymbol{U}_b = \begin{pmatrix} 2 & 0 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{V}_{bs} = k \begin{pmatrix} 2 & 0 & 2 \\ 0 & 2 & 2 \\ 2 & 2 & 0 \end{pmatrix} \tag{2.1}$$

This implies that two points $\boldsymbol{r}$ and $\boldsymbol{r}'$ are congruent, if there exits some integers $\boldsymbol{n}$ such that

$$\boldsymbol{r}' = \boldsymbol{r} + \boldsymbol{V}_{bs} \cdot \boldsymbol{n} \tag{2.2}$$

This is equivalent to the existence of integers (n_x, n_y, n_z) or (n_1, n_2, n_3) such that

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} - \begin{pmatrix} x \\ y \\ z \end{pmatrix} = 2k \begin{pmatrix} n_x + n_z \\ n_y + n_z \\ n_y + n_x \end{pmatrix} = 2k \begin{pmatrix} n_1 \\ n_2 \\ n_1 + n_2 - 2n_3 \end{pmatrix} \tag{2.3}$$

In other words, two points are congruent if,

$$\mathrm{mod}(x', 2k) = \mathrm{mod}(x, 2k), \quad \mathrm{mod}(y', 2k) = \mathrm{mod}(y, 2k), \quad \mathrm{mod}(x' + y' + z', 4k) = \mathrm{mod}(x + y + z, 4k).$$

Any BCC lattice coordinate $[\boldsymbol{r}]$ in non-orthogonal basis can be converted into Cartesian coordinates

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \boldsymbol{U}_b \cdot [\boldsymbol{r}] = \begin{pmatrix} 2[\boldsymbol{r}]^1 + [\boldsymbol{r}]^3 \\ 2[\boldsymbol{r}]^2 + [\boldsymbol{r}]^3 \\ [\boldsymbol{r}]^3 \end{pmatrix} \tag{2.4}$$

Thus, two BCC lattice coordinates c and c' are congruent if there exists integers which satisfies $U_b \boldsymbol{c}' = U_b \boldsymbol{c} + V_{bs} \cdot \boldsymbol{n}$

$$\begin{pmatrix} c^1 \\ c^2 \\ c^3 \end{pmatrix}' - \begin{pmatrix} c^1 \\ c^2 \\ c^3 \end{pmatrix} = k \begin{pmatrix} -n_2 + n_3 \\ -n_1 + n_3 \\ 2n_1 + 2n_2 \end{pmatrix} = k \begin{pmatrix} -n_1' \\ -n_2' \\ 4n_3' + 2n_1' + 2n_2' \end{pmatrix} \tag{2.5}$$

In other words, the congruency test in BCC coordinate is[1]

$$\mathrm{mod}(c^{'1}, k) = \mathrm{mod}(c^1, k), \ \mathrm{mod}(c^{'2}, k) = \mathrm{mod}(c^2, k), \quad \mathrm{mod}(2c^{'1} + 2c^{'2} + c^{'3}, 4k) = \mathrm{mod}(2c^1 + 2c^2 + c^3, 4k)$$

[1] For a comparison, the congruency test in FCC coordinate is

$$\mathrm{mod}(c^{'1}, k) = \mathrm{mod}(c^{'2}, k), \ \mathrm{mod}(c^{'2} - c^{'1}, 4k) = \mathrm{mod}(c^2 - c^1, 4k), \ \mathrm{mod}(c^{'3} - c^{'2}, 4k) = \mathrm{mod}(c^3 - c^2, 4k). \tag{2.6}$$

2.1.1 metric tensor

covariant basis vectors are,

$$\boldsymbol{u}_1 = 2\hat{e}_x \quad \boldsymbol{u}_2 = 2\hat{e}_y, \quad \boldsymbol{u}_3 = \hat{e}_x + \hat{e}_y + \hat{e}_z. \tag{2.7}$$

And contravariant basis vectors

$$\boldsymbol{u}^1 = \frac{1}{2}\hat{e}_x - \frac{1}{2}\hat{e}_z, \quad \boldsymbol{u}^2 = \frac{1}{2}\hat{e}_y - \frac{1}{2}\hat{e}_z, \quad \boldsymbol{u}^3 = \hat{e}_z. \tag{2.8}$$

Also we get metrics,[2]

$$g_{ij} = \begin{pmatrix} 4 & 0 & 2 \\ 0 & 4 & 2 \\ 2 & 2 & 3 \end{pmatrix}, \quad g^{ij} = \begin{pmatrix} \frac{1}{2} & \frac{1}{4} & -\frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & 1 \end{pmatrix} \tag{2.9}$$

2.1.2 Laplacian and others

A Laplacian in BCC lattice coordinate is

$$\begin{aligned} \nabla^2\psi &= g^{ki}\frac{\partial^2\psi}{\partial q^k \partial q^i} \\ &= \left[\frac{1}{2}(\frac{\partial}{\partial q^1})^2 + \frac{1}{2}(\frac{\partial}{\partial q^2})^2 + \frac{1}{2}(\frac{\partial^2}{\partial q^1 \partial q^2}) + (\frac{\partial}{\partial q^3})^2 - \frac{\partial^2}{\partial q^1 \partial q^3} - \frac{\partial^2}{\partial q^2 \partial q^3}\right]\psi \end{aligned} \tag{2.10}$$

Jacobian is $J = 4$, thus

$$d^3x = 4dq^1dq^2dq^3 \tag{2.11}$$

gradient (We may use Cartesian basis for spin and iso-spin)

$$\boldsymbol{\sigma} \cdot \nabla\pi = \sigma^i \frac{\partial\pi}{\partial q^i} \tag{2.12}$$

where

$$\begin{pmatrix} \sigma^1 \\ \sigma^2 \\ \sigma^3 \end{pmatrix} = \boldsymbol{U}_b^{-1} \begin{pmatrix} \sigma^x \\ \sigma^y \\ \sigma^z \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & 0 & -\frac{1}{2} \\ 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \sigma^x \\ \sigma^y \\ \sigma^z \end{pmatrix} \tag{2.13}$$

2.2 Discrete Fourier Transform on a BCC lattice

2.2.1 Construction $S_{\Omega,coef}$

A Voronoi cell Ω centered at $\mathbf{0}$ is

$$\Omega := \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} : |x| + |y| < 2k,\ |x| + |z| < 2k,\ |y| + |z| < 2k \right\} \tag{2.14}$$

The condition for BCC lattice points $\Xi = \Omega \cap L_b$ are, in terms of lattice coordinates, $\boldsymbol{c} \in S_{\Omega,coef}$,

$$\begin{aligned} &-k \le c_1 \le k, \\ &\max\{-k, c_1 - k\} \le c_2 \le \min\{k, c_1 + k\}, \\ &\max\{-k - c_1 - c_2, -k - c_1, -k - c_2\} \le c_3 \le \min\{k - c_1 - c_2, k - c_1, k - c_2\}. \end{aligned} \tag{2.15}$$

[2] $\det(g) = 16$

(In fact, as long as S becomes a coset representative of L/L_0, the choice of S or S_{coef} is arbitrary. One of the simplest way to construct $S_{\Omega,coef}$ is to choose $0 \le c_1 \le k-1$, $0 \le c_2 \le k-1$, $0 \le c_3 \le 4k-1$. However, the shape of S will not be centered on the origin and the shape becomes a skewed cubic.)

If we denote $\boldsymbol{p} = \boldsymbol{E}^{-1} \cdot c$, we have

$$\begin{aligned} & p_1 = -c_2, \quad p_2 = -c_1, \quad p_3 = 2c_1 + 2c_2 + c_3, \\ \Rightarrow \quad & -k \le p_2 \le k, \\ & -\min\{k, -p_2 + k\} \le p_1 \le -\max\{-k, -p_2 - k\}, \\ & \max\{-k + p_1 + p_2, -k + p_2, -k + p_1\} - 2p_2 - 2p_1 \\ & \quad \le p_3 \le \min\{k + p_1 + p_2, k + p_2, k + p_1\} - 2p_2 - 2p_1. \end{aligned} \tag{2.16}$$

Note: Though the range of p_i are given. not all values of p_i are independent.(p_i and p_i' is equivalent if $p_i' = p_i + D_{ii}n$.)

The lattice points with Voronoi cell, Ξ can be constructed with the algorithm given in Zheng's paper[1] which is searching BCC lattice points within Ω. The algorithm generates Ξ which have $(4k^3 + 6k^2 + 4k + 1)$ number of lattice points. However, one have to remove redundant points to obtain coset representative S_Ω of the quotient group and the explicit algorithm for $S_{\Omega,coef}$ is not given in [1]. (One choice of $S_{\Omega,coef}$ is given in the Appendix.)

After removing redundant congruent points on the surface or edge , one gets $4k^3$ number of lattice points, $S_{\Omega,coef}$,

$$S_{\Omega,coef} := \{\boldsymbol{c} : \boldsymbol{U}_b \cdot \boldsymbol{c} \in S_\Omega\}. \tag{2.17}$$

Corresponding frequency domain can be obtained by

$$S^A_{\Omega,coef} = \{A^{-1}\boldsymbol{c} : \boldsymbol{c} \in S_{\Omega,coef}\}, \quad \boldsymbol{A} = \begin{pmatrix} 0 & -1 & 1 \\ -1 & 0 & 1 \\ 1 & 1 & -1 \end{pmatrix} \tag{2.18}$$

FCC lattice case: Similar $S_{\Omega,coef}$ condition for FCC lattice is

$$\begin{aligned} & -\frac{3}{2}k \le c_1 \le \frac{3}{2}k, \\ & \max\{-\frac{3}{2}k, -2k - c_1\} \le c_2 \le \min\{\frac{3}{2}k, 2k - c_1\}, \\ & \max\{-\frac{3}{2}k - c_1 - c_2, -\frac{3}{2}k, -2k - c_1, -2k - c_2\} \le c_3 \le \min\{\frac{3}{2}k - c_1 - c_2, \frac{3}{2}k, 2k - c_1, 2k - c_2\}. \end{aligned}$$

Also, for $\boldsymbol{p} = \boldsymbol{E}^{-1} \cdot c$,

$$\begin{aligned} & p_1 = c_1, \quad p_2 = -c_1 + c_2, \quad p_3 = -c_2 + c_3, \\ \Rightarrow \quad & -\frac{3}{2}k \le p_1 \le \frac{3}{2}k, \\ & \max\{-\frac{3}{2}k, -2k - p_1\} - p_1 \le p_2 \le \min\{\frac{3}{2}k, 2k - p_1\} - p_1, \\ & \max\{-\frac{3}{2}k - 2p_1 - p_2, -\frac{3}{2}k, -2k - p_1, -2k - p_1 - p_2\} - p_1 - p_2 \\ & \quad \le p_3 \le \min\{\frac{3}{2}k - 2p_1 - 2_2, \frac{3}{2}k, 2k - p_1, 2k - p_1 - p_2\} - p_1 - p_2. \end{aligned} \tag{2.19}$$

2.2.2 algorithm of FFT on lattice

Let us suppose the indexing for position space and frequency domain is done as follows.

$$\begin{aligned} S_\Omega &:= \{\boldsymbol{r} : \text{A coset representative of quotient group } L/L_0\}, \\ S_{\Omega,coef} &:= \{[\boldsymbol{r}] : \boldsymbol{U}\cdot[\boldsymbol{r}] \in S_\Omega\}, \\ S^A_{\Omega,coef} &:= \{\boldsymbol{A}^{-1}\boldsymbol{c} : \boldsymbol{c} \in S_{\Omega,coef}\}, \\ S^A_\Omega &:= \{\boldsymbol{s} : \text{A coset representative of quotient group } L_0^*/L^*\}, \\ &= \{(\boldsymbol{V}^T)^{-1}\boldsymbol{c} : \boldsymbol{c} \in S^A_{\Omega,coef}\} \end{aligned} \tag{2.20}$$

we can express these as $3\times 4k^3$ matrix (like $S_{\Omega,coef}(1:3, 1:4k^3)$) such that each column vector corresponds to one point.

Suppose $f_{in}(l)$ is defined on $1 \le l \le 4k^3$ such that $f_{in}(l)$ is evaluated at l-th column vector of $S_{\Omega,coef}$. We can obtain corresponding DFT on BCC lattice, $f_{in} \to \hat{f}$, where $\hat{f}(l)$ is defined on l-th column vector of $S^A_{\Omega,coef}$, by using usual DFT from the fact,

$$\begin{aligned} \langle \boldsymbol{r}, \boldsymbol{s}\rangle = \boldsymbol{r}^T\cdot\boldsymbol{s} &= [\boldsymbol{r}]^T\cdot(\boldsymbol{M}^{-1})^T\cdot[\boldsymbol{s}]_{\tilde{V}} \\ &= \left(E^{-1}[\boldsymbol{r}]\right)^T D^{-1}\left((F^{-1})^T[\boldsymbol{s}]_{\tilde{V}}\right). \end{aligned} \tag{2.21}$$

For each $1 \le l \le 4k^3$, we can associate $(p,q,r) = E^{-1}[\boldsymbol{r}]_l$ with modular $(k,k,4k)$ and define $f(p,q,r) = f_{in}(l)$. (then p and q in range $(0, k-1)$ and r in range $(0, 4k-1)$). Let us denote such points as $S_{\Omega,cartes}$ and also $S^A_{\Omega,cartes}$ in frequency similarly.

$$\begin{aligned} S_{\Omega,cartes} &:= \{E^{-1}[\boldsymbol{r}] : [r] \in S_{\Omega,coef}\}, \\ S^A_{\Omega,cartes} &:= \{(F^{-1})^T[\boldsymbol{s}]_{\tilde{V}} : [s]_{\tilde{V}} \in S^A_{\Omega,coef}\} \end{aligned} \tag{2.22}$$

The FFT of f_{in} can be done with the algorithm given in [1]:

(1) For each l, find $(p,q,r) \in S_{\Omega,cartes}$ corresponding to $[\boldsymbol{r}]_l \in S_{\Omega,coef}$ and put $f_{cartes}(p,q,r) = f_{in}(l)$

(2) Do DFT as usual with $f_{cartes} \to \hat{f}_{cartes}$.

(3) For each l, find $(p,q,r) \in S^A_{\Omega,cartes}$ corresponding to $[\boldsymbol{s}]_l \in S^A_{\Omega,coef}$ and put $\hat{f}(l) = \hat{f}_{cartes}(p,q,r)$.

Some comments:

- size of lattice is determined by integer k such that number of lattice points $L = 4k^3$.
- $BCCfftBCCout(S_{\Omega,coef}, f_{in})$ returns $\hat{f}_{out}$ which is FFT components. It is equivalent to

$$\hat{f}_{out}(n) = \sum_{m=1}^{L} f_{in}(m) e^{-2\pi i [c]_m^T\cdot(M^{-1})^T\cdot[s]_n} \tag{2.23}$$

  where $[c]_m$ is a point in $S_{\Omega,coef}$ and $[s]_n$ is a point in $S^A_{\Omega,coef}$

- $iBCCfftBCCout(S_{\Omega,coef}, \hat{f}_{in})$ returns f_{re} from

$$f_{re}(n) = \frac{1}{L}\sum_{m=1}^{4k^3} \hat{f}_{in}(m) e^{2\pi i [c]_n^T\cdot(M^{-1})^T\cdot[s]_m} \tag{2.24}$$

- Note that for using FFTW algorithm, one have to divide the output with $L = 4k^3$.

- Note that the corresponding momentum $p_i = -i\nabla_{q_i}$ is from

$$\langle \boldsymbol{r}_m, \boldsymbol{s}_n \rangle = [\boldsymbol{r}]^T_m \cdot (M^{-1})^T \cdot [s]_n \tag{2.25}$$

where $[\boldsymbol{r}] \in S_{\Omega,coef}$ and $[s] \in S^A_{\Omega,coeff}$. [3] In other words, $2\pi(M^{-1})^T \cdot [s]_n$ corresponds to derivative (or momentum) $\boldsymbol{p} = -i\nabla_q$.

2.3 Numerical Derivatice in BCC lattice

For the derivative, we may use simple approximation

$$\frac{\partial \psi(\boldsymbol{q})}{\partial q^i} \simeq \frac{\psi(q^i + \Delta) - \psi(q^i - \Delta)}{2\Delta} \tag{2.27}$$

For in $\boldsymbol{u}_3$ direction, this derivative use nearest neighbor points, $\Delta = \sqrt{3}a$. However, in $\boldsymbol{u}_{1,2}$ direction, they are not using nearest neighbor points. Thus, it may be better to use those nearest neighbor points for derivative approximation.

Let us use lattice points index n_i such as $q^{1,2} = 2an_{1,2}$ and $q^3 = \sqrt{3}an_3$, from the displacement for 8 nearest neighbors are

$$\Delta\boldsymbol{n} = (0,0,1), (0,-1,1), (1,0,-1), (1,1,-1), (-1,0,1), (0,1,-1), (0,0,-1), (-1,-1,1) \tag{2.28}$$

, One possible form is

$$\begin{aligned} \frac{\partial \psi_{n_1,n_2,n_3}}{\partial q^1} \simeq \frac{1}{2a}\Big[\frac{1}{4}(\psi_{n_1,n_2,n_3+1} + \psi_{n_1,n_2-1,n_3+1} + \psi_{n_1+1,n_2,n_3-1} + \psi_{n_1+1,n_2+1,n_3-1}) \\ -\frac{1}{4}(\psi_{n_1-1,n_2,n_3+1} + \psi_{n_1,n_2+1,n_3-1} + \psi_{n_1,n_2,n_3-1} + \psi_{n_1-1,n_2-1,n_3+1})\Big] \end{aligned} \tag{2.29}$$

while

$$\frac{\partial \psi_{n_1,n_2,n_3}}{\partial q^3} \simeq \frac{1}{2\sqrt{3}a}\left[\psi_{n1,n2,n3+1} - \psi_{n1,n2,n3-1}\right] \tag{2.30}$$

It may be more easier to use DFT to compute derivatives. (Obtain $\hat{f}_k$ from DFT of $f(n)$. Multiply momentum factors to $\hat{f}_k$. Then inverse DFT to get $\nabla f(n)$.) However, it should be noted that computing derivative using DFT only works for periodic functions. If the function does not satisfy the periodic condition, the results would have errors.

2.4 Free Hamiltonian in BCC lattice

The free action becomes

$$\begin{aligned} & \int dt \int_V d^3x \frac{-1}{2m}\psi^\dagger(x)\nabla^2\psi(x) \\ = & \int dt \int_{S_{\Omega,coeff}} 4d^3q \frac{-1}{2m}\psi^\dagger(\boldsymbol{q})\left[\frac{1}{2}(\frac{\partial}{\partial q^1})^2 + \frac{1}{2}(\frac{\partial}{\partial q^2})^2 + \frac{1}{2}(\frac{\partial^2}{\partial q^1 \partial q^2}) + (\frac{\partial}{\partial q^3})^2 - \frac{\partial^2}{\partial q^1\partial q^3} - \frac{\partial^2}{\partial q^2 \partial q^3}\right]\psi(\boldsymbol{q}) \\ \to & \; a_t \sum_{n_t}(4a^3)\sum_{\boldsymbol{q}_n}\frac{-1}{2m}\psi^\dagger(\boldsymbol{q}_n)\left[\frac{1}{2}(\frac{\partial}{\partial q^1})^2 + \frac{1}{2}(\frac{\partial}{\partial q^2})^2 + \frac{1}{2}(\frac{\partial^2}{\partial q^1 \partial q^2}) + (\frac{\partial}{\partial q^3})^2 - \frac{\partial^2}{\partial q^1\partial q^3} - \frac{\partial^2}{\partial q^2 \partial q^3}\right]\psi(\boldsymbol{q}_n) \end{aligned}$$

[3] In curvilinear coordinate convention,

$$(q^1, q^2, q^3) \leftrightarrow [\boldsymbol{r}], \quad (p_1, p_2, p_3) = -i(\frac{\partial}{\partial q^1}, \frac{\partial}{\partial q^2}, \frac{\partial}{\partial q^3}) \leftrightarrow 2\pi(M^{-1})^T \cdot [s] \tag{2.26}$$

After Fourier transformation, this becomes

$$\rightarrow \quad a_t \sum_{n_t} (4a^3)(L) \sum_{\boldsymbol{p}_n} \frac{1}{2m} \hat{\psi}^\dagger(\boldsymbol{p}_n)(\frac{1}{2}p_{n,1}^2 + \frac{1}{2}p_{n,2}^2 + \frac{1}{2}p_{n,1}p_{n,2} + p_{n,3}^2 - p_{n,1}p_{n,3} - p_{n,2}p_{n,3})\hat{\psi}(\boldsymbol{p}_n)$$

where ($L = 4k^3$) factor can be absorbed into $\hat{\psi}^\dagger(p)$ and $\hat{\psi}(p)$ depending on the convention of DFT.

We may compute free action by using DFT,(instead of approximate numerical derivative), as follows

- Define

$$\Phi(\boldsymbol{q}_n) \equiv -\left[\frac{1}{2}(\frac{\partial}{\partial q^1})^2 + \frac{1}{2}(\frac{\partial}{\partial q^2})^2 + \frac{1}{2}(\frac{\partial^2}{\partial q^1 \partial q^2}) + (\frac{\partial}{\partial q^3})^2 - \frac{\partial^2}{\partial q^1 \partial q^3} - \frac{\partial^2}{\partial q^2 \partial q^3}\right] \psi(\boldsymbol{q}_n) \quad (2.31)$$

- First do FFT of $\psi(\boldsymbol{q}_n)$, to get $\hat{\psi}(\boldsymbol{p}_m)$

$$\hat{\psi}(\boldsymbol{p}_n) = \sum_{\boldsymbol{q}_n} \psi(\boldsymbol{q}_n) e^{-i\boldsymbol{q}_n \cdot \boldsymbol{p}_n}, \quad \psi(\boldsymbol{q}_n) = \frac{1}{L} \sum_{\boldsymbol{p}_n} \hat{\psi}(\boldsymbol{p}_n) e^{i\boldsymbol{q}_n \cdot \boldsymbol{p}_n} \quad (2.32)$$

- compute

$$\begin{aligned} \hat{\Phi}(\boldsymbol{p}_n) &= p_i g^{ij} p_j \hat{\psi}(\boldsymbol{p}_n) \\ &= \left((\frac{p_1}{2})^2 + (\frac{p_2}{2})^2 + (-\frac{p_1}{2} - \frac{p_2}{2} + p_3)^2\right) \hat{\psi}(\boldsymbol{p}_n) \end{aligned} \quad (2.33)$$

- inverse transform $\hat{\Phi}(\boldsymbol{p}_n)$, to get $\Phi(\boldsymbol{q}_n)$

$$\Phi(\boldsymbol{q}_n) = \frac{1}{L} \sum_{\boldsymbol{p}_n} \hat{\Phi}(\boldsymbol{p}_n) e^{i\boldsymbol{q}_n \cdot \boldsymbol{p}_n} \quad (2.34)$$

- Then free particle action becomes

$$\int dt \int_V d^3x \frac{-1}{2m} \psi^\dagger(x) \nabla^2 \psi(x) \quad \rightarrow \quad a_t \sum_{n_t} (4a^3) \sum_{\boldsymbol{q}_n} \frac{1}{2m} \psi^\dagger(\boldsymbol{q}_n) \Phi(\boldsymbol{q}_n) \quad (2.35)$$

- Note the factor $4a^3$ in the action. This implies that when we compute the action we have to multiply $4a^3$ for the H_{free} term. This factor will appear in every part of the action. But it is possible to absorb this factor in the normalization of fields $\sqrt{4a^3}\psi(x) \rightarrow \psi(x)$ which only change overall normalization in path integral $\int \mathcal{D}\psi$. Thus, it does not affect the lattice calculation.

2.4.1 Dispersion relation of a free particle on lattice

Let us consider a spectrum of free particle in a periodic BCC lattice.

Obviously, the lowest energy free particle state is $\boldsymbol{p} = \boldsymbol{0}$ with $p^2 = 0$.

From the periodicity condition, possible values of $\boldsymbol{p} = (p_x, p_y, p_z)$ must satisfy for any integers $\boldsymbol{n} = (n_1, n_2, n_3)$,

$$\boldsymbol{p}^T \cdot \boldsymbol{V}\boldsymbol{n} = 2\pi(\text{integer}). \quad (2.36)$$

This condition can be satisfied for BCC lattice if

$$p_x + p_z = \frac{2\pi}{2k}(\text{integer}), \quad p_y + p_z = \frac{2\pi}{2k}(\text{integer}), \quad 2p_z = \frac{2\pi}{2k}(\text{integer}). \quad (2.37)$$

In other words, momentums in BCC lattice are

$$\boldsymbol{p} = \frac{2\pi}{4k}(m_x, m_y, m_z), \quad p^2 = (\frac{2\pi}{4k})^2(m_x^2 + m_y^2 + m_z^2) \tag{2.38}$$

where integers $m_{x,y,z}$ have the same parity(all even or all odd).

Similary, for FCC lattice, the condition can be satisfied if

$$2p_x = \frac{2\pi}{2k}(\text{integer}), \quad 2p_y = \frac{2\pi}{2k}(\text{integer}), \quad p_x + p_y + p_z = \frac{2\pi}{2k}(\text{integer}). \tag{2.39}$$

In other words, momentums in FCC lattice are

$$\boldsymbol{p} = \frac{2\pi}{4k}(m_x, m_y, m_z) \tag{2.40}$$

where $m_x + m_y + m_z$ is an even number.

Thus, first excited state of free particle in BCC have $p^2 = 3(\frac{2\pi}{4k})^2$ while $p^2 = 2(\frac{2\pi}{4k})^2$ for FCC, $p^2 = (\frac{2\pi}{k})^2$ for cubic. (In the code, one can find all possible momentum values (p_1, p_2, p_3) from $S_{freq,coef}$.)

By using this fact, one can construct a periodic initial wave function in BCC(FCC) lattice coordinates , with momentum $\boldsymbol{p} = (p_x, p_y, p_z)$ satisfying above conditions,

$$e^{i\boldsymbol{r}^T \cdot \boldsymbol{p}} = e^{i[\boldsymbol{q}]^T \cdot \boldsymbol{U}^T \boldsymbol{p}} \tag{2.41}$$

Or, more simply, one can use a symmetric function $f(\boldsymbol{r})$ centered at the origin with vanishing tail for an initial wave with zero momentum .

Chapter 3

Numerical realization in nuclear lattice code

Unlike the simple Cartesian lattice, in which $S_\Omega = S_{\Omega,coef}$, for BCC or FCC, one have to choose how the single particle wave function of nucleon or auxiliary particle(and pion) configuration should be represented in the code. There may be three possible choices(Also for frequency domain).

- $\boldsymbol{r} \in S_\Omega$: This is the same as the Cartesian lattice, but all functions $f(\boldsymbol{r})$ have to be evaluated only on S_Ω. **Note:** To work with DFT on lattice, the function should satisfy the periodicity. Thus, one have to be careful to define initial waves on lattice.

- $[\boldsymbol{r}] \in S_{\Omega,coef}$: One construct a index matrix $3\times$(lattice size), $\boldsymbol{S}_{\Omega,coef}$, and evaluate functions $g([\boldsymbol{r}]_l)$ at l-th column vectors of index matrix. (Or in Cartesian coordinates, $g([\boldsymbol{r}]_l) = f(\boldsymbol{r}_l)$ This would be a natural choice of BCC lattice formulation.

I will use the $[\boldsymbol{r}] \in S_{\Omega,coef}$ for coordinates in the lattice code. As a test case, let us modify the **nuclei** code to use **lattice_fft** codes for a cubic lattice. The key subroutine of **nuclei** for transfer matrix calculation is **getzvecs** and **getzdualvecs**. For cubic lattice, by adding a layer of conversion in front and end of using **lattice_fft** code, I can replace these subroutines into a **lattice_fft** subroutines.

Though **nuclei** and **lattice_fft** use both the cubic lattice , because they use different range of index, one needs conversion.

$$\begin{aligned} &\text{zvecs}(n_x, n_y, n_z) \leftrightarrow \text{zvecs}(l), \\ &n_{x,y,z} \to q_{1,2,3} = n_{x,y,z} - L * \text{int}(2 * n_{x,y,z}/L), \\ &q_{1,2,3} \to n_{x,y,z} = \text{mod}(q_{1,2,3} + L, L). \end{aligned} \tag{3.1}$$

3.1 init_lattice

The **init_lattice** initialize the lattice.

- lattice_type and lsize_k gives lattice_size

type	description	lattice_size	1d-period	Voronoi cell
1	cubic with cubic boundary	k^3	k	cube
2	BCC with FCC boundary	$4k^3$		rhombic dodecahedron
3	FCC with BCC boundary	$16k^3$		truncated octahedron
4	BCC with cubic boundary	$2k^3$	$2k$	cube
5	FCC with cubic boundary	$4k^3$	$2k$	cube

- nSomega : list of Cartesian coordinates of non-redundant lattice points. The origin is at the center.
- nSomega_coef : list of coordinates in lattice generating vector basis of lattice points.
- get_congruent : return lattice index which is equivalent to the input lattice point.
- ...

3.2 getzvecs

- The free action in **nuclei** code used "improved action" while it is better to use FFT in **lattice_fft** code. Thus the exact numerical comparison requires additional consideration of dispersion relation.
- Discussion on center of mass coordinate on lattice is necessary.
- For the free action part, it is calculated in following way. Assume a s.p. state $|\boldsymbol{n}\rangle$, $|f\rangle$, s.p. wave functions $f(\boldsymbol{n})$ such that

$$|\boldsymbol{n}\rangle = a^\dagger(\boldsymbol{n})|0\rangle, \quad \langle \boldsymbol{n}|f\rangle = \langle 0|a(\boldsymbol{n})|f\rangle = f(\boldsymbol{n}). \tag{3.2}$$

  Then, the transfer matrix element for free action gives

$$\langle \boldsymbol{n}|M|f\rangle \simeq \langle \boldsymbol{n}| : \left(1 - \frac{1}{2m}\sum_{\boldsymbol{n}'} \rho(\boldsymbol{n}')(-\nabla^2_{\boldsymbol{n}'})\right) : |f\rangle = f(\boldsymbol{n}) - g(\boldsymbol{n}) \tag{3.3}$$

  where

$$\begin{aligned} g(\boldsymbol{n}) &\equiv \frac{1}{|G|^2}\sum_{[\boldsymbol{s}]} e^{2\pi i\langle \boldsymbol{q},\boldsymbol{s}\rangle}\frac{\Pi^2(\boldsymbol{s})}{2m}\hat{f}(\boldsymbol{s}), \\ \hat{f}(\boldsymbol{s}) &\equiv \sum_{[\boldsymbol{n}]} e^{-2\pi i\langle \boldsymbol{n},\boldsymbol{s}\rangle} f(\boldsymbol{n}), \quad \Pi^2(\boldsymbol{s}) \equiv g^{ij}p_i p_j, \quad p_i = \left(2\pi(\boldsymbol{M}^{-1})^T[\boldsymbol{s}]\right)_i \end{aligned} \tag{3.4}$$

- the Wall potential,**Vwall** (V_W), is included as like **nuclei** code,

$$\langle \boldsymbol{n}| - \sum_{\boldsymbol{n}'} V_W(\boldsymbol{n}')|f\rangle = -V_W(\boldsymbol{n})f(\boldsymbol{n}). \tag{3.5}$$

  This wall potential is an external potential, not between nucleons.

- Since the change of lattice, all interaction parts also have to be reformulated. For non-local SU(4) interaction, one have to change the definition $\rho_{NL}(\boldsymbol{n})$ and coupling strength. For example, we may use similar definition like,

$$a_{NL}(\boldsymbol{n}) = a(\boldsymbol{n}) + s_{NL}\sum_{\langle \boldsymbol{n}',\boldsymbol{n}\rangle} a(\boldsymbol{n}'), \quad \rho_{NL}(\boldsymbol{n}) = a^\dagger_{NL}(\boldsymbol{n})a_{NL}(\boldsymbol{n}). \tag{3.6}$$

  with nearest neighbors of a lattice point. However, nearest neighbors are 6(12,8) points for cubic(FCC,BCC) lattice. Need to compute

$$\langle \boldsymbol{n}| : \sum_{\boldsymbol{n}',\boldsymbol{n}''} \rho_{NL}(\boldsymbol{n}')f_{s_L}(\boldsymbol{n}' - \boldsymbol{n}'')s(\boldsymbol{n}'') : |f\rangle. \tag{3.7}$$

Define

$$\begin{aligned} s_{smeared}(\boldsymbol{n}) &= \sum_{\boldsymbol{n}'} f_{s_L}(\boldsymbol{n}-\boldsymbol{n}')s(\boldsymbol{n}') = s(\boldsymbol{n}) + \sum_{\langle \boldsymbol{n}',\boldsymbol{n}\rangle} s_L s(\boldsymbol{n}'), \\ f_{smeared}(\boldsymbol{n}) &= f(\boldsymbol{n}) + \sum_{\langle \boldsymbol{n}',\boldsymbol{n}\rangle} s_{NL} f(\boldsymbol{n}') \end{aligned} \tag{3.8}$$

Then,

$$\begin{aligned} & \langle \boldsymbol{n}| : \sum_{\boldsymbol{n}',\boldsymbol{n}''} \rho_{NL}(\boldsymbol{n}') f_{s_L}(\boldsymbol{n}'-\boldsymbol{n}'')s(\boldsymbol{n}'') : |f\rangle \\ = & \sum_{\boldsymbol{n}'} \left(\delta_{\boldsymbol{n}\boldsymbol{n}'} + s_{NL} \sum_{\langle \boldsymbol{n}'',\boldsymbol{n}'\rangle} \delta_{\boldsymbol{n},\boldsymbol{n}''} \right) s_{smeared}(\boldsymbol{n}') f_{smeared}(\boldsymbol{n}') \\ = & s_{smeared}(\boldsymbol{n}) f_{smeared}(\boldsymbol{n}) + s_{NL} \sum_{\langle \boldsymbol{n}',\boldsymbol{n}\rangle} s_{smeared}(\boldsymbol{n}') f_{smeared}(\boldsymbol{n}') \end{aligned} \tag{3.9}$$

Displacements for nearest neighbors for a BCC lattice point are

$$\begin{aligned} \Delta \boldsymbol{n}_{BCC} &= (-1,0,1),(0,0,1),(0,-1,1),(-1,-1,1), \\ & \quad (0,1,-1),(1,1,-1),(1,0,-1),(0,0,-1), \end{aligned} \tag{3.10}$$

, and for a FCC lattice point are

$$\begin{aligned} \Delta \boldsymbol{n}_{FCC} &= (-1,0,0),(-1,0,1),(-1,1,0),(0,-1,0),(0,-1,1),(0,0,-1), \\ & \quad (0,0,1),(0,1,-1),(0,1,0),(1,-1,0),(1,0,-1),(1,0,0) \end{aligned} \tag{3.11}$$

- For pion coupling, **nuclei** used 1-d **q_hop** function in cubic lattice. However, we need to use FFT of pion for the pion coupling term for general lattice(there is no simple delta function relation in each direction for other lattice.)

$$\begin{aligned} & \langle \boldsymbol{n}, s_1, i_1| : \sum_{\boldsymbol{n}'} \rho_{SI}(\boldsymbol{n}') \nabla_S \pi_I(\boldsymbol{n}') : |f\rangle \\ = & \sum_{s_1 s_2} \sum_{i_1 i_2} f(\boldsymbol{n}, s_2, i_2) [\boldsymbol{\sigma}_S]_{s_1 s_2} [\tau_I]_{i_1 i_2} \nabla_S \pi_I(\boldsymbol{n}) \end{aligned} \tag{3.12}$$

where we need to connect Cartesian spin matrix with derivative of lattice coordinate,

$$\boldsymbol{\sigma} \cdot \nabla = \boldsymbol{\sigma}_i \frac{\partial}{\partial r^i} = \boldsymbol{\sigma}_i (U^{-1})^T_{ij} \frac{\partial}{\partial q^j} \tag{3.13}$$

Thus this can be computed as

$$\begin{aligned} & \langle \boldsymbol{n}, s_1, i_1| : \sum_{\boldsymbol{n}'} \rho_{SI}(\boldsymbol{n}') \nabla_S \pi_I(\boldsymbol{n}') : |f\rangle \\ = & \sum_{s_2, i_2} \sum_{S,I} [\boldsymbol{\sigma}_S]_{s_1 s_2} [\tau_I]_{i_1 i_2} f(\boldsymbol{n}, s_2, i_2) \phi_{SI}(\boldsymbol{n}). \end{aligned} \tag{3.14}$$

by using the FFT of pion, $\hat{\pi}_I([\boldsymbol{s}])$, with[1]

$$\begin{aligned}\phi_{SI}(\boldsymbol{n}) &= \frac{1}{|G|^2}\sum_{[\boldsymbol{s}]} e^{2\pi i\langle \boldsymbol{n},\boldsymbol{s}\rangle}(i\boldsymbol{k}_S)\hat{\pi}_I([\boldsymbol{s}]),\\ \boldsymbol{k}_S &= 2\pi\left(\boldsymbol{U}^{-1,T}\boldsymbol{M}^{-1,T}[\boldsymbol{s}]\right)_S \end{aligned} \tag{3.16}$$

It will be convenient to prepare $\phi_{SI}(\boldsymbol{n})$ at the same time as preparing true-pion $\pi_I(\boldsymbol{n})$. Also, it should be noted that the relation between true-pion π_I and auxiliary pion ϕ_I should be modified such that

$$\hat{\pi}_I(\boldsymbol{p}) = \frac{\hat{\phi}_I(\boldsymbol{p})}{\sqrt{p_i g^{ij} p_j + m_\pi^2}}. \tag{3.17}$$

The calculation can be broken down into steps

$$\langle \boldsymbol{n}, s_1, i_1| : \sum_{\boldsymbol{n}'} \rho_{SI}(\boldsymbol{n}')\nabla_S \pi_I(\boldsymbol{n}') : |f\rangle = \sum_{s_2,i_2} \left[\Xi(\boldsymbol{n})\right]_{s_1 s_2, i_1 i_2} f(\boldsymbol{n}, s_2, i_2) \tag{3.18}$$

where,

$$\begin{aligned}\left[\boldsymbol{\Xi}(\boldsymbol{n})\right]_{s_1 s_2, i_1 i_2} &= \sum_I \left[\Phi_I(\boldsymbol{n})\right]_{s_1 s_2}\left[\tau_I\right]_{i_1 i_2}\\ \left[\boldsymbol{\Phi}_I(\boldsymbol{n})\right]_{s_1 s_2} &= \sum_S \left[\sigma_S\right]_{s_1 s_2}\phi_{SI}(\boldsymbol{n}).\end{aligned} \tag{3.19}$$

3.3 waveinit

If a function does not satisfy the periodic condition, the derivative calculation by using DFT/FFT will give wrong results. Also, since we want to work in center of mass frame, it is important to prepare initial single particle waves so that they are periodic and total momentum of system is zero.

This can be achieved if one use explicit momentum eigen states, $e^{\pm i\boldsymbol{r}\cdot\boldsymbol{p}}$ on the lattice with choosing set of momentum values which gives total momentum of the system zero and also satisfies periodic boundary conditions. (Details in section 2.4.1.) For cubic lattice, $(p_x, p_y, p_z) = \frac{2\pi}{k}(m_x, m_y, m_z)$ with integers $m_{x,y,z}$. For BCC lattice, $(p_x, p_y, p_z) = \frac{2\pi}{4k}(m_x, m_y, m_z)$ where all $m_{x,y,z}$ have the same parity. For FCC lattice, $(p_x, p_y, p_z) = \frac{2\pi}{4k}(m_x, m_y, m_z)$ where $m_x + m_y + m_z$ is an even number.

Another way is to choose central peaked symmetric function $f(\boldsymbol{r}) = f(\boldsymbol{U}[\boldsymbol{q}])$ which vanish at the boundary.

3.4 HMC update

For HMC update, we need to compute derivative over auxiliary field

$$\frac{\partial V(s,\pi_I)}{\partial s(\boldsymbol{n}, n_t)}, \quad \frac{\partial V(s,\pi_I)}{\partial \phi_I(\boldsymbol{n}, n_t)} \tag{3.20}$$

[1] In case of cubic lattice, suppose i, j are different direction with S. Then,

$$\begin{aligned}\phi_{SI}(\boldsymbol{n}) &= \frac{1}{L^3}\sum_{\boldsymbol{m}} e^{i\frac{2\pi}{L}\boldsymbol{m}\cdot\boldsymbol{n}}(i\frac{2\pi}{L}m_S)\hat{\pi}_I(\boldsymbol{m})\\ &= \frac{1}{L^3}\sum_{\boldsymbol{m}}\sum_{\boldsymbol{n}'} e^{i\frac{2\pi}{L}\boldsymbol{m}\cdot(\boldsymbol{n}-\boldsymbol{n}')}(i\frac{2\pi}{L}m_S)\pi_I(\boldsymbol{n}')\\ &= \frac{1}{L}\sum_{n'_S}\sum_{m_S} e^{-i\frac{2\pi}{L}m_S n'_S}(i\frac{2\pi}{L}m_S)\pi_I(n_i, n_j, n_S + n'_S)\\ &= \sum_{n'_S}\Delta_S(n'_S)\pi_I(n_i, n_j, n_S + n'_S)\end{aligned} \tag{3.15}$$

This requires

$$\sum_S \sum_{\boldsymbol{n}'} \left(\frac{\partial \nabla_S \pi_I(\boldsymbol{n}', n_t)}{\partial \phi_I(\boldsymbol{n}, n_t)} \right) \left(\frac{\partial V(s, \pi_I)}{\partial \nabla_S \pi_I(\boldsymbol{n}', n_t)} \right) \tag{3.21}$$

The $\nabla_S \pi_I(\boldsymbol{n})$ may be obtained by using the FFT and iFFT, in another way, we may introduce a 3-d **q_hop** function(matrix) $\Delta_S(\boldsymbol{n}, \boldsymbol{n}')$, (in cubic lattice),

$$\begin{aligned}
\nabla_S \pi_I(\boldsymbol{n}) &= \frac{1}{L^3} \sum_{\boldsymbol{q}} e^{i\boldsymbol{q}\cdot\boldsymbol{n}} (i\boldsymbol{q})_S \sum_{\boldsymbol{n}'} e^{-i\boldsymbol{q}\cdot\boldsymbol{n}'} \pi_I(\boldsymbol{n}') = \frac{1}{L^3} \sum_{\boldsymbol{n}'} \pi_I(\boldsymbol{n}') \sum_{\boldsymbol{q}} e^{i\boldsymbol{q}\cdot(\boldsymbol{n}-\boldsymbol{n}')} (i\boldsymbol{q})_S \\
&= \sum_{\boldsymbol{n}'} \pi_I(\boldsymbol{n}') \Delta_S(\boldsymbol{n}' - \boldsymbol{n}) = \sum_{\boldsymbol{n}'} \pi_I(\boldsymbol{n} + \boldsymbol{n}') \Delta_S(\boldsymbol{n}'), \\
\Delta_S(\boldsymbol{n} - \boldsymbol{n}') &= \frac{1}{L^3} \sum_{\boldsymbol{q}} e^{-i\boldsymbol{q}\cdot(\boldsymbol{n}-\boldsymbol{n}')} (i\boldsymbol{q})_S = -\Delta_S(\boldsymbol{n}' - \boldsymbol{n}).
\end{aligned} \tag{3.22}$$

Thus, the transformation $\pi_I \to \nabla_S \pi_I$ becomes a matrix multiplication,

$$\nabla_S \pi_I(\boldsymbol{n}) = -\sum_{\boldsymbol{n}'} \Delta_S(\boldsymbol{n}, \boldsymbol{n}') \pi_I(\boldsymbol{n}') \tag{3.23}$$

In a similar way, similar transformation matrix can be defined for transformation $\phi_I \to \pi_I$,

$$\begin{aligned}
\pi_I(\boldsymbol{n}) &= \sum_{\boldsymbol{n}'} K(\boldsymbol{n}, \boldsymbol{n}') \phi_I(\boldsymbol{n}') \\
K(\boldsymbol{n}, \boldsymbol{n}') &= \frac{1}{L^3} \sum_{\boldsymbol{q}} e^{-i\boldsymbol{q}\cdot(\boldsymbol{n}-\boldsymbol{n}')} \frac{1}{\sqrt{\boldsymbol{q}^2 + m_\pi^2}}
\end{aligned} \tag{3.24}$$

In other words, transform $\phi_I \to \nabla_S \pi_I$ can be done as a matrix multiplication, (the matrix can be large size)

$$\nabla_S \pi_I(\boldsymbol{n}) = -\left(\boldsymbol{\Delta}_S \boldsymbol{K} \boldsymbol{\phi}_I\right)_{\boldsymbol{n}}. \tag{3.25}$$

This has advantage when computing derivative of HMC potential since we can use

$$\frac{\partial \nabla_S \pi_{I'}(\boldsymbol{n}')}{\partial \phi_I(\boldsymbol{n})} = -\left(\boldsymbol{\Delta}_S \boldsymbol{K}\right)_{\boldsymbol{n}', \boldsymbol{n}} \delta_{II'} \tag{3.26}$$

Once $\Delta_S(\boldsymbol{n}, \boldsymbol{n}')$ and $\boldsymbol{K}(\boldsymbol{n}, \boldsymbol{n}')$ is computed for a lattice, it can be used repeatedly and does not requires FFT to get π_I, $\nabla_S \pi_I$ and also derivative of HMC. (Instead one have to multiply matrix to get those.) Also, because only difference $\boldsymbol{n} - \boldsymbol{n}'$ is relevant, one can store $\Delta_S(\boldsymbol{n})$, $\boldsymbol{K}(\boldsymbol{n})$ instead of matrix. This can be easily generalized to other lattice (FCC,BCC).

However, for the moment, to keep the original code structure as much as possible, I will use following form, (dV is an output from Subroutine **dV**)

$$\begin{aligned}
\frac{\partial V(s, \pi_I)}{\partial \phi_I(\boldsymbol{n}, n_t)} &= \phi_I(\boldsymbol{n}, n_t) - \sum_S \sum_{\boldsymbol{n}'} \frac{\partial \nabla_S \pi_I(\boldsymbol{n}', n_t)}{\partial \phi_I(\boldsymbol{n}, n_t)} (-\frac{g_A \alpha_t}{2 f_\pi}) dV(\boldsymbol{n}, n_t, S, I) \\
&= \phi_I(\boldsymbol{n}, n_t) + \frac{g_A \alpha_t}{2 f_\pi} \sum_S \frac{1}{L^3} \sum_{\boldsymbol{q}} e^{i\boldsymbol{q}\cdot\boldsymbol{n}} (-i\boldsymbol{q})_S \frac{\widehat{dV}(\boldsymbol{q}, n_t, S, I)}{\sqrt{\boldsymbol{q}^2 + m_\pi^2}}.
\end{aligned} \tag{3.27}$$

Instead of using Δ_S or $\boldsymbol{K}$ matrix, I will directly inverse FFT transform $(-i\boldsymbol{q})_S \frac{\widehat{dV}(\boldsymbol{q}, n_t, S, I)}{\sqrt{\boldsymbol{q}^2 + m_\pi^2}}$.

3.5 pinhole

By pinhole algorithm, one can obtain the density distribution of a nuclei or configurations of clusters. For A-nucleons, let us represent i-th pinhole position and spin-isospin as $\boldsymbol{r}_i$, s_i, t_i for a given pinhole configuration within S_Ω. One wants to find the corresponding center of pinholes in S_Ω and distances of each pinholes from the center. However, because of periodic boundary condition, it is not obvious.

Refer the previous chapters for the distance calculations of two points.

To find a center of points, one first note that simple numerical form

$$\boldsymbol{R} = \frac{1}{A}\sum_{i=1}^{A} \boldsymbol{r}_i, \tag{3.28}$$

does not work. (For example, for 1-d lattice points $x = -2, -1, 0, 1$ with period $L = 4$, The correct center of two points $x = -2$ and $x = 1$ is $\frac{3}{2}$ instead of $-\frac{1}{2}$.) Thus, we may define the center of points such that the average distance to all points from that point becomes minimum.

Be careful that $\boldsymbol{R}$ can be a real value even when $\boldsymbol{r}_i$ are integer values($A\boldsymbol{R}$ will be integer values).

In case of cubic periodic boundary condition, where $\boldsymbol{V}$ is diagonal, one can search the center of points in each direction separately. In other words, one can search R_x by minimize the distance squares in x-direction, regardless R_y and R_z. (Also the distance calculation in each direction is easy).

One may construct a table for lattice index with scaled by A(in other words, change lattice unit to a/A lattice units) and try every points to minimize the distance square to pinhole locations. However, as increasing number of particles and size of lattice, it becomes very inefficient and slow. Thus, only cubic periodic boundary conditions will be used for the pinhole algorithm.

Chapter 4

Appendix

4.1 Curvilinear coordinate

Here, the coordinate transformation to curvilinear coordinate is summarized. (Main reference is Arfken's book).

Covariant and Contravariant vectors are defined, according to their transformation, as

$$\begin{aligned} A'_i &= \sum_j \frac{\partial x'_i}{\partial x_j} A_j \quad \text{(contravariant)}, \\ (\nabla\phi)'_i &= \sum_j \frac{\partial x_j}{\partial x'_i}\frac{\partial \phi}{\partial x_j} \quad \text{(covariant)}. \end{aligned} \tag{4.1}$$

Let us use upper index for contravariant components and lower index for covariant components.

A position vector can be written in terms of covariant basis vectors or contravariant basis vectors,

$$\begin{aligned} \boldsymbol{r} &= x\hat{\boldsymbol{e}}_x + y\hat{\boldsymbol{e}}_y + z\hat{\boldsymbol{e}}_z \\ &= q^1\boldsymbol{u}_1 + q^2\boldsymbol{u}_2 + q^3\boldsymbol{u}_3 \\ &= q_1\boldsymbol{u}^1 + q_2\boldsymbol{u}^2 + q_3\boldsymbol{u}^3. \end{aligned} \tag{4.2}$$

(Note that the combination of contravariant components and covariant basis vectors makes the vector itself does not change under coordinate transformation.)

The covariant basis vector can be written as

$$\boldsymbol{u}_i = \frac{\partial \boldsymbol{r}}{\partial q^i} = \frac{\partial x}{\partial q^i}\hat{\boldsymbol{e}}_x + \frac{\partial y}{\partial q^i}\hat{\boldsymbol{e}}_y + \frac{\partial z}{\partial q^i}\hat{\boldsymbol{e}}_z. \tag{4.3}$$

The contravariant basis vectors is , (note $\boldsymbol{u}^i\boldsymbol{u}_j = \delta^i_j$),

$$\boldsymbol{u}^i = \frac{\partial q^i}{\partial \boldsymbol{r}} = \frac{\partial q^i}{\partial x}\hat{\boldsymbol{e}}_x + \frac{\partial q^i}{\partial y}\hat{\boldsymbol{e}}_y + \frac{\partial q^i}{\partial z}\hat{\boldsymbol{e}}_z \tag{4.4}$$

A distance can be written as

$$\begin{aligned} dx &= \frac{\partial x}{\partial q^1}dq^1 + \frac{\partial x}{\partial q^2}dq^2 + \frac{\partial x}{\partial q^3}dq^3, \\ ds^2 &= (dx)^2 + (dy)^2 + (dz)^2 = \sum_{ij} g_{ij}dq^i dq^j \end{aligned} \tag{4.5}$$

where metric tensor

$$g_{ij} = \boldsymbol{u}_i \cdot \boldsymbol{u}_j = \frac{\partial x}{\partial q_i}\frac{\partial x}{\partial q_j} + \frac{\partial y}{\partial q_i}\frac{\partial y}{\partial q_j} + \frac{\partial z}{\partial q_i}\frac{\partial z}{\partial q_j} \tag{4.6}$$

A scalar product of two vector can be defined as

$$\boldsymbol{A}\cdot\boldsymbol{B}=A^iB_i=A_iB^i \tag{4.7}$$

Gradient of a scalar function is

$$\nabla\psi=\frac{\partial\psi}{\partial q^i}\boldsymbol{u}^i=\left(\frac{\partial\psi}{\partial q^i}g^{ij}\right)\boldsymbol{u}_j, \tag{4.8}$$

where the last expression is written with covariant basis vectors.

Divergence of a vector requires Christoffel symbol Γ,

$$\begin{aligned}\nabla\cdot\boldsymbol{V} &= \boldsymbol{u}^j\cdot(\frac{\partial}{\partial q^j}(V^i\boldsymbol{u}_j))=\frac{\partial V^i}{\partial q^i}+V^k\Gamma^i_{ik},\\ \Gamma^i_{ik} &= \frac{1}{2}g^{im}\frac{\partial g_{im}}{\partial q^k}.\end{aligned} \tag{4.9}$$

In more compact form, it can be written as

$$\nabla\cdot\boldsymbol{V}=\frac{1}{[\det(g)]^{1/2}}\frac{\partial}{\partial q^k}\left([\det(g)]^{1/2}V^k\right) \tag{4.10}$$

A Laplacian is

$$\nabla^2\psi=\frac{1}{[\det(g)]^{1/2}}\frac{\partial}{\partial q^k}\left([\det(g)]^{1/2}g^{ki}\frac{\partial\psi}{\partial q^i}\right) \tag{4.11}$$

Volume element is

$$d\tau=Jdq^1dq^2dq^3,\quad J=\frac{\partial(x,y,z)}{\partial(q^1,q^2,q^3)}=\det J_{ij}=\sqrt{\det(g_{ij})}. \tag{4.12}$$

4.1.1 Metrics in lattice

- Metric: By using the generator matrix $\boldsymbol{U}$ and its inverse $\boldsymbol{U}^{-1}$, we find[1]

$$\boldsymbol{r}_i=U_{ij}\boldsymbol{q}^j,\quad \boldsymbol{q}^i=(U^{-1})_{ij}\boldsymbol{r}_j\quad\rightarrow\quad \frac{\partial r_i}{\partial q^j}=U_{ij},\quad \frac{\partial q^i}{\partial r_j}=(U^{-1})_{ij}. \tag{4.14}$$

  we can find the metric tensors for Cubic(c), FCC(f) and BCC(b), as

$$\begin{aligned}g_{ij}(c) &= \begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix},\quad g^{ij}(c)=\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix},\\ g_{ij}(f) &= \begin{pmatrix}2&1&1\\1&2&1\\1&1&2\end{pmatrix},\quad g^{ij}(f)=\begin{pmatrix}3/4&-1/4&-1/4\\-1/4&3/4&-1/4\\-1/4&-1/4&3/4\end{pmatrix},\\ g_{ij}(b) &= \begin{pmatrix}4&0&2\\0&4&2\\2&2&3\end{pmatrix},\quad g^{ij}(b)=\begin{pmatrix}1/2&1/4&-1/2\\1/4&1/2&-1/2\\-1/2&-1/2&1\end{pmatrix}.\end{aligned} \tag{4.15}$$

[1] covariant basis vectors are column vectors of $\boldsymbol{U}$ and contravariant basis vectors are row vectors of $\boldsymbol{U}^{-1}$.

$$g_{ij}=\boldsymbol{u}_i\cdot\boldsymbol{u}_j=\left(\boldsymbol{U}^T\boldsymbol{U}\right)_{ij},\quad g^{ij}=\boldsymbol{u}^i\cdot\boldsymbol{u}^j=\left(\boldsymbol{U}^T\boldsymbol{U}\right)^{-1}_{ij} \tag{4.13}$$

- Laplacian: Then Laplacian is simply,

$$\nabla^2\psi = g^{ki}\frac{\partial^2\psi}{\partial q^k \partial q^i} \tag{4.16}$$

- Jacobian for Volume element is

$$J_c = 1, \quad J_f = 2, \quad J_b = 4 \tag{4.17}$$

- We may define momentum operator as

$$\hat{p}_i = -i\frac{\partial}{\partial q^i}. \tag{4.18}$$

- The distance between lattice points are

$$\begin{aligned} dx &= \frac{\partial x}{\partial q^1}dq^1 + \frac{\partial x}{\partial q^2}dq^2 + \frac{\partial x}{\partial q^3}dq^3, \\ ds^2 &= (dx)^2 + (dy)^2 + (dz)^2 = \sum_{ij} g_{ij}dq^i dq^j \end{aligned} \tag{4.19}$$

4.2 DFT of BCC/FCC lattice with Cubic boundary condition

4.2.1 DFT for BCC lattice with Cubic boundary condition

BCC lattice point (x, y, z) can be represented as $(2r, 2s, 2t)$ and $(2r+1, 2s+1, 2t+1)$ for $r, s, t \in Z$. Considering Cubic periodic boundary condition, $f_{BCC}(x, y, z) = f_{BCC}(x + lN, y + mN, z + nN)$ with $l, m, n \in Z$, for even integer N, we can define the DFT of BCC lattice as follows. First define $N \times N \times N$ samples as

$$f^{BCC}_{i,j,k} = \begin{cases} f(2r, 2s, 2t) & \text{if } i = 2r, j = 2s, k = 2t, \\ f(2r+1, 2s+1, 2t+1) & \text{if } i = 2r+1, j = 2s+1, k = 2t+1, \\ 0 & \text{otherwise} \end{cases} \tag{4.20}$$

DFT is

$$\begin{aligned} F^{BCC}_{u,v,w} &= \sum_{r=0}^{N/2-1}\sum_{s=0}^{N/2-1}\sum_{t=0}^{N/2-1} \Big(f^{BCC}_{2r,2s,2t} E^{\frac{1}{N}(u*2r+v*2s+w*2t)} \\ &\quad + f^{BCC}_{2r+1,2s+1,2t+1} E^{\frac{1}{N}(u*(2r+1)+v*(2s+1)+w*(2t+1))} \Big), \quad E = e^{-i2\pi} \\ &= DFT[f^{BCC}_{i,j,k}]_{u,v,w} \end{aligned} \tag{4.21}$$

where frequency $\nu_x = \frac{u}{N}$, $\nu_y = \frac{v}{N}$, $\nu_z = \frac{w}{N}$ and $DFT[f^{BCC}_{i,j,k}]_{u,v,w}$ is a standard DFT in $N \times N \times N$ dimension.

inverse DFT can be

$$f^{BCC}_{i,j,k} = \frac{1}{N^3}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\sum_{w=0}^{N-1} F^{BCC}_{u,v,w} E^{-\frac{1}{N}(u*i+v*j+w*k)} \tag{4.22}$$

However, this form have 4-fold redundancy because the actual non-zero samplings are $N^3/4$ but require N^3 FFT. This redundancy comes from "FCC pattern" of $F^{BCC}_{u,v,w}$,

$$F^{BCC}_{u+lN/2,v+mN/2,w+nN/2} = F^{BCC}_{u,v,w}, \quad \text{if } l+m+n = even. \tag{4.23}$$

To remove redundancy, let us consider following change of variable,

$$F^{BCC}_{u,v,w} = \sum_{r=0}^{N/2-1}\sum_{s=0}^{N/2-1}\sum_{t=0}^{N/2-1}(f^{BCC}_{2r,2s,2t}E^{\frac{1}{N}(u*(2r-2t)r+v*(2s-2t)+(u+v+w)*2t)}$$
$$+f^{BCC}_{2r+1,2s+1,2t+1}E^{\frac{1}{N}(u*(2r-2t)+v*(2s-2t)+(u+v+w)*(2t+1))}),$$
$$\Rightarrow \quad F^{BCC}_{p_1,p_2,p_3} = \sum_{x_3=0}^{N-1}\sum_{r=0}^{N/2-1}\sum_{s=0}^{N/2-1} f^{BCC}_{x_1,x_2,x_3}E^{\frac{p_1*x_1}{N/2}+\frac{p_2*x_2}{N/2}+\frac{p_3*x_3}{N}} \tag{4.24}$$

where $p_1 = u, p_2 = v, p_3 = u+v+w$ and $x_1 = r-t$, $x_2 = s-t$ and $x_3 = 2t$ or $2t+1$. By using the periodicity, we can rewrite $y_1 =$ mod $(x_1, N/2)$, $y_1 =$ mod $(x_2, N/2)$, $y_3 =$ mod (x_3, N),

$$F^{BCC}_{p_1,p_2,p_3} = \sum_{y_1=0}^{N/2-1}\sum_{y_2=0}^{N/2-1}\sum_{y_3=0}^{N-1} f^{BCC}_{y_1,y_2,y_3}E^{\frac{p_1*y_1}{N/2}+\frac{p_2*y_2}{N/2}+\frac{p_3*y_3}{N}} \tag{4.25}$$

This has the form of standard DFT of dimension $N^3/4$ without redundancy. It also explicitly shows periodicity and redundancy of $F^{BCC}_{u,v,w}$. Thus, one can set $p_1 =$ mod $(u, N/2)$, $p_2 =$ mod $(v, N/2)$ and $p_3 =$ mod $(u+v+w, N)$. Now, its inverse DFT can be written as

$$f^{BCC}_{y_1,y_2,y_3} = \frac{1}{N^3/4}\sum_{p_1=0}^{N/2-1}\sum_{p_2=0}^{N/2-1}\sum_{p_3=0}^{N-1} F^{BCC}_{p_1,p_2,p_3}E^{-(\frac{p_1*x_1}{N/2}+\frac{p_2*x_2}{N/2}+\frac{p_3*x_3}{N})}. \tag{4.26}$$

One can get original $f^{BCC}_{i,j,k}$ and $F^{BCC}_{u,v,w}$ up to periodicity.

4.2.2 DFT for FCC lattice with cubic boundary condition

Samplings at the FCC lattice points can be considered as $N \times N \times N$ values ,

$$f^{FCC}_{i,j,k} = \begin{cases} f(2r,2s,2t) & \text{if } i=2r, j=2s, k=2t, \\ f(2r,2s+1,2t+1) & \text{if } i=2r, j=2s+1, k=2t+1, \\ f(2r+1,2s,2t+1) & \text{if } i=2r+1, j=2s, k=2t+1, \\ f(2r+1,2s+1,2t) & \text{if } i=2r+1, j=2s+1, k=2t, \\ 0 & \text{otherwise,} \end{cases} \tag{4.27}$$

where $r, s, t \in Z$. DFT $F^{FCC}_{u,v,w}$ can be defined with frequency $\nu_x = \frac{u}{N}$, $\nu_y = \frac{v}{N}$, $\nu_z = \frac{w}{N}$, $u, v, w \in Z$, with even integer N,

$$\begin{aligned} F^{FCC}_{u,v,w} &= \sum_{r=0}^{N/2-1}\sum_{s=0}^{N/2-1}\sum_{t=0}^{N/2-1}\Big(f^{FCC}_{2r,2s,2t}E^{\frac{1}{N}(u*2r+v*2s+w*2t)} \\ &\quad +f^{FCC}_{2r,2s+1,2t+1}E^{\frac{1}{N}(u*(2r)+v*(2s+1)+w*(2t+1))} \\ &\quad +f^{FCC}_{2r+1,2s,2t+1}E^{\frac{1}{N}(u*(2r+1)+v*(2s)+w*(2t+1))} \\ &\quad +f^{FCC}_{2r+1,2s+1,2t}E^{\frac{1}{N}(u*(2r+1)+v*(2s+1)+w*(2t))}\Big), \quad E = e^{-i2\pi} \\ &= DFT[f^{FCC}_{i,j,k}]_{u,v,w} \end{aligned} \tag{4.28}$$

where $DFT[f^{FCC}_{i,j,k}]_{u,v,w}$ is a standard DFT in $N \times N \times N$ dimension.

And inverse DFT,

$$f^{FCC}_{i,j,k} = \frac{1}{N^3}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\sum_{w=0}^{N-1} F^{FCC}_{u,v,w}E^{-\left(\frac{ui}{N}+\frac{vj}{N}+\frac{wk}{N}\right)} \tag{4.29}$$

However, there are 2-fold degeneracy because

$$F^{FCC}_{u+lN/2,v+mN/2,w+nN/2} = F^{FCC}_{u,v,w}, \quad \text{if l,m,n are either all odd or all even.} \tag{4.30}$$

Let us rewrite as follows, showing above redundancy explicitly,

$$\begin{aligned} F^{FCC}_{u,v,w} &= \sum_{r=0}^{N/2-1}\sum_{s=0}^{N/2-1}\sum_{t=0}^{N/2-1} \Big(f^{FCC}_{2r,2s,2t} E^{\frac{u(r-s+t)}{N/2}+\frac{(v+u)(2s)}{N}+\frac{(w-u)(2t)}{N}} \\ &+ f^{FCC}_{2r,2s+1,2t+1} E^{\frac{u(r-s+t)}{N/2}+\frac{(v+u)(2s+1)}{N}+\frac{(w-u)(2t+1)}{N}} \\ &+ f^{FCC}_{2r+1,2s,2t+1} E^{\frac{u(r-s+t+1)}{N/2}+\frac{(v+u)(2s)}{N}+\frac{(w-u)(2t+1)}{N}} \\ &+ f^{FCC}_{2r+1,2s+1,2t} E^{\frac{u(r-s+t)}{N/2}+\frac{(v+u)(2s+1)}{N}+\frac{(w-u)(2t)}{N}} \Big) \end{aligned} \tag{4.31}$$

Let us denote $x_2 = 2s$ or $2s+1$ and $x_3 = 2t$ or $2t+1$ and $x_1 = r-s+t$ or $r-s+t+1$. Then define $y_1 =$ mod $(x_1, N/2)$, $y_2 =$ mod (x_2, N), $y_3 =$ mod (x_3, N), $p_1 =$ mod $(u, N/2)$, $p_2 =$ mod $(v+u, N)$, $p_3 =$ mod $(w-u, N)$. Then, we can have

$$F^{FCC}_{p1_1,p_2,p_3} = \sum_{y_1=0}^{N/2-1}\sum_{y_2=0}^{N-1}\sum_{y_3=0}^{N-1} f^{FCC}_{y_1,y_2,y_3} E^{\frac{p_1 y_1}{N/2}+\frac{p_2 y_2}{N}+\frac{p_3 y_3}{N}} \tag{4.32}$$

$$f^{FCC}_{y_1,y_2,y_3} = \frac{1}{N^3/2}\sum_{p_1=0}^{N/2-1}\sum_{p_2=0}^{N-1}\sum_{p_3=0}^{N-1} F^{FCC}_{p1_1,p_2,p_3} E^{-(\frac{p_1 y_1}{N/2}+\frac{p_2 y_2}{N}+\frac{p_3 y_3}{N})} \tag{4.33}$$

4.3 algorithm to generate $S_{\Omega,coef}$ for BCC lattice

There is no unique way to determine $S_{\Omega,coef}$. The explicit algorithm for $S_{\Omega,coef}$ is not given in [1].[2]

Let us give one explicit algorithm to generate $(c_1, c_2, c_3) \in S_{\Omega,coef}$ in one particular choice:

[2]Let us consider more detail on how to get S_Ω.

There is no congruent points in lattice points within the volume of Voronoi cells. This gives $|\Xi_0| = 4k^3 - 6k^2 + 4k - 1$ points. Let us list the 12 surfaces,

$$\begin{aligned} &f_1 : x+y = 2k, \quad f_2 : -x+y = 2k, \quad f_3 : -x-y = 2k, \quad f_4 : x-y = 2k, \\ &f_5 : x+z = 2k, \quad f_6 : -x+z = 2k, \quad f_7 : -x-z = 2k, \quad f_8 : x-z = 2k, \\ &f_9 : y+z = 2k, \quad f_{10} : -y+z = 2k, \quad f_{11} : -y-z = 2k, \quad f_{12} : y-z = 2k. \end{aligned} \tag{4.34}$$

For the points on the surface not on edges, Ξ_{f-e}, each points are congruent with one other points.

There are 24 edges. For points on edges but not on the vertice, Ξ_{e-v}, each points have other two congruent points. By using notation for edges, e_{i-j}, is a edge between face $f_i - f_j$,

$$\begin{aligned} &e_{1-5}, e_{1-8}, e_{1-9}, e_{1-12}, e_{2-6}, e_{2-7}, e_{2-9}, e_{2-12}, \\ &e_{3-6}, e_{3-7}, e_{3-10}, e_{3-11}, e_{4-5}, e_{4-8}, e_{4-10}, e_{4-11}, \\ &e_{5-9}, e_{5-10}, e_{6-10}, e_{6-9}, e_{7-11}, e_{7-12}, e_{8-11}, e_{8-12} \end{aligned} \tag{4.35}$$

There are 8 vertices where 3-edges meet. Ξ_{3v}. Each vertex have other 3 congruent vertices.

$$\begin{aligned} &v_{1,5,9} = (k,k,k) \simeq v_{2,7,12} = (-k,k,-k) \simeq v_{3,6,10} = (-k,-k,k) \simeq v_{4,8,11} = (k,-k,-k), \\ &v_{1,8,12} = (k,k,-k) \simeq v_{2,6,9} = (-k,k,k) \simeq v_{3,7,11} = (-k,-k,-k) \simeq v_{4,5,10} = (k,-k,k) \end{aligned}$$

There are 6 vertices where 4-edges meet, Ξ_{4v}. Every vertex are congruent.

$$\begin{aligned} & v_{1,5,4,8} = (2k,0,0) \simeq v_{2,6,3,7} = (-2k,0,0) \\ \simeq\ & v_{1,9,2,12} = (0,2k,0) \simeq v_{3,10,4,11} = (0,-2k,0) \\ \simeq\ & v_{5,10,6,9} = (0,0,2k) \simeq v_{7,11,8,12} = (0,0,-2k) \end{aligned} \tag{4.36}$$

The points on 12 surfaces but not on the edge gives $|\Xi_{f-e}| = 12(k-1)^2$. Each lattice points in Ξ_{f-e} is congruent to exactly 1 other point in Ξ_{f-e}.

```
! Algorithm for Somega_coef of BCC
for c1=-k
for c2=[-k:0]
for c3=[-c2:k-c2]
! (c2=0, c3=0) is not in Somega_coef
for c1=[-k+1:-1]
for c2=[-k:c1]
for c3=[-k-c1-c2:k-c1]
! all points are in Somega_coef
for c2=[c1+1:0]
for c3=[-k-c1-c2:k-c2]
! (c2=0,c3=-k-c1) are not in Somega_coef
for c2=[1:c1+k]
for c3=[-k-c1:k-c2]
! c3=-k-c1 points are not in Somega_coef
for c1=[0:k-1]
for c2=[c1-k:-1]
for c3=[-k-c2:k-c1]
! c2=c1-k points are not in Somega_coef
! c3=-k-c2 points are not in Somega_coef
for c2=[0:c1]
for c3=[-k-c2:k-c1-c2]
! c3=-k-c2  points are not in Somega_coef
! c3=k-c1-c2 points are not in Somega_coef
for c2=[c1+1:k]
for c3=[-k-c1:k-c1-c2]
! c3=(-k-c1)  points are not in Somega_coef
! c3=(k-c1-c2)   points are not in Somega_coef
! c2=k points are not in Somega_coef
for c1=k
for c2=[0:k]
for c3=[-k-c2:-c2]
! every points are not in Somega_coef
```

4.4 Algorithm for $S_{\Omega,coef}$ in FCC lattice

```
! Algorithm for Somega_coef of FCC
for c1=[-3*k/2:-k/2]
  for c2=[-2*k-c1:k/2]
    for c3=[-3*k/2-c1-c2:3*k/2]
      ! (c1=-k/2, c2=-2*k-c1) points are not in Somega_coef
      ! (c1=-k/2,c2=k/2,c3=-3*k/2-c1-c2) point is  not in Somega_coef
      ! (c1=-k/2,c2=k/2,c3=3*k/2) point is  not in Somega_coef
```

The points on 24 edges but not on the vertex gives $|\Xi_{e-v}| = 24(k-1)$ points. Each lattice points on Ξ_{e-v} is congruent to exactly 2 other lattice points.

Each points on 8 vertices with three-edges $\Xi_{v,3}$ is congruent with exactly three other lattice points on $\Xi_{v,3}$.

All points on 6 vertices with four-edges $\Xi_{v,4}$ are congruent to each other.

```
  for c2=[k/2+1:3*k/2]
    for c3=[-2*k-c1:2*k-c2]
      ! (c1=-k/2,c3=-2*k-c1) points are not in Somega_coef
      ! (c1=-k/2,c3=2*k-c2) points are not in Somega_coef

for c1=[-k/2+1:k/2]
  for c2=[-3*k/2:-c1]
    for c3=[-3*k/2-c1-c2:3*k/2]
      ! (c2=-3*k/2) points are not in Somega_coef
      ! (c2=-c1,c3=-3*k/2-c1-c2) point is not in Somega_coef
      ! (c2=-c1,c3=3*k/2) point is not in Somega_coef
      ! (c1=k/2,c3=-3*k/2-c1-c2) points are not in Somega_coef
      ! (c1=k/2,c3=3*k/2) points are not in Somega_coef
  for c2=[-c1+1:3*k/2]
    for c3=[-3*k/2:3*k/2-c1-c2]
      ! (c3=-3*k/2) points are not in Somega_coef
      ! (c3= 3*k/2-c1-c2 ) points are not in Somega_coef
      ! (c1=k/2, c2=3*k/2) points are not in Somega_coef
for c1=[k/2+1:3*k/2]
  for c2=[-3*k/2:-k/2]
    for c3=[-2*k-c2:2*k-c1]
      ! (c2=-3*k/2) points are not in Somega_coef
      ! (c3=-2*k-c2) points are not in Somega_coef
      ! (c3=2*k-c1) points are not in Somega_coef
      ! (c1=3*k/2) points are not in Somega_coef
  for c2=[-k/2+1:2*k-c1]
    for c3=[-3*k/2:3*k/2-c1-c2]
      ! (c2=2*k-c1) points are not in Somega_coef
      ! (c3=-3*k/2) points are not in Somega_coef
      ! (c3=3*k/2-c1-c2) points are not in Somega_coef
      ! (c1=3*k/2) points are not in Somega_coef
```

Bibliography

[1] Xiqiang Zheng and Feng Gu, J. Math. Imaging Vis (2014) 49:530-550.

[2] Balazs Csebfalvi and Balazs Domonkos, VMV 2008, proceeding