1

Angular Convolution, A Better Algorithm

In previous sections, the spatial convolution in the excess functional gradient is treated by **FFT!** thanks to the transitional invariance that leads to $\mathbf{r}_{12} = \mathbf{r}_1 - \mathbf{r}_2$. However, as the angular grid is not homogeneous, the relative coordinates of two angles cannot be simply represented $\mathbf{\Omega}_{12} = \mathbf{\Omega}_1 - \mathbf{\Omega}_2$, therefore for the angles we cannot take advantage of the convolution property shown in eq. (**??**-**??**). On the other hand, these two-particle quantities have rotational invariance. As proposed by Blum [**Blum_I**, **Blum_II**], a rotational invariant expansion technique is used to reduce the molecular Ornstein-Zernike (**MOZ!**) equation into smaller irreducible matrix equations (§**??**). Owing to the mathematical equivalence between **IET!** and **DFT!** approach (§**??**), where eq. (**??**) about the Fourier transform of the excess functional gradient can be regarded as the **MOZ!** equation, the formalism of Blum can also be applied to **MDFT!**.

1.1 Angular convolution using Blum's reduction

To build a relation between the irreducible form of the **MOZ!** equation for homogeneous fluid deduced by Blum (detailed in §**??**, with the symmetry $\hat{c}'^{nm}_{\nu\mu,\chi}(k) = \hat{c}'^{mn*}_{\mu\nu,\underline{\chi}}(k)$)

$$\hat{\gamma}'^{lm}_{\lambda\mu,\underline{\chi}}(k) = \sum_{n=0}^{n_{\max}} \sum_{\nu=-n}^{n} (-)^{\chi+\nu} \Delta\hat{\rho}'^{ln}_{\lambda\underline{\nu},\underline{\chi}}(k) \hat{c}'^{mn*}_{\mu\nu,\chi}(k) \tag{1.1}$$

and the **MDFT!** formalism, a generalized spherical harmonic transform (**GSHT!**) treatment is proposed by developing the functional gradient $\hat{\gamma}$ and the density $\hat{\rho}$ in eq. (**??**) on Wigner generalized spherical harmonics (**GSH!**):

$$\hat{\gamma}(\mathbf{k}, \mathbf{\Omega}_1) = \sum_{m\mu'\mu} f_m \hat{\gamma}^m_{\mu'\mu}(\mathbf{k}) R^m_{\mu'\mu}(\mathbf{\Omega}_1) \tag{1.2}$$

$$\Delta\hat{\rho}(\mathbf{k}, \mathbf{\Omega}_2) = \sum_{n\nu'\nu} f_n \Delta\hat{\rho}^n_{\nu',\nu}(\mathbf{k}) R^n_{\nu',\nu}(\mathbf{\Omega}_2) \tag{1.3}$$

where $0 \le m, n \le n_{\max}$, $|\mu'|, |\mu| \le m$ and $|\nu'|, |\nu| \le n$. $f_m = (2m+1)^{\frac{1}{2}} = \left\| R^m_{\mu'\mu} \right\|^{-1}$ is the normalization factor.

The **DCF!** can also be expanded on rotational invariants:

$$\hat{c}(k, \mathbf{\Omega}_1, \mathbf{\Omega}_2) = \sum_{mnl\mu\nu} f_m f_n \hat{c}^{mnl}_{\mu\nu}(k) \sum_{\mu'\nu'\lambda'} \begin{pmatrix} m & n & l \\ \mu' & \nu' & \lambda' \end{pmatrix} R^m_{\mu'\mu}(\mathbf{\Omega}_1) R^n_{\nu'\nu}(\mathbf{\Omega}_2) R^l_{\lambda' 0}(\hat{\mathbf{k}}) \tag{1.4}$$

As **GSH!** possess orthogonality eq. (**??**) and symmetry eq. (**??**), eq. (**??**) can be rewritten by (1.2, 1.3, 1.4), which gives:

$$\hat{\gamma}^m_{\mu'\mu}(\mathbf{k}) = \sum_{nl\nu} \hat{c}^{mnl}_{\mu\nu}(k) \sum_{\nu'\lambda'} (-)^{\nu'+\nu} \Delta\hat{\rho}^n_{\underline{\nu'\nu}}(\mathbf{k}) \begin{pmatrix} m & n & l \\ \mu' & \nu' & \lambda' \end{pmatrix} R^l_{\lambda' 0}(\hat{\mathbf{k}}) \tag{1.5}$$

thus the **OZ!** equation is expanded on **GSH!**s and rotational invariants.

*Here the projections $F^{mn}_{\mu\nu,\chi}$ are defined as in [**Fries_Patey_1985**]. Eq. (1.1) is mathematically identical with those in [**Blum_I**, **Blum_II**] but using $R^m_{\mu'\mu} = D^{m*}_{\mu\mu'}$. The difference between the conventions of **GSH!** are listed in §**??**.*

Note that eq. (1.5) is reducible. Blum's χ-transform [**Blum_II**] defines:

$$\hat{c'}^{mn}_{\mu\nu,\chi}(k) = \sum_{l=|m-n|}^{m+n} \begin{pmatrix} m & n & l \\ \chi & -\chi & 0 \end{pmatrix} \hat{c}^{mnl}_{\mu\nu}(k) \tag{1.6}$$

*In fact, in **MDFT!**, $\hat{c'}^{mn}_{\mu\nu,\chi}(k)$ is the conjugate of the invariant in eq. (1.1).*

$$\hat{c}^{mnl}_{\mu\nu}(k) = (2l+1) \sum_{\chi=-\min(m,n)}^{\min(m,n)} \begin{pmatrix} m & n & l \\ \chi & -\chi & 0 \end{pmatrix} \hat{c'}^{mn}_{\mu\nu,\chi}(k) \tag{1.7}$$

Invariants of form $F^{mn}_{\mu\nu,\chi}(k)$ have a very simple relation with their combined function $F(k,\boldsymbol{\omega_1},\boldsymbol{\omega_2})$ in intermolecular coordinate system (see appendix **??**, eq. (**??**, **??**)), that is how the **OZ!** equation can be reduced. In **MDFT!**, we can also take advantage of this quantity by defining the projections of $\hat{\gamma}$ and $\hat{\rho}$ in the local frame ($\boldsymbol{\omega}_i = \hat{\mathbf{k}}\boldsymbol{\Omega}_i$):

$$\hat{\gamma'}(\mathbf{k},\boldsymbol{\omega}_1) = \sum_{m\chi\mu} f_m \hat{\gamma'}^{m}_{\chi\mu}(\mathbf{k}) R^{m}_{\chi\mu}(\boldsymbol{\omega}_1) \tag{1.8}$$

$$\Delta\hat{\rho'}(\mathbf{k},\boldsymbol{\omega}_2) = \sum_{n\chi\nu} f_n \Delta\hat{\rho'}^{n}_{\chi\nu}(\mathbf{k}) R^{n}_{\chi\nu}(\boldsymbol{\omega}_2) \tag{1.9}$$

and with the rotation formula of **GSH!** (eq. (**??**)), we have

$$\hat{\gamma'}^{m}_{\chi\mu}(\mathbf{k}) = \sum_{\mu'} \hat{\gamma}^{m}_{\mu'\mu}(\mathbf{k}) R^{m}_{\mu'\chi}(\hat{\mathbf{k}}) \tag{1.10}$$

$$\Delta\hat{\rho}^{n}_{\underline{\nu'}\underline{\nu}}(\mathbf{k}) = \sum_{\chi} \Delta\hat{\rho'}^{n}_{\chi\underline{\nu}}(\mathbf{k}) R^{n*}_{\underline{\nu'}\chi}(\hat{\mathbf{k}}) = \sum_{\chi} \Delta\hat{\rho'}^{n}_{\chi\underline{\nu}}(\mathbf{k}) \,(-)^{\,\chi+\nu'} R^{n}_{\nu'\underline{\chi}}(\hat{\mathbf{k}}) \tag{1.11}$$

Using eq. (1.10), (1.5), eq. (1.11), eq. (1.7) and **GSH!** products relation eq. (**??**) and 3j-symbol orthogonality eq. (**??**), we deduce that:

$$\hat{\gamma'}^{m}_{\chi\mu}(\mathbf{k}) = \sum_{n\nu} (-)^{\,\chi+\nu} \hat{c'}^{mn}_{\mu\nu,\chi}(k) \Delta\hat{\rho'}^{n}_{\chi\underline{\nu}}(\mathbf{k}) \tag{1.12}$$

Eq. (1.12) is essential to the new algorithm. It makes that, for the terms with the same index χ, the **OZ!** equation is a simple product of matrix:

$$\tilde{\gamma'}_{\chi} = \left[(-)^{\,\chi+\nu} \tilde{c'}_{\chi}\right] \tilde{\rho'}_{\chi} \tag{1.13}$$

The index χ shares the same role with $\mathbf{k}$ in the treatment of spatial convolution, where the recombination of projections on exponential orthogonal bases gives for each $\mathbf{k}$, a simple product form of the **OZ!** equation.

If we take the conjugate of $\hat{c'}^{mn*}_{\mu\nu,\chi}(k)$, eq. (1.12) is mathematically identical to eq. (1.1), as:

$$\hat{\gamma'}^{m}_{\chi\mu}(\mathbf{k}) = \sum_{l\lambda} \hat{\gamma'}^{lm}_{\lambda\mu,\underline{\chi}}(k) R^{l}_{\underline{\chi}\lambda}(\boldsymbol{\omega}) \tag{1.14}$$

$$\hat{\rho'}^{n}_{\chi\underline{\nu}}(\mathbf{k}) = \sum_{l\lambda} \Delta\hat{\rho'}^{ln}_{\lambda\underline{\nu},\chi}(k) R^{l*}_{\underline{\chi}\lambda}(\boldsymbol{\omega}) \tag{1.15}$$

according to the rotational invariant transform in eq. (**??**). The conjugate comes from that we take particle 2-1 formalism for **MDFT!**, and **IET!** takes the 1-2 formalism. The same issue arrives in the comparison of $g^{mnl}_{\mu\nu}$ of the two approaches, where in **MDFT!** the $g^{mnl}_{\mu\nu}$ corresponds to the $g^{nml}_{\nu\mu}$ in **IET!**.

With the approach described above, the integral of the angular part in eq. (**??**) can be reduced to a sum of a few terms. Table 1.1 shows some parameters linking to computing cost of different algorithms. It shows that the expansion on **GSH!**s (eq. (1.5)) does not give any reduction of **FE!** compared to its 6D function form (eq. (**??**))*; but after the Blum's χ-transform, the **OZ!** equation is largely reduced. The fact is that as the treatment of spatial convolution takes advantage of the transitional invariance r_{12}, the χ-transform makes use of the rotational invariance.

** Only if we do not need to calculate $\hat{c}(\mathbf{k},\boldsymbol{\Omega}_1,\boldsymbol{\Omega}_2)$.*

$m_{\max}$	0	1	2	3	4	5
N_Θ	1	2	3	4	5	6
N_{ang} (Gauss-Legendre)	1 (1)	18 (6)	75 (45)	196 (84)	405 (225)	726 (330)
N_{ang} (Lebedev×ψ)	1 (1)	18 (6)	70 (42)	182 (78)	342 (190)	550 (250)
N_{proj}	1 (1)	10 (4)	35 (19)	84 (40)	165 (85)	286 (140)
FE for eq. (**??**)	1 (1)	324 (36)	5625 (2025)	38416 (7056)	164025 (50625)	527076 (108900)
FE for eq. (1.5)	1 (1)	262 (34)	4787 (1459)	36588 (8116)	175989 (47221)	633490 (150566)
FE for eq. (1.12)	1 (1)	34 (6)	259 (75)	1092 (252)	3333 (877)	8294 (2002)

Table 1.1: Number of **FE!** needed by **OZ!** equation of different form for arbitrary solvent (outside the parentheses) and solvent possessing C_{2v} symmetry (inside the parentheses)

1.2 FAST GENERALIZED SPHERICAL HARMONIC TRANSFORM

The algorithm above for angular convolution takes advantage of the orthogonality and symmetries of **GSH!**s. To use this algorithm as analogous to the treatment of the convolution with **FFT!** for spatial grids, the transform described in eq. (1.2) and (1.3), here defined as the generalized spherical harmonic transform (**GSHT!**), *a priori* should be fast. This is possible owing to the exponential components in the definition of **GSH!**, that will be discussed later as the fast generalized spherical harmonic transform (**FGSHT!**).

GSHT! provides a forward-backward transform between a general angular function $F(\mathbf{\Omega}) \equiv F(\cos\Theta, \Phi, \Psi)$ and its projections $F^m_{\mu'\mu}$ ($|\mu'|, |\mu| \leq m$):

$$F^m_{\mu'\mu} = \frac{f_m}{8\pi^2} \int \mathrm{d}\mathbf{\Omega} F(\mathbf{\Omega}) R^{m*}_{\mu'\mu}(\mathbf{\Omega}) \quad \text{(forward)} \tag{1.16}$$

$$F(\mathbf{\Omega}) = \sum_{m,\mu',\mu} f_m F^m_{\mu'\mu} R^m_{\mu'\mu}(\mathbf{\Omega}) \quad \text{(backward)} \tag{1.17}$$

where $\left\{R^m_{\mu'\mu}(\mathbf{\Omega})\right\}$ are the Wigner generalized spherical harmonics (Appendix **??**), which form a complete orthogonal set, being defined as:

$$R^m_{\mu'\mu}(\mathbf{\Omega}) = r^m_{\mu'\mu}(\Theta) e^{-i(\mu'\Phi + \mu\Psi)} \tag{1.18}$$

1.2.1 *Equivalence of order in angular quadratures and projections*

Suppose that $F(\mathbf{\Omega})$ is a polynomial of both $\cos\Theta$, $\cos\Phi$ and $\cos\Psi$ of order n, ($n+1$ polynomial terms). To completely expand this function as shown in equation (1.17), at least $m_{\max} = n$ is needed, where $m_{\max}$ is the highest order of projections $F^m_{\mu'\mu}$ in the expansion. Note that $m_{\max} = n$ is not always sufficient to completely expand $F(\mathbf{\Omega})$, a discussion to this issue will be given in §**??**.

To evaluate exactly the integration in equation (1.16), at least $n+1$ for $\cos\Theta$ (Gauss-Legendre grid), $2n+1$ for Φ (equal-spaced grid), $2n+1$ for Ψ (equal-spaced grid) points of angular grid are needed (c.f. appendix **??**). In the case of water which possesses a C_2 symmetry $F(\Psi + \pi) = F(\Psi)$, only projections of even μ are nonzero:

$$\begin{aligned} F_\mu &= \int \mathrm{d}\Psi F(\Psi) e^{i\mu\Psi} = \int \mathrm{d}(\Psi+\pi) F(\Psi+\pi) e^{i\mu(\Psi+\pi)} \\ &= e^{i\mu\pi} \int \mathrm{d}\Psi F(\Psi) e^{i\mu\Psi} = e^{i\mu\pi} F_\mu \end{aligned} \tag{1.19}$$

$$F_\mu = \begin{cases} 0 & \mu = 2n+1, n \in \mathbb{Z} \\ F_\mu & \mu = 2n, n \in \mathbb{Z} \end{cases} \tag{1.20}$$

Therefore the function

$$F(\Psi) = \sum_\mu F_\mu e^{-i\mu\Psi} \tag{1.21}$$

can be rewritten as:

$$F(\Psi_2/2 \equiv \Psi) = \sum_{\mu_2 \equiv \mu/2} F_{2\mu_2} e^{-i\mu_2\Psi_2} \tag{1.22}$$

As $|\mu_2| \leq n/2$, $F(\Psi_2/2 \equiv \Psi)$ is a polynomial of $\cos\Psi_2$ of order $\mathrm{floor}(n/2) \equiv \lfloor n/2 \rfloor$, in the forward transform

$$F_{2\mu_2 \equiv \mu} = \int \mathrm{d}\Psi F(\Psi) e^{i\mu\Psi} = \frac{1}{2}\int \mathrm{d}\Psi_2 F(\Psi_2/2 \equiv \Psi) e^{i\mu_2\Psi_2} \tag{1.23}$$

the total degree $\cos\Psi_2$ polynomial in the integrand is $2\lfloor n/2 \rfloor$, then $2\lfloor n/2 \rfloor + 1$ points of Ψ_2 (or Ψ) are needed.

For further implementation, we take these conclusions, but distinguish the order of quadrature $m_{\max}$ (linked to the angular grid) and the order of projection $n_{\max}$ (linked to the **GSH!** transform) for numerical reason.

1.2.2 *Integration of Φ, Ψ using FFT*

Here we write eq. (1.16, 1.17) in an explicit way:

$$F^m_{\mu'\mu} = \frac{f_m}{8\pi^2} \sum_{i=0}^{m_{\max}} w_i \sum_{j=0}^{2m_{\max}} \sum_{k=0}^{2\lfloor m_{\max}/s \rfloor} F(\Theta_i, \Phi_j, \Psi_k) R^{m*}_{\mu'\mu}(\Theta_i, \Phi_j, \Psi_k) \tag{1.24}$$

$$F(\Theta_i, \Phi_j, \Psi_k) = \sum_{m=0}^{n_{\max}} f_m \sum_{\mu'=-m}^{m} \sum_{\substack{\mu=-m \\ \mathrm{mod}\ (\mu,s)=0}}^{m} F^m_{\mu'\mu} R^m_{\mu'\mu}(\Theta_i, \Phi_j, \Psi_k) \tag{1.25}$$

where w_i is the weight of Gauss-Legendre quadrature ($m_{\max}+1$ points of Θ_i), normalized to the total angular integration; and s is the molecule rotation symmetry order (**MRSO!**), $s = 1$ or 2 according to the symmetry C_s of solvent.

To integrate eq. (1.24) in a direct way, $(m_{\max}+1)(2m_{\max}+1)(2\lfloor m_{\max}/s \rfloor + 1) = N_\Theta N_{\Phi\Psi} = N$ **FE!** are needed for each $F^m_{\mu'\mu}$, an overall $O(N^2_{FE})$ process is needed and *vice versa*. Therefore, a faster algorithm proposed by Numerical Recipes [**Numerical_Recipes_3ed**] suggests to reduce this cost to $O(N^2_\Theta N_{\Phi\Psi} \ln N_{\Phi\Psi} \simeq N^{4/3})$ by **FFT!**.

Following this idea, eq. (1.24) can be rewritten as:

$$F^m_{\mu'\mu} = \frac{f_m}{8\pi^2} \sum_{i=0}^{m_{\max}} w_i r^m_{\mu'\mu}(\Theta_i) F_{\mu'\mu}(\Theta_i) \tag{1.26}$$

where $F_{\mu'\mu}(\Theta_i)$ is the Φ, Ψ integration evaluated using trapezoid (or Gauss-Chebyshef) quadrature:

$$\begin{aligned} F_{\mu'\mu}(\Theta_i) &= \sum_{j=0}^{2m_{\max}} \sum_{k=0}^{2\lfloor m_{\max}/s \rfloor} F(\Theta_i, \Phi_j, \Psi_k) e^{i(\mu'\Phi_j + \mu\Psi_k)} \\ &= \sum_{j=0}^{2m_{\max}} \sum_{k=0}^{2\lfloor m_{\max}/s \rfloor} F(\Theta_i, \Phi_j, \Psi_k) e^{2\pi i\mu' j/(2m_{\max}+1)} e^{2\pi i\mu k/(2\lfloor m_{\max}/s \rfloor + 1)} \end{aligned} \tag{1.27}$$

that shares the same formula with an **FFT!**-2D process of $(2m_{\max}+1)\,(2\lfloor m_{\max}/s\rfloor+1)$ elements.

Similarly, the backward process (1.17) can be rewritten as:

$$\begin{aligned} F(\Theta_i,\Phi_j,\Psi_k) &= \sum_{m=0}^{n_{\max}} f_m \sum_{\mu'=-m}^{m} \sum_{\substack{\mu=-m \\ \text{mod }(\mu,s)=0}}^{m} F^m_{\mu'\mu} R^m_{\mu'\mu}(\Theta_i,\Phi_j,\Psi_k) \\ &= \sum_{\mu'=-n_{\max}}^{n_{\max}} \sum_{\substack{\mu=-n_{\max} \\ \text{mod }(\mu,s)=0}}^{n_{\max}} \sum_{m=\max(|\mu'|,|\mu|)}^{n_{\max}} f_m F^m_{\mu'\mu} R^m_{\mu'\mu}(\Theta_i,\Phi_j,\Psi_k) \\ &= \sum_{\mu'=-n_{\max}}^{n_{\max}} \sum_{\substack{\mu=-n_{\max} \\ \text{mod }(\mu,s)=0}}^{n_{\max}} F_{\mu'\mu}(\Theta_i) e^{2\pi i\mu' j/(2m_{\max}+1)} e^{2\pi i\mu k/(2\lfloor m_{\max}/s\rfloor+1)} \end{aligned} \tag{1.28}$$

with

$$F_{\mu'\mu}(\Theta_i) = \sum_{m=\max(|\mu'|,|\mu|)}^{n_{\max}} f_m F^m_{\mu'\mu} r^m_{\mu'\mu}(\Theta_i) \tag{1.29}$$

*Note that the **GSHT!** is able to treat the case $n_{\max} > m_{\max}$.*

When $n_{\max} \le m_{\max}$, the double sum in eq. (1.28) is included in the **FFT!**-2D process of $(2m_{\max}+1)\,(2\lfloor m_{\max}/s\rfloor+1)$ elements. However, if $n_{\max} > m_{\max}$, the **FFT!**-2D process only gives a partial sum of $|\mu'|, |\mu| \le m_{\max}$, the other terms in eq. (1.28) can only be calculated by a **GSHT!** process, as $F_{\mu'\mu}(\Theta_i)$ is not periodic for μ' and μ. There can be further approximations to treat this problem, but for practical usage, we only consider the case of $n_{\max} \le m_{\max}$.

The FFTW3 library [**FFTW3**] is used for implementation, which performs discrete Fourier Transform (**DFT!**) as defined below:

$$Y_k = \sum_{j=0}^{n-1} X_j e^{-2\pi ijk/n} \quad \text{(forward)} \tag{1.30}$$

$$X_j = \sum_{k=0}^{n-1} Y_k e^{2\pi ijk/n} \quad \text{(backward)} \tag{1.31}$$

Note that after a forward-backward Fourier transform, the original function is multiplied by a normalization factor N_k, which is the total number of nodes k.

For input function Y_k ($k = 0, \ldots, n-1$) in real number, FFTW3 only outputs elements $k = 0, \ldots, \lfloor n/2\rfloor$ ($\lfloor n/2\rfloor + 1$ complex numbers of X_j are stocked), with the "Hermitian" symmetry

$$Y_k = Y^*_{n-k} \tag{1.32}$$

used to regenerate elements of $k > \lfloor n/2\rfloor$. The resulting X_j issue from the corresponding backward transform is purely real. As the angular function $F(\mathbf{\Omega})$ is real, and the **GSH!**s possess symmetry of eq. (**??**):

$$R^m_{-\mu',-\mu}(\mathbf{\Omega}) = (-1)^{\mu'+\mu} R^{m*}_{\mu'\mu}(\mathbf{\Omega}) \tag{1.33}$$

the symmetry relation between the projections are

$$F^m_{-\mu',-\mu} = (-1)^{\mu'+\mu} F^{m*}_{\mu'\mu} \tag{1.34}$$

Therefore only the projections of $\mu \ge 0$ need to be stocked, which can be calculated with only these FFTW3 output elements reduced by the Hermitian symmetry. The full process of FFTW3-2D real to real transform is illustrated in figure 1.1.

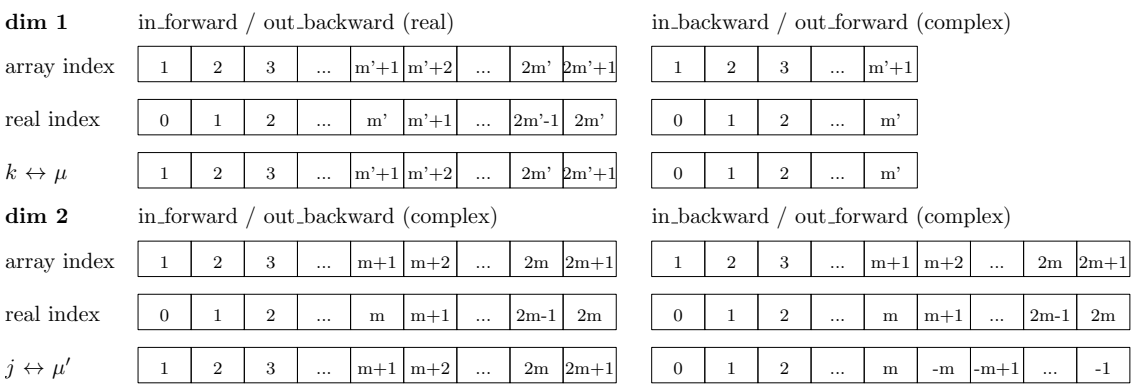

Figure 1.1: Indices arrangement in a complete forward-backward **FFT!**-2D process of m'×m elements. The **DFT!** of dim 1 (k to μ) and dim 2 (j to μ') are done sequentially and *vice versa*. Array index is the one used by Fortran array, real index is the one shown in eq. (1.30) and (1.31), k and j indices shown in the left as well as μ and μ' in the right are those in eq. (1.27) and (1.28). Here $\mathrm{m} = m_{\max}$ and $\mathrm{m}' = \lfloor m_{\max}/s \rfloor$.

As the output array of FFTW3 is periodic,

$$e^{2\pi i\mu k/n} = e^{2\pi i(\mu-n)k/n} e^{2\pi ik} = e^{2\pi i(\mu-n)k/n} \tag{1.35}$$

the indices $\mu = m_{\max}+1, \ldots, 2m_{\max}$ actually correspond to $\mu = -m_{\max}, \ldots, -1$. Note that eq. (1.27) and (1.28) do not possess the periodicity of eq. (1.35), only in the domain of definition of μ' and μ some intermediary functions share the same formula with **FFT!**.

Moreover, from eq. (1.27), (1.29) and (1.34), we can verify that

$$F_{\mu'\mu}(\Theta) = F^*_{-\mu',-\mu}(\Theta) \tag{1.36}$$

The latter is used in the code since, according to the definition in eq. (1.30) and (1.31), $F_{-\mu',-\mu}(\Theta)$ is calculated instead of $F_{\mu'\mu}(\Theta)$.

1.3 OPERATIONAL ALGORITHM

As described above, the whole process of γ and $\mathcal{F}_{\mathrm{exc}}$ functional evaluation proposed by this algorithm can be concluded as 8 operations:

1. Firstly, the Fourier transform of the density is computed:

$$\Delta\hat{\rho}(\mathbf{k}, \mathbf{\Omega}) = \int \mathrm{d}\mathbf{r} \Delta\rho(\mathbf{r}, \mathbf{\Omega}) e^{-i\mathbf{k}\cdot\mathbf{r}} \tag{1.37}$$

2. Then $\Delta\hat{\rho}(\mathbf{k}, \mathbf{\Omega})$ is expanded on **GSH!**s:

$$\Delta\hat{\rho}^m_{\mu'\mu}(\mathbf{k}) = \frac{f_m}{8\pi^2} \int \mathrm{d}\mathbf{\Omega} \Delta\hat{\rho}(\mathbf{k}, \mathbf{\Omega}) R^{m*}_{\mu'\mu}(\mathbf{\Omega}) \tag{1.38}$$

   Note that these two steps, the same with their backward transform, are commutable, which will be discussed after.

3. Afterwards the projections in k-frame are then rotated into the local coordinate system along the unit vector $\hat{\mathbf{k}}$:

$$\Delta\hat{\rho'}^m_{\chi\mu}(\mathbf{k}) = \sum_{\mu'} \Delta\hat{\rho}^m_{\mu'\mu}(\mathbf{k}) R^m_{\mu'\chi}(\hat{\mathbf{k}}) \tag{1.39}$$

where the rotation matrix elements $R^m_{\mu'\chi}(\hat{\mathbf{k}})$ should be calculated directly because of the huge memory required by its storage. The algorithm by recurrence used to evaluate $R^m_{\mu'\chi}(\hat{\mathbf{k}})$ in this thesis is detailed in appendix **??**.

4. Next, computing the OZ equation with Blum's reduction:

$$\hat{\gamma'}^m_{\chi\mu}(\mathbf{k}) = \sum_{n,\nu}(-1)^{\chi+\nu}\hat{c}^{mn}_{\mu\nu,\chi}(\mathbf{k})\Delta\hat{\rho'}^n_{\chi\underline{\nu}}(\mathbf{k}) \tag{1.40}$$

5. The γ projections are then transformed back to global coordinates system:

$$\hat{\gamma}^m_{\mu'\mu}(\mathbf{k}) = \sum_{\chi}\hat{\gamma'}^m_{\chi\mu}(\mathbf{k})R^{m*}_{\mu'\chi}(\hat{\mathbf{k}}) \tag{1.41}$$

6. From here the function in angular frame can thus be rebuilt:

$$\hat{\gamma}(\mathbf{k},\mathbf{\Omega}) = \sum_{m,\mu',\mu} f_m\hat{\gamma}^m_{\mu'\mu}(\mathbf{k})R^m_{\mu'\mu}(\mathbf{\Omega}) \tag{1.42}$$

7. Then the inverse Fourier transform of these projections is:

$$\gamma(\mathbf{r},\mathbf{\Omega}) = \int \mathrm{d}\mathbf{k}\hat{\gamma}(\mathbf{k},\mathbf{\Omega})e^{i\mathbf{r}\cdot\mathbf{k}} \tag{1.43}$$

8. Finally, the functional $\mathcal{F}_{\mathrm{exc}}$ is computed by:

$$\mathcal{F}_{\mathrm{exc}} = \frac{1}{2}\int \mathrm{d}\mathbf{r}\mathrm{d}\mathbf{\Omega}\Delta\rho(\mathbf{r},\mathbf{\Omega})\gamma(\mathbf{r},\mathbf{\Omega}) \tag{1.44}$$

1.3.1 *Commutativity between operations*

As mentioned in the operational algorithm, three types of operations are being done before and after the **OZ!** equation. They are:

1. Fast Fourier transform for 3-dimensional spatial grid (**FFT!**3D): implemented by package FFTW3 [**FFTW3**], mathematically leading to no accuracy lost;
2. Fast generalized spherical harmonics transform (**FGSHT!**): has real or complex input, is exact if $F(\mathbf{\Omega})$ can be given as an expansion of **GSH!**s of order at most $m_{\max}$;
3. Rotation between laboratory coordinate system and local system linked to vector $\mathbf{k}$ (RotS): can be done for both function and projections. It introduces a minus error in accuracy at origin and border of the box, which will be discussed in §**??**.

Their commutativity is shown in figure 1.2.

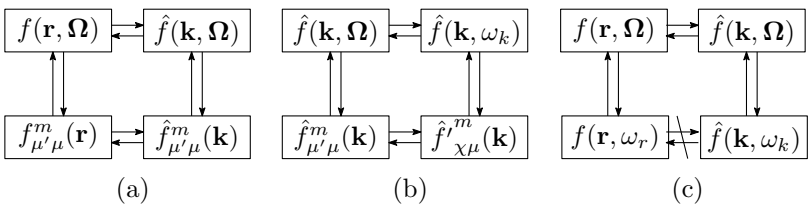

Figure 1.2: Commutativity of operations. (a) FFT3D and FGSHT; (b) RotS and FGSHT; (c) FFT3D and RotS.

As shown in figure 1.2, the **FFT!**3D does not depend on the angular part of the function, and the **FGSHT!** does not depend on the spatial part of the function. The two operations are commutative.

It can be also proven that the passage from the function $\hat{f}$ in laboratory frame $\hat{f}(\mathbf{k}, \mathbf{\Omega})$ to the projections in local frame ${f'_{\chi\mu}}^m(\mathbf{k})$ can be achieved either by a rotation to the function $\hat{f}'(\mathbf{k}, \boldsymbol{\omega}_k)$ in intermolecular frame following by an **GSH!** expansion as eq. (1.8), or an **GSH!** expansion that gives the projections $f^m_{\mu'\mu}(\mathbf{k})$ following by a rotation as eq. (1.10).

However, the rotation from $f(\mathbf{r}, \mathbf{\Omega})$ to $f(\mathbf{r}, \boldsymbol{\omega})$ depends on the vector $\mathbf{r}$, of which the information is totally lost after **FFT!**3D. The rotation from $f(\mathbf{k}, \mathbf{\Omega})$ to $f(\mathbf{k}, \boldsymbol{\omega})$ can only depend on the vector $\mathbf{k}$; they are not the same rotation, therefore non-commutative.

1.3.2 *Reduction by symmetry*

A further reduction of computing cost can be made by performing approximately only half of the operations, thanks to the symmetric relations between the projections.

In eq. (1.38), $\Delta\rho(\mathbf{r}, \mathbf{\Omega})$ is real. Thanks to the property of **GSH!** (eq. (**??**)):

$$R^m_{\mu'\mu}(\mathbf{\Omega}) = (-)^{\mu'+\mu} R^{m*}_{-\mu'-\mu}(\mathbf{\Omega}) \tag{1.45}$$

we find

$$\Delta\hat{\rho}^m_{\mu'\mu}(\mathbf{r}) = (-)^{\mu'+\mu} \Delta\hat{\rho}^{m*}_{-\mu',-\mu}(\mathbf{r}) \tag{1.46}$$

Therefore only the projections of $\mu' \geq 0$ or $\mu \geq 0$ are needed to generate all information.

When $\Delta\hat{\rho}^m_{\mu'\mu}(\mathbf{r})$ is transformed into k-space, replacing

$$\Delta\hat{\rho}^m_{\mu'\mu}(\mathbf{k}) = \int \mathrm{d}\mathbf{r} \Delta\rho^m_{\mu'\mu}(\mathbf{r}) e^{-i\mathbf{r}\cdot\mathbf{k}} \tag{1.47}$$

with eq. (1.46) gives

$$\Delta\hat{\rho}^m_{\mu'\mu}(\mathbf{k}) = (-)^{\mu'+\mu} \Delta\hat{\rho}^{m*}_{-\mu'-\mu}(-\mathbf{k}) \tag{1.48}$$

Only the projections of $\mu' \geq 0$, $\mu \geq 0$, or a half of $\mathbf{k}$ where one of the dimensions $k_i \geq 0$ are independent.

The rotation to local frame is governed by the relation deduced from the symmetries in appendix **??**, such that

$$r^m_{\mu\mu'}(-\theta) = (-)^{\mu'-\mu} r^m_{\mu'\mu}(-\theta) \tag{1.49}$$

$$r^m_{\underline{\mu'}\mu}(\theta) = (-)^{\mu'-\mu} r^m_{\mu'\underline{\mu}}(\theta) \tag{1.50}$$

$$r^m_{\mu'\mu}(\theta) = (-)^{\mu'-\mu} r^m_{\mu'\mu}(-\theta) \tag{1.51}$$

$$r^m_{\mu'\mu}(\theta) = (-)^{m+\mu} r^m_{\underline{\mu'}\mu}(\pi-\theta) = (-)^{m+\mu'} r^m_{\mu'\underline{\mu}}(\pi-\theta) \tag{1.52}$$

$$R^m_{\mu'\mu}(\phi\theta\psi) = (-)^{m+\mu} R^m_{\underline{\mu'}\mu}(-\phi, \pi-\theta, \psi) = (-)^{m+\mu'} R^m_{\mu'\underline{\mu}}(\phi, \pi-\theta, -\psi) \tag{1.53}$$

$$R^m_{\mu'\mu}(\phi\theta\psi) = (-)^{m+\mu} R^m_{\underline{\mu'}\mu}(-\phi, \pi-\theta, \psi) = (-)^{m+\mu'} R^{m*}_{\mu'\underline{\mu}}(-\phi, \pi-\theta, \psi) \tag{1.54}$$

$$R^m_{\mu'\chi}(\hat{\mathbf{k}}) = (-)^{m+\mu'} R^m_{\mu',\underline{\chi}}(-\hat{\mathbf{k}}) = (-)^{m+\chi} R^m_{\underline{\mu'},\chi}(-\hat{\mathbf{k}})$$

$$\Delta\hat{\rho}^m_{\mu'\mu}(\mathbf{k}) = (-)^{\mu'+\mu} \Delta\hat{\rho}^{m*}_{-\mu'-\mu}(-\mathbf{k}) \tag{1.55}$$

1.

$$\Delta\hat{\rho'}^{m}_{\chi\mu}(\mathbf{k}) = \sum_{\mu'} \Delta\hat{\rho}^{m}_{\mu'\mu}(\mathbf{k}) R^{m}_{\mu'\chi}(\hat{\mathbf{k}}) \tag{1.56}$$

$$\Delta\hat{\rho'}^{m}_{\chi\mu}(\mathbf{k}) = \sum_{\mu'} (-)^{\mu'+\mu} \Delta\hat{\rho}^{m*}_{-\mu'-\mu}(-\mathbf{k}) R^{m}_{\mu'\chi}(\hat{\mathbf{k}}) \tag{1.57}$$

$$\hat{\gamma}^{m}_{\mu'\mu}(\mathbf{k}) = \sum_{nl\nu} \hat{c}^{mnl}_{\mu\nu}(k) \sum_{\nu'\lambda'} (-)^{\nu'+\nu} \Delta\hat{\rho}^{n}_{\underline{\nu'\nu}}(\mathbf{k}) \begin{pmatrix} m & n & l \\ \mu' & \nu' & \lambda' \end{pmatrix} R^{l}_{\lambda' 0}(\hat{\mathbf{k}}) \tag{1.58}$$

$$\hat{\gamma}^{m*}_{\underline{\mu'\mu}}(-\mathbf{k}) = \sum_{nl\nu} \hat{c}^{mnl}_{\mu\nu}(k) \sum_{\nu'\lambda'} (-)^{\nu'+\nu} \Delta\hat{\rho}^{n}_{\underline{\nu'\nu}}(\mathbf{k}) \begin{pmatrix} m & n & l \\ \mu' & \nu' & \lambda' \end{pmatrix} R^{l*}_{\underline{\lambda'}0}(-\hat{\mathbf{k}}) (-)^{l+\mu+\nu'} \tag{1.59}$$

$$\hat{\gamma}^{m}_{\mu'\mu}(\mathbf{k}) = \sum_{nl\nu} \hat{c}^{mnl}_{\mu\nu}(k) \sum_{\nu'\lambda'} (-)^{\nu'+\nu} \Delta\hat{\rho}^{n*}_{\nu'\nu}(-\mathbf{k}) \begin{pmatrix} m & n & l \\ \underline{\mu'} & \underline{\nu'} & \underline{\lambda'} \end{pmatrix} R^{l}_{\underline{\lambda'}0}(-\hat{\mathbf{k}}) (-)^{\nu'+\nu+m+n} \tag{1.60}$$

(prove?):
(proof: should refer to eq (1.6) in Luc's notes. So add this set of equations):
(But I just can't deduce it !!!!!)

$$R^{m}_{\mu'\chi}(\hat{\mathbf{k}}) = (-)^{m} R^{m}_{\mu',-\chi}(-\hat{\mathbf{k}}) = (-)^{m+\mu'+\chi} R^{m}_{-\mu',\chi}(-\hat{\mathbf{k}}) \tag{1.61}$$

which gives

$$\Delta\hat{\rho'}^{m}_{\chi\mu}(\mathbf{k}) = (-)^{m+\mu+\chi} \Delta\hat{\rho'}^{m*}_{\chi,-\mu}(-\mathbf{k}) \tag{1.62}$$

Thanks to the symmetry (eq. (appendix))

$$\hat{c'}^{mn}_{\mu\nu,\chi}(k) = (-)^{m+n+\mu+\nu} \hat{c'}^{mn*}_{\underline{\mu\nu},\chi}(k) \tag{1.63}$$

$$\hat{c'}^{mn}_{\mu\nu,\chi}(k) = (-)^{m+n} \hat{c'}^{nm}_{\nu\mu,\chi}(k) \tag{1.64}$$

$$\hat{c'}^{mn}_{\mu\nu,\underline{\chi}}(k) = \hat{c'}^{mn}_{\underline{\mu\nu},\chi}(k) = (-)^{m+n} \hat{c'}^{mn*}_{\mu\nu,\chi}(k) \tag{1.65}$$

$\hat{\gamma'}^{m}_{\chi\mu}(\mathbf{k})$ possesses the same symmetry. Thus the**OZ!** equation can be reduced by a factor of two. THIS IS NOT EXPLICIT ENOUGH; there are several choices to exploit

the symmetry. You should explain at least your choice and how YOU personnaly DID in your code

(In the code I did only the first step 1.48 and the last step)

In the implementation, it is a natural choice to reduce the calculation by two as either the **FFT!**3D or the **FGSHT!** process for real function gives implicitly only a half of information.

It should be noted that not exactly the half of points are calculated. If we choose to calculate a half of $\mathbf{k}$, as shown in figure 1.3, where the 2D plan corresponds two of the three dimensions in k-space grid, the green points can be generated from the black points by the symmetries of eq. (1.48), but the red points should be all calculated, of which the corresponding points is also a red point or even itself. This ever caused a huge problem in the implementation, as we put $\Delta\hat{\rho}^{m}_{\mu'\mu}(\mathbf{k})$ and $\hat{\gamma}^{m}_{\mu'\mu}(\mathbf{k})$ in the same array for reason of memory. It should be assured that these points are calculated only once.

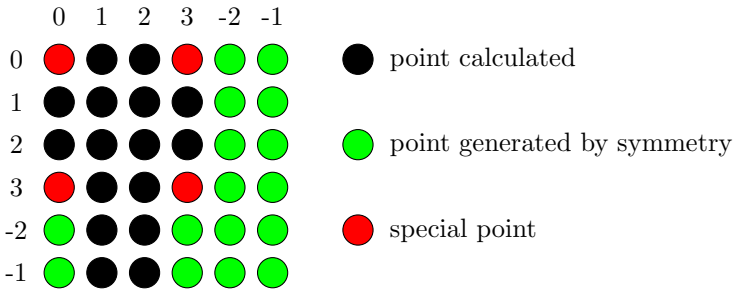

Figure 1.3: Distribution of points to be calculated according to symmetry in a 2D plan