

ANGULAR INTEGRATION IN EXCESS FUNCTIONAL

As discussed in last chapter, the Fourier transform of the excess functional gradient is:

$$\hat{\gamma}(\mathbf{k}, \boldsymbol{\Omega}_1) = \int d\boldsymbol{\Omega}_2 \Delta \hat{\rho}(\mathbf{k}, \boldsymbol{\Omega}_2) \hat{c}(\mathbf{k}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2) \quad (1.1)$$

It should be pointed out that the direct correlation function (**DCF!**), $\hat{c}(\mathbf{k}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2)$, used as an input data in eq. (1.1) is very memory-costly. In the previous work [**gendre_classical_2009**, **Zhao_2011**, **borgis_molecular_2012**], the **DCF!** has been stocked in the intermolecular form $\hat{c}(k, \boldsymbol{\omega}_1, \boldsymbol{\omega}_2)$ to profit an economy of memory, where $(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2) \equiv (\cos \theta_1, \cos \theta_2, \phi_{12})$, and the correspondence of $(\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2)$ to $(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2)$ is calculated directly in the code. These works adapt well with linear solvents, but are proved less powerful for molecular solvent such as water. However, in the case of full Euler angles intermolecular **DCF!** (fig. 1.1),

$$\hat{c}(k, \boldsymbol{\omega}_1, \boldsymbol{\omega}_2) \equiv \hat{c}(k, \cos \theta_1, \cos \theta_2, \phi, \psi_1, \psi_2) \quad (1.2)$$

neither the storage of $\hat{c}(\mathbf{k}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2)$ which is definitively impossible, nor the direct calculation of correspondence $(\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2)$ to $(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2)$ due to the increased complexity that makes it too costly, can be regard as a possible solution. For instance, with a normal setting of 64^3 spatial grid and a Lebedev quadrature of order 2 (14 angles for Θ and Φ), and 3 Ψ -angles, even if the **DCF!** is stocked in simple precision, it takes $64^3 \times 42^2 \times 4 \text{ bytes} = 1.76 \text{ GB}$, and for a Lebedev quadrature of order 5 and correspondingly 5 Ψ -angles, it takes $64^3 \times 250^2 \times 4 \text{ bytes} = 65.5 \text{ GB}$. As a normal PC has only 4 to 16 GB of RAM, it can cause a memory leak.

Therefore, two strategies are developed treat the full **DCF!** case. The first one is a direct extension of the previous work, which uses the full intermolecular **DCF!** with a more complicate angle correspondence pre-tabulated in the beginning of the implementation. The other calculates the **DCF!** directly from rotational invariant projections. The test below gives a complete discussion of these two strategies.

1.1 USING FULL INTERMOLECULAR DCF

The first strategy works in the so-called

To store the **DCF!** in the intermolecular coordinates system, it can be defined as $\hat{c}(k, \boldsymbol{\omega}_1, \boldsymbol{\omega}_2)$, where $(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2) \equiv (\cos \theta_1, \cos \theta_2, \phi, \psi_1, \psi_2)$ and k is always oriented along the z axis (figure 1.1) such that only 6 variables are needed instead of 9 for $\hat{c}(\mathbf{k}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2)$, and the storage is considerably reduced. The transform from $\hat{c}(k, \boldsymbol{\omega}_1, \boldsymbol{\omega}_2)$ to $\hat{c}(\mathbf{k}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2)$ relies on the correspondence $\boldsymbol{\omega}(\mathbf{k}, \boldsymbol{\Omega}) \equiv (\cos \theta, \phi, \psi)$, which is pre-calculated as a table of data.

Finding $\boldsymbol{\omega}$ from $\boldsymbol{\Omega}$ amounts to defining the correspondence between the rotation matrices of the two coordinate systems.

The rotation matrix $\hat{\mathbf{R}}_{\boldsymbol{\Omega}}$ rotates the solvent molecule from \mathbf{I} to its orientation $\hat{\mathbf{R}}_{\boldsymbol{\Omega}}$:

$$\hat{\mathbf{R}}_{\boldsymbol{\Omega}} \mathbf{I} = \hat{\mathbf{R}}_{\boldsymbol{\Omega}} \quad (1.3)$$

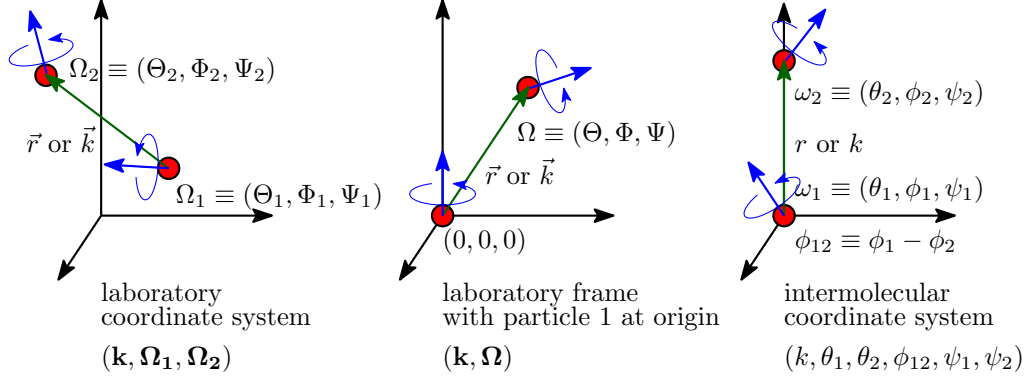


Figure 1.1: Molecules 1 and 2 in different coordinate systems. The laboratory coordinate system is the coordinate system of our grid with a fixed reference view. When one of the molecules is considered as the reference, as the solute in the case of $\rho(\mathbf{r}, \mathbf{\Omega})$, only one orientation needs to be described. For the intermolecular frame, in \mathbf{r} -space, the z axis is oriented along the vector $\mathbf{r}_{12} = \mathbf{r}_2 - \mathbf{r}_1$, or in \mathbf{k} -space is oriented along the vector \mathbf{k} . An orientation $\mathbf{\Omega} \equiv (\Theta, \Phi, \Psi)$ in laboratory frame corresponds to $\boldsymbol{\omega} \equiv (\theta, \phi, \psi)$ in intermolecular frame.

It can be expressed by 3 rotation operations $\hat{\mathbf{R}}_\Phi$, $\hat{\mathbf{R}}_\Theta$, and $\hat{\mathbf{R}}_\Psi$ which rotate along $z - y - z$ axes (the same convention as defined in Messiah [Messiah] and Gray-Gubbins [Gray-Gubbins]):

$$\begin{aligned} \hat{\mathbf{R}}_\Omega &= \begin{bmatrix} R_{xx} & R_{xy} & R_{xz} \\ R_{yx} & R_{yy} & R_{yz} \\ R_{zx} & R_{zy} & R_{zz} \end{bmatrix} \\ &= \begin{bmatrix} \cos \Phi & -\sin \Phi & 0 \\ \sin \Phi & \cos \Phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \Theta & 0 & \sin \Theta \\ 0 & 1 & 0 \\ -\sin \Theta & 0 & \cos \Theta \end{bmatrix} \begin{bmatrix} \cos \Psi & -\sin \Psi & 0 \\ \sin \Psi & \cos \Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos \Phi \cos \Theta \cos \Psi - \sin \Phi \sin \Psi & -\cos \Phi \cos \Theta \sin \Psi - \sin \Phi \cos \Psi & \cos \Phi \sin \Theta \\ \sin \Phi \cos \Theta \cos \Psi + \cos \Phi \sin \Psi & -\sin \Phi \cos \Theta \sin \Psi + \cos \Phi \cos \Psi & \sin \Phi \sin \Theta \\ -\sin \Theta \cos \Psi & \sin \Theta \sin \Psi & \cos \Theta \end{bmatrix} \end{aligned} \quad (1.4)$$

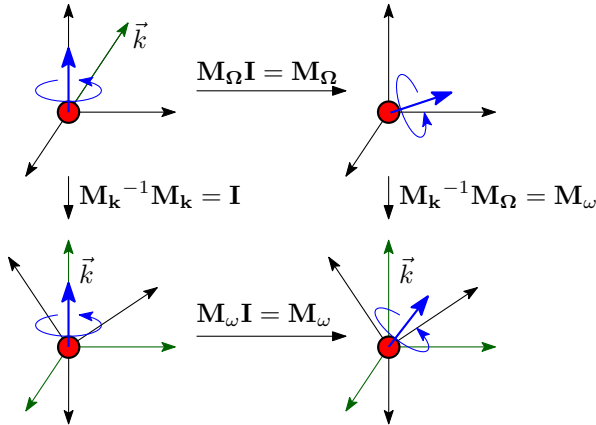


Figure 1.2: Rotation matrices

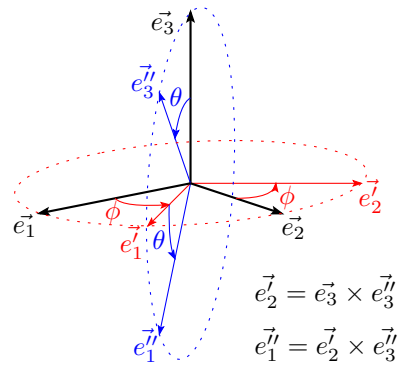


Figure 1.3: Rotation to k-frame

As shown in fig. 1.2, the rotation matrix for transforming the **DCF!** from the intermolecular coordinates to laboratory coordinates $\hat{\mathbf{R}}_\omega$ can be written as:

$$\hat{\mathbf{R}}_\omega = \hat{\mathbf{R}}_\mathbf{k}^{-1} \hat{\mathbf{R}}_\Omega \quad (1.5)$$

with the rotation matrix related to \mathbf{k} vector:

$$\hat{\mathbf{R}}_{\mathbf{k}}^{-1} = \begin{bmatrix} \cos \theta_k \cos \phi_k & \cos \theta_k \sin \phi_k & -\sin \theta_k \\ -\sin \phi_k & \cos \phi_k & 0 \\ \sin \theta_k \cos \phi_k & \sin \theta_k \sin \phi_k & \cos \theta_k \end{bmatrix} \quad (1.6)$$

Here we fix $\psi_k = 0$. As θ_k and ϕ_k are calculated from Cartesian coordinates (k_x, k_y, k_z) , in the extreme cases where we cannot define θ_k (for $\|\mathbf{k}\| = 0$) and ϕ_k (for $k_x^2 + k_y^2 = 0$), we can arbitrarily fix those angles to zero.

A faster way to find the rotation matrix of \mathbf{k} , avoiding the evaluation of trigonometric functions, is shown in figure 1.3, where the matrix can be calculated by the cross products of basis vectors from z axis and \mathbf{k} vector ($\mathbf{k} = \mathbf{e}_3''$):

$$\begin{bmatrix} \mathbf{e}_1'' & \mathbf{e}_2' & \mathbf{e}_3'' \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} \hat{\mathbf{R}}_{\mathbf{k}} = \hat{\mathbf{R}}_{\mathbf{k}} \quad (1.7)$$

The two ways to calculate \mathbf{k} differ only in the case of $\hat{\mathbf{k}} = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^T$, where one is the inverse of the other. This is due to the different definitions of ϕ_k (0 or π when \vec{k}_z superposes with \vec{k}_z) in the two cases. Tests have shown that it has no influence on the final result of the excess functional evaluation.

Therefore, the elements of $\hat{\mathbf{R}}_{\omega}$ can be calculated according to eq. (1.5):

$$\begin{aligned} \hat{\mathbf{R}}_{\omega} &= \begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{bmatrix} \\ &= \begin{bmatrix} \cos \phi \cos \theta \cos \psi - \sin \phi \sin \psi & -\cos \phi \cos \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \sin \theta \\ \sin \phi \cos \theta \cos \psi + \cos \phi \sin \psi & -\sin \phi \cos \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \sin \theta \\ -\sin \theta \cos \psi & \sin \theta \sin \psi & \cos \theta \end{bmatrix} \end{aligned} \quad (1.8)$$

The angles ω are thus found as:

$$\begin{aligned} \cos \theta &= w_z \\ \phi &= \arccos(w_x / (w_x^2 + w_y^2)^{\frac{1}{2}}) \\ \psi &= \arccos(-u_z / (u_z^2 + v_z^2)^{\frac{1}{2}}) \end{aligned} \quad (1.9)$$

The resulting angles are between normal intervals, $\cos \theta \in [-1, 1]$, $\phi \in [0, 2\pi]$. As water possesses C_{2v} symmetry, we take $\psi \in [0, \pi]$.

Here the **DCF!** $c(k, \omega_1, \omega_2) \equiv c(k, \cos \theta_1, \cos \theta_2, \phi_{12}, \psi_1, \psi_2)$ is stored in a discrete set of angles for each value of k (typically (8, 8, 8, 8, 8) in the case of water, which uses the symmetries in §?? to reduce the number of ϕ and ψ by two) such that the correspondence from (Ω_1, Ω_2) to (ω_1, ω_2) usually falls in between angular grid points of the intermolecular grid. An interpolation can be done at different orders: zeroth order interpolation, which directly takes the nearest point, or linear interpolation.

1.1.1 Zero-order interpolation of DCF

At this order, for each possible value of \mathbf{k} and Ω , the corresponding $\cos \theta$ and ψ which relate to a single solvent molecule are stored as an index (single precision integer), which gives the nearest angle in a pre-defined table:

$$\begin{aligned} i_{\cos \theta} &= \lfloor (\cos \theta + 1)(n_{\cos \theta} / 2) \rfloor + 1 \\ i_{\psi} &= \text{mod}(\lfloor \psi(n_{\psi} / \pi) \rfloor, n_{\psi}) + 1 \end{aligned} \quad (1.10)$$

where $\lfloor f \rfloor$ is the floor function. For the angle ϕ which relate to two solvent molecules, the operation $\phi = \phi_1 - \phi_2$ introduces a double error when integer indices are used, as shown in figure 1.4.

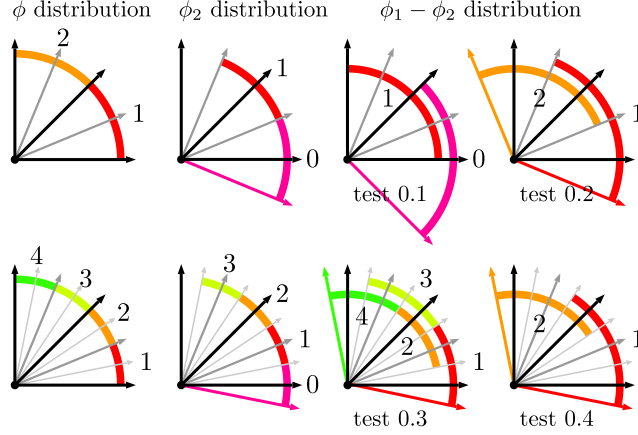


Figure 1.4: $\phi_1 - \phi_2$ distribution: Test 0.1 is the direct subtraction of ϕ established in the same way with θ and ψ , as shown in the top first schema. Test 0.2 tabulates ϕ_2 by taking the nearest point in another manner, as shown in the second schema. In test 0.3-0.4, all ϕ or only ϕ_2 is doubled.

In the actual implementation, as an integer takes 4 bytes and a real takes 8 bytes, there is no profit to tabulate ϕ in integer two times, thus ϕ is stored directly in real.

1.1.2 Linear interpolation of DCF

At this order, $\omega(\mathbf{k}, \mathbf{\Omega})$ is stored in double precision. All angles are stored in real type, and the corresponding **DCF!** is calculated as

$$c(\omega) = w_0 c(\omega_0) + w_1 c(\omega_1) \quad (1.11)$$

where $w_0 = \frac{\omega_1 - \omega}{\omega_1 - \omega_0}$ and $w_1 = \frac{\omega - \omega_0}{\omega_1 - \omega_0}$. Here ω is one of the 5 dimensions in $\tilde{\omega}(\mathbf{k}, \mathbf{\Omega}_1, \mathbf{\Omega}_2) \equiv (\cos \theta_1, \cos \theta_2, \phi, \psi_1, \psi_2)$, ω_0 and ω_1 are the 2 nearest value points, while other variables are fixed. If we express the weight for each dimension as $w_{n_i}^i$ where $i = 1, 2, 3, 4, 5$ is the i th variable, the total equation with 5 variables is:

$$c(\tilde{\omega}) = \left[\sum_{n_1=0}^1 \sum_{n_2=0}^1 \sum_{n_3=0}^1 \sum_{n_4=0}^1 \sum_{n_5=0}^1 \left(\prod_i^5 w_{n_i}^i c(\tilde{\omega}_{n_1, n_2, n_3, n_4, n_5}) \right) \right] \quad (1.12)$$

These two equations are available for both interpolation and extrapolation, where the latter applies, e.g., for $\cos \theta_1$ and $\cos \theta_2$.

An error evaluation of these two strategies of interpolation is shown in appendix ???. Results demonstrate that the linear interpolation scheme is absolutely essential. On the other hand, as seen in eq. (1.12), it is computationally much more expensive than the simple histogram scheme as it requires $2^5 = 32$ times the number of operations.

1.2 DIRECT CALCULATION OF DCF FROM ROTATIONAL INVARIANT PROJECTIONS

Another strategy to calculate $\hat{c}(\mathbf{k}, \mathbf{\Omega}_1, \mathbf{\Omega}_2)$ is to use the **DCF!** expressed in terms of rotational invariant projections, which takes far less memory than in the intermolecular form thanks to their angular independence and symmetric properties.

1.2.1 Using projections in form of $\hat{c}_{\mu\nu}^{mnl}(k)$

As described by Blum [Blum_I, Blum_II], $\hat{c}(\mathbf{k}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2)$ can be expanded as

$$\hat{c}(\mathbf{k}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2) = \sum_{mnl\mu\nu} \hat{c}_{\mu\nu}^{mnl}(k) \Phi_{\mu\nu}^{mnl}(\hat{\mathbf{k}}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2) \quad (1.13)$$

where $\Phi_{\mu\nu}^{mnl}(\hat{\mathbf{k}}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2)$ are rotational invariants that depends on both the spatial and angular coordinates of the two particles (detailed in appendix ??).

For projections of order $n_{\max} = 1$ ($n, m \leq 1$), the **DCF!** can be expressed in very simple form. Only 4 projections $\hat{c}^{mnl}(k)$ are independent: $\hat{c}_S \equiv \hat{c}^{000}$, $\hat{c}_\Delta \equiv \hat{c}^{110}$, $\hat{c}_D \equiv \hat{c}^{112}$ and $\hat{c}_+ \equiv \hat{c}^{011} = -\hat{c}^{101}$, with the corresponding rotational invariants expressed below both in laboratory and intermolecular frames:

$$\begin{aligned} \Phi^{000} &= 1 \\ \Phi^{011} &= i\mathbf{k} \cdot \boldsymbol{\Omega}_1 = i \cos \theta_1 \\ \Phi^{101} &= i\mathbf{k} \cdot \boldsymbol{\Omega}_2 = i \cos \theta_2 \\ \Phi^{110} &= -\sqrt{3}\boldsymbol{\Omega}_1 \cdot \boldsymbol{\Omega}_2 = -\sqrt{3}(\sin \theta_1 \sin \theta_2 \cos \phi_{12} + \cos \theta_1 \cos \theta_2) \\ \Phi^{112} &= \sqrt{\frac{3}{10}} [3(\mathbf{k} \cdot \boldsymbol{\Omega}_1)(\mathbf{k} \cdot \boldsymbol{\Omega}_2) - \boldsymbol{\Omega}_1 \cdot \boldsymbol{\Omega}_2] \\ &= \sqrt{\frac{3}{10}} (2 \cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2 \cos \phi_{12}) \end{aligned} \quad (1.14)$$

where the orientations in laboratory frame $\boldsymbol{\Omega}$ are here expressed as an orientational vector $\boldsymbol{\Omega} = (\sin \Theta \cos \Phi, \sin \Theta \sin \Phi, \cos \Theta)$ in the cartesian coordinate system. To express the **DCF!** at higher orders, the number of **FE!** needed for $\Phi_{\mu\nu}^{mnl}(\hat{\mathbf{k}}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2)$ becomes huge and the **DCF!** should be calculated in intermolecular frame as indicated above.

1.2.2 Using projections in form of $\hat{c}_{\mu\nu,\chi}^{mn}(k)$

Compared to the expression of $\Phi_{\mu\nu}^{mnl}(\hat{\mathbf{k}}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2)$ in laboratory frame (eq. (??) in appendix ??), its intermolecular form has far fewer terms (eq. (??) in appendix ??), such that

$$\hat{c}(k, \boldsymbol{\omega}_1, \boldsymbol{\omega}_2) = \frac{1}{2l+1} \sum_{mn\mu\nu\chi} \hat{c}_{\mu\nu,\chi}^{mn}(k) r_{\chi\mu}^m(\theta_1) r_{-\chi\nu}^n(\theta_2) e^{-i\chi(\phi_{12} \equiv \phi_1 - \phi_2)} e^{-i\mu\psi_1} e^{-i\nu\psi_2} \quad (1.15)$$

where r is the generalized Legendre polynomial, $m, n \leq n_{\max}$, $|\mu| \leq m$, $|\nu| \leq n$, and $\chi \in [-\min(m, n), \min(m, n)]$.

$r_{\chi\mu}^m(\theta)$, $e^{-i\chi\phi}(\phi)$ and $e^{-i\mu\psi}(\psi)$ can be separately pre-tabulated for each given \mathbf{k} , to avoid double evaluation of each term.

Eq. (1.15) replaces the interpolation of eq. (1.12) by an exact formula and it requires the projections $\hat{c}_{\mu\nu,\chi}^{mn}(k)$ to be stored in memory rather than the full angular representation $\hat{c}(k, \boldsymbol{\omega}_1, \boldsymbol{\omega}_2)$. It also requires the passage from orientations in laboratory frame to orientations in intermolecular frame, i.e. use of the formulae (1.9) for each \mathbf{k} vector.