Multislice Diffraction Functions

Erik Bigwood

7/23/17

Contents

1 Relativistic functions **I**
1.1 lambda_from_eV I
1.2 mass_from_eV I

2 Geometric functions **I**
2.1 rotate_vec_array I

3 Lattice functions **II**
3.1 lattice_populate_single II
3.2 lattice_populate_fcc II
3.3 detector_populate III

4 Geometric Diffraction functions **III**
4.1 phase_point III

1 Relativistic functions

1.1 lambda_from_eV

This function takes an energy in eV and returns the relativistic electron wavelength. The equation is shown below:

$$\lambda = \sqrt{\frac{h^2c^2}{E^2 - m_0^2c^2}} \tag{1}$$

1.2 mass_from_eV

Returns relativistic electron mass from kinetic energy in eV.

$$m = \sqrt{\frac{K^2 + 2Km_0c^2}{c^4} + m_0^2} \tag{2}$$

2 Geometric functions

2.1 rotate_vec_array

Iterates over each vector in the n x 3 array Λ and rotates them around $\hat{x}$, $\hat{y}$, then $\hat{z}$ by tx, ty, and tz (rad) respectively. That is,

$$\Big(\forall v \in \Lambda\Big)\Big(v \to R_z(\theta_z) \cdot R_y(\theta_y) \cdot R_x(\theta_x) \cdot v = \mathcal{R}(v - \vec{\mu}) + \vec{\mu}\Big) \tag{3}$$

rotation_mat Returns a matrix corresponding to a rotation around $\hat{x}$, $\hat{y}$, then $\hat{z}$ by tx, ty, and tz (rad) respectively.[1] Note – rotates around the mean vector $\vec{\mu}$.

That is,

$$\mathcal{R} = R_z(\theta_z) \cdot R_y(\theta_y) \cdot R_x(\theta_x) \tag{4}$$

rotation_mat_x Returns the rotation matrix around $\hat{x}$ by θ_x, $R_x(\theta_x)$.

rotation_mat_y Returns the rotation matrix around $\hat{y}$ by θ_y, $R_y(\theta_y)$.

rotation_mat_z Returns the rotation matrix around $\hat{z}$ by θ_z, $R_z(\theta_z)$.

translate_vec_array Translates each vector in a $n \times 3$ array of vectors $\vec{l}$ by a vector $\vec{t}$.

$$\Big(\forall \vec{v} \in \vec{l}\Big)\Big(\vec{v} \to \vec{v} + \vec{t}\Big) \tag{5}$$

[1] See rotation matrices.

mean_vector_array Returns the mean vector $\vec{\mu}$ of a $n \times 3$ array of vectors $\vec{l}$, by summing over each vector.

$$\vec{\mu} = \frac{1}{n} \sum_{i=0}^{n} \vec{l}_i \tag{6}$$

rotate_translate_array Returns the array of vectors $\vec{l}$ rotated around its center by $\vec{\theta}$ and then translated by $\vec{t}$.

$$\left(\forall \vec{v} \in \vec{l}\right)\left(\vec{v} \to \text{rotate\_vec\_array}\big(\vec{v}\big) + \vec{t}\right) \tag{7}$$

rand_vec_len Returns a random vector with length l.

3 Lattice functions

3.1 lattice_populate_single

Creates and populates a simple cubic lattice with side number n, lattice parameter *clen*, and centered at *latt_center*.

$$\left(\forall a_i, i \in [0, n]_{\mathbb{Z}}, [0, 2]_{\mathbb{Z}}\right)\left(\vec{l}_{a_0 n^2 + a_1 n + a_2} = clen\big\langle a_0 - n/2, a_1 - n/2, a_2 - n/2\big\rangle\right) \tag{8}$$

3.2 lattice_populate_fcc

Creates, populates, and prunes a spherical nanoparticle with radius r, lattice parameter c, and centered at *latt_center*. There is an internal overestimation parameter f, to ensure full population, currently set at 2.

$$n = \text{int}\left(\text{floor}(2rf)\right)$$

$$\left(\forall a_i, i \in [-n/2, n/2]_{\mathbb{Z}}, [0, 2]_{\mathbb{Z}}\right)\left(\vec{l}_{a_0 n^2 + a_1 n + a_2} = c\big((a_0)\vec{v}_0 + (a_1)\vec{v}_1 + (a_2)\vec{v}_2\big)\right)$$

Where $\vec{v}_i$ are the primitive vectors for a fcc lattice

$$\vec{v}_0 = \big\langle 0, 0.5, 0.5\big\rangle$$
$$\vec{v}_1 = \big\langle 0.5, 0, 0.5\big\rangle$$
$$\vec{v}_2 = \big\langle 0.5, 0.5, 0\big\rangle$$

Then, mark vectors outside the radius:

$$\left(\forall \vec{v} \in \vec{l}\right)\left(||\vec{v} - latt\_center|| > r^2 \implies \vec{v} \to \left\langle 0, 0, 0 \right\rangle\right) \tag{9}$$

Now remove all zero vectors from $\vec{l}$ to prune the nanoparticle.

3.3 detector_populate

Creates and populates a square detector $\vec{d}$ with width w, n^2 elements, and centered at $\vec{t}$.

$$\left(\forall a_i, i \in [0, n]_{\mathbb{Z}}, [0, 1]_{\mathbb{Z}}\right)\left(\vec{d}_{a_0 n + a_1} = w\left\langle (a_0 - n/2)/n, (a_1 - n/2)/n, 0 \right\rangle\right) + \vec{t} \tag{10}$$

4 Geometric Diffraction functions

4.1 phase_point

Returns the geometric phase at $\vec{p}$ from a lattice $\vec{l}$, with wavelength λ.

$$\phi(\vec{p}, \vec{l}) = \sum \cos\left(\frac{2\pi}{\lambda}\sqrt{\sum_i \left(\vec{l}e'_i - \vec{p}e'_i\right)^2}\right) \tag{11}$$