POCSI: POint Charge SImulation of electric field gradients

Jonathan Frassineti

November 18, 2022

1 Introduction

1.1 Introduction and aim of the package

This Python package allows the calculation of the Electric Field Gradient (EFG, see https://en.wikipedia.org/wiki/Electric_field_gradient) using the Point Charge approximation.

The basic concept behind this model is to approximate each ion in a crystal lattice by its ionic charge, centered at a point of radius zero. In general, the EFG is a difficult quantity to compute, since it must be derived by differentiating the potential

$$V(r_0) = \frac{1}{4\pi\epsilon_0}\int \frac{\rho(r')}{r} d\tau' \tag{1}$$

where the integral is taken over a volume swept out by r', and the vector $r = r' - r_0$. So long as the charge distribution, ρ(r), is unknown, this integral cannot be solved. Since simulations in this project will assume a point charge distribution, however, the equation can be simplified to

$$V(r_0) = \frac{1}{4\pi\epsilon_0}\sum_k \frac{q_k}{r'_k} \tag{2}$$

by substituting in the charge distribution of k point charges $\rho(r) = \sum_k q_k \delta^3(r - r_k)$.

This equation for V(r_0) is a quantity that is at least possible to differentiate, as can be easily shown:

$$\begin{aligned} V_{ij}(r_0) &= \frac{\partial^2}{\partial x_i \partial x_j}\left(\frac{1}{4\pi\epsilon_0}\sum_k \frac{q_k}{r'_k}\right) \\ &= \frac{1}{4\pi\epsilon_0}\sum_k q_k \left(\frac{\partial^2}{\partial x_i \partial x_j}\frac{1}{r'_k}\right) \\ &= \frac{1}{4\pi\epsilon_0}\sum_k V_{ij}^{(k)} \end{aligned}$$

where some algebra will yield

$$V_{ij}^{(k)} = q_k\left(\frac{3x'^i 3x'^j - \delta_{ij}r'^2_k}{r'^5_k}\right) \tag{3}$$

This quantity is the (unscaled) contribution of each charge towards the (i,j)th component of the EFG tensor. Thus, to calculate the net EFG, one need only specify the partial EFG due to each point charge, and sum each contribution.

2 Usage of package

- In the file *Crystalline_structure.py*, the user can insert the atomic base of the material under study.
- In the file *EFG_functions.py*, there are the definitions of the functions used to calculate the EFG. There is no need to modify this file.
- In the file *EFG_point_charge.py*, the user can finally calculate the EFG using the point charge approximation and diagonalize it. From this, the quadrupolar parameters can be extracted.

At the end, the user has to interact only with files 1) and 3).

3 Results

As an example, we report here the EFG calculated by using the point charge simulation for V_3Si, a very well known superconductor with A15 cubic structure. The total EFG from point charge of a V nucleus inside V_3Si is equal to:

$$\begin{pmatrix} -2.30 & 0.00 & 0.00 \\ 0.00 & -2.30 & 0.00 \\ 0.00 & 0.00 & 4.60 \end{pmatrix}$$

The values have to be multiplied by 10^{20}, and they are expressed in V/m^2. The result is in good agreement with NMR measurements on pure V_3Si, where it is shown that the EFG tensor is perfectly diagonal with the asymmetry parameter $\eta = 0$.

Since this model is an approximation, it does not include any possible shielding or antishielding (Sternheimer) factor, that can improve the agreement between experimental and simulated EFGs.