

Fitting the Full Shape of the Scattering Rate in the Relaxation Time Approximation of Boltzmann Transport Using a Finite Element Method

Saleh Shamloo

September 18, 2026

1 Context

We have been using the Boltzmann transport equation in the relaxation time approximation to analyze the transport behavior of various quantum materials, often with great success. However, this analysis involves a lot of guesswork and manual testing when it comes to implementing an appropriate mathematical form for the scattering rate. This means there is always some uncertainty when we run into contradictions with the analysis. We start to think "Maybe we just have not found a smart enough ansatz." Therefore, it is hard to be certain when we have actually ran into a limitation of the method itself (for example, a breakdown of the relaxation time approximation).

To extract the scattering rate from experimental data, we rely on functions that respect the symmetries of the lattice and that we can flexibly tune it terms of their "intensity" and "spread". For example, for Nd-LSCO, we have

$$1/\tau_{\mathbf{k}} = 1/\tau_{\text{isotropic}} + 1/\tau_{\text{anisotropic}}, \quad (1)$$

$$1/\tau_{\text{anisotropic}} = \Gamma_{\mathbf{k}} |\cos(2\phi)|^\nu, \quad (2)$$

where ϕ is the in-plane angle and $\Gamma_{\mathbf{k}}$ and ν can set the aforementioned intensity and spread of peaks around the antinodal points on the Fermi surface (that is, the points $(\pi, 0)$, $(0, \pi)$, $(-\pi, 0)$, $(0, -\pi)$). This provides a flexible, easy-to-interpret mathematical form for the scattering rate. After this, our job is to fit the free parameters of this model to the experiment. This is usually an Angle-Dependent Magnetoresistance (ADMR) measurement that can sufficiently constrain the parameters of the fit. Next, we check the consistency with other experiments, and if we indeed find consistency, we know that we have successfully found a fundamental property of the scattering of the system. This method is based on a physical understanding of the system, and has strong explanatory power due to the limited number of parameters involved.

However, what happens if we fail to find *any* suitable form for the scattering rate that can self-consistently explain multiple experiments (e.g. the ADMR and the Hall effect)? How can we be sure that a solution does not exist at all, and it is not that we have not searched well enough? The only way to know for sure is to extract the full shape of the scattering rate from the data, without any imposed ansatz. This is the goal of this project. **TODO: show examples with figures of Nd-LSCO and LSCO data.**

2 Boltzmann Transport Solved with an FEM

In a previous report, we have fully derived how one can solve the Boltzmann transport equation using a Finite Element Method (FEM). I will just be using the results here. You can consult the other report for computational details.

The central equation for calculating the conductivity is

$$\sigma_{ab} = \frac{2e^2}{(2\pi)^d \hbar} \sum_{ij} \left[\left(\sum_k M_{ik} (\hat{v}_a)^k \right) (A_{ij})^{-1} \left(\sum_k M_{jk} (\hat{v}_b)^k \right) \right]. \quad (3)$$

where the matrix M is the “overlap matrix” coming from the chosen FEM basis and the matrix A is

$$\begin{aligned} A_{ij} &= \int d\mathbf{k} \left(-i\omega + \frac{1}{\tau(\mathbf{k})} \right) \frac{\phi_i(\mathbf{k}) \phi_j(\mathbf{k})}{v(\mathbf{k})} \\ &\quad - \int d\mathbf{k} \phi_i(\mathbf{k}) \frac{e(\hat{\mathbf{v}}(\mathbf{k}) \times \mathbf{B})}{\hbar} \cdot \nabla_{\mathbf{k}} \phi_j(\mathbf{k}) \\ &\quad - \int d\mathbf{k} \int d\mathbf{k}' \phi_i(\mathbf{k}) \frac{C(\mathbf{k}, \mathbf{k}')}{v(\mathbf{k})} \phi_j(\mathbf{k}') \\ &\equiv \Gamma_{ij} - \frac{eB}{\hbar} (D_{\hat{\mathbf{v}} \times \hat{\mathbf{B}}})_{ij} + S_{ij}. \end{aligned} \quad (4)$$

where $1/\tau(\mathbf{k})$ is the scattering rate, $C(\mathbf{k}, \mathbf{k}')$ is the scattering kernel, ω is the oscillation frequency of the externally applied electromagnetic fields, and the ϕ functions are FEM basis functions. In the relaxation time approximation, we neglect the last term (involving the explicit use of the full scattering kernel).

We assume a piecewise linear basis functions on a triangulated Fermi surface. If we define α_{ijk} as the tensor equal to the jacobian converting area of the triangle containing the vertices i , j , and k , and set it equal to 0 when the mesh does not contain a triangle between these vertices, then we define the area matrix α as

$$\alpha_{ij} = \sum_{(i,j,k) \in T_i \cap T_j} \alpha_{ijk}, \quad (6)$$

where T_i is the triangle containing the vertex i . When $i = j$, this is just a sum over the triangles that contain this one node with index $i = j$. Next, we define the local derivative matrix $\mathbf{d}_{ij}$ as

$$\mathbf{d}_{ij} = \sum_{(i,j,k) \in T_i \cap T_j} \text{sgn}(i, j, k) (\mathbf{k}_i - \mathbf{k}_k), \quad (7)$$

Where the $\text{sgn}(i, j, k)$ function comes from the sign needed for the orientation of the triangle, such that the normals point outward. Finally, we define the inverse scattering length vector γ as

$$\gamma^k = -\frac{i\omega}{v_k} + \frac{1}{v_k \tau_k}, \quad (8)$$

then the different matrices can be expressed as

$$M_{ij} = \frac{\alpha_{ij}}{24}\delta_{ij} + \frac{\alpha_{ij}}{24}, \quad (9)$$

$$\Gamma_{ij} = \frac{1}{60} \left(\sum_k \alpha_{ik} \gamma^k \delta_{ij} + \alpha_{ij} (\gamma^i + \gamma^j) + \sum_k \frac{\alpha_{ijk} \gamma^k}{2} \right) \quad (10)$$

$$(D_{\hat{\mathbf{v}} \times \hat{\mathbf{B}}})_{ij} = \frac{1}{6} \hat{\mathbf{B}} \cdot \mathbf{d}_{ij}. \quad (11)$$

Note that, compared to the report these equations are taken from, I have removed the unneeded neighboring node Kronecker deltas (like $\delta_{\langle ij \rangle}$) and merged the vector α_i into the matrix α_{ij} . In order to avoid confusion, I will not be using the appropriate subscript/superscript (covariant/constravariant) position for the indices in the rest of this document. They are not needed for the mathematics here.

2.1 The Optimization Problem

From here, in principle, we have everything we need to fit the scattering rate at every node on the Fermi surface using data from an experiment (usually ADMR). However, without the Jacobian, it would be very computationally intensive and difficult to solve this optimization problem. Since we formulated the problem in a linear-algebraic form with our Finite Element Method, it is possible for us to calculate the Jacobian explicitly.

From experiments, the actual data we collect is not the conductivity σ , but its inverse ρ . In pure linear algebra notation,

$$\rho = \frac{(2\pi)^d \hbar}{2e^2} (\hat{V}^T A^{-1} \hat{V})^{-1} \quad (12)$$

where $\hat{V}$ is a matrix containing all the three $M\hat{v}$ vectors on three columns next to each other.

$$\hat{V} = \begin{pmatrix} | & | & | \\ M\hat{v}_x & M\hat{v}_y & M\hat{v}_z \\ | & | & | \end{pmatrix} \quad (13)$$

From here, to get the full shape of the scattering rate, we can fit it at every point discretized on the Fermi surface. To be more exact, we have the components of the vector γ_k (or $1/\tau_k$) as the parameters of the model, and we fit them to the measured *rho* under different conditions (for example, for ADMR, we fit to the ρ_{zz} calculated at different θ and ϕ spherical angles of the magnetic field with the sample).

The challenge with this optimization problem is the computational cost of fitting thousands of parameters at the same time, with a relatively high cost for every iteration of the optimization procedure. However, we can alleviate that and make the optimization problem much more workable by explicitly calculating the gradient of the loss. To calculate the gradient of any type of loss function, it is sufficient to have the derivative of our output values with respect to the fitting parameters.

First, remember that

$$\sigma = \frac{2e^2}{(2\pi)^d \hbar} (\hat{V}^T A^{-1} \hat{V}) \quad (14)$$

Then, taking the derivative of $\rho = \sigma^{-1}$ with respect to each optimization parameter γ_p , we get

$$\frac{\partial \rho}{\partial \gamma_p} = -\sigma^{-1} \frac{\partial \sigma}{\partial \gamma_p} \sigma^{-1} = -\rho \frac{\partial \sigma}{\partial \gamma_p} \rho, \quad (15)$$

with

$$\frac{\partial \sigma}{\partial \gamma_p} = \frac{2e^2}{(2\pi)^d \hbar} \left(\hat{V}^T \frac{\partial A^{-1}}{\partial \gamma_p} \hat{V} \right) = -\frac{2e^2}{(2\pi)^d \hbar} \left(\hat{V}^T A^{-1} \frac{\partial A}{\partial \gamma_p} A^{-1} \hat{V} \right) \quad (16)$$

Since only the Γ matrix in A depends on γ ,

$$\frac{\partial A_{ij}}{\partial \gamma_p} = \frac{\partial \Gamma_{ij}}{\partial \gamma_p} = \frac{1}{60} \left(\alpha_{ip} \delta_{ij} + \alpha_{ij} (\delta_{ip} + \delta_{jp}) + \frac{\alpha_{ijp}}{2} \right). \quad (17)$$

Keep in mind that in equation (16), if we do the correct order of operations, there is no need for many matrix inverses for every different γ_p parameter. The only two big (but sparse) inverses needed are

$$\begin{cases} X := A^{-1} \hat{V}, \\ Y := (A^{-1})^T \hat{V}, \end{cases} \quad (18)$$

$$(19)$$

then the expression for the Jacobian becomes

$$\frac{\partial \rho}{\partial \gamma_p} = \frac{2e^2}{(2\pi)^d \hbar} \rho \left(\hat{Y}^T \frac{\partial \Gamma}{\partial \gamma_p} \hat{X} \right) \rho \quad (20)$$

with the derivative $d\Gamma/d\gamma_p$ calculated in (17). When doing the actual calculations, we are not solving the linear system $A^{-1}V$ for every column in V , as we use a factorization of A . So, we do not need to redo an inverse for $(A^{-1})^T$ either, and it is only a transpose operation followed by another matrix multiplication.

Practically, we do not want to calculate $\partial\Gamma/\partial\gamma_p$ separately and then multiply by X and Y^T , as that would involve making a 3-tensor mostly filled with diagonal terms. The α_{ijp} term, which is the only term forcing a 3-tensor instead of a matrix, only involves contributions involving neighboring nodes of p . Therefore, expanding the matrix multiplication should give us a more efficient construction of the $Y^T(\partial\Gamma/\partial\gamma_p X)$ term, based on neighboring nodes of p instead of tensor multiplication.

$$\left(Y^T \frac{\partial \Gamma}{\partial \gamma_p} X \right)_{ab} = \frac{1}{60} \left(\sum_i \alpha_{ip} Y_{ia} X_{ib} + Y_{pa} \sum_j \alpha_{pj} X_{jb} + X_{pb} \sum_i \alpha_{ip} Y_{ia} + \sum_{i,j} \frac{\alpha_{ijp}}{2} Y_{ia} X_{jb} \right) \quad (21)$$

Since α_{ijp} is only nonzero when i , j , and p all lie on the same triangle, the last term is only a sum over the mesh triangles that contain the node p . This can be separately handled efficiently by a few sums over the triangle indices list. The rest of the terms are simple matrix multiplications.