**RV College of Engineering®**

**(Autonomous Institution Affiliated to VTU, Belagavi)**



**Write a program to calculate the conductivity (longitudinal and transverse) of a material given its eigen functions and eigen values**

***Experiential Learning Report***

Submitted by:

**SHREEJAY PANDEY-1RV22CS187**

**SKANDA P R-1RV22CS199**

**SUHAS RAJ H R-1RV22CS209**

**TALASILA DHEERAJ-1RV22CS216**

Submitted to:

|  |  |
| --- | --- |
| **Dr. Tribikram Gupta** | Assistant professor  Department of Physics |
| **Dr. Nivya Muchikel** | Assistant professor  Department of Mathematics |
| **Dr. Pratiba.D** | Assistant professor  Department of Computer Science |

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sl No.** | CONTENTS | **Page No.** |
| **1.** | INTRODUCTION | **3-5** |
| **2.** | ALGORITHM USING KUBO FORMULA | **6-7** |
| **3.** | ELECTRICAL CONDUCTIVITY CALCULATOR USING TKINTER APPLICATION | **8-11** |
| **4.** | MATLAB CODE | **12-14** |
| **5.** | DIFFERENCE BETWEEN KUBO GREEN FORMULA AND BOLTZMANN TRANSPORT EQUATION | **15** |
| **6.** | REFERENCES | **16** |

**INTRODUCTION:**

1. The Kubo formula for the electrical conductivity tensor is applicable for electrons in applied electric field. This formula correctly accounts for the interaction of charge carriers with a medium. It takes into account the energy levels, the Fermi-Dirac distribution, and the matrix elements between different states, allowing us to compute the electrical conductivity for a quantum system.

2**. Quantum States and Eigenvalues**: In quantum mechanics, physical systems are described by wave functions. The wave function is denoted by ψ, and it contains all the information about the quantum state of the system. When we solve the Schrödinger equation for a given system, we obtain a set of possible wave functions, and the corresponding allowed energies for the system are the eigenvalues (εᵢ).

3. **Fermi-Dirac Distribution**: The Fermi-Dirac distribution function (f) is a key concept in statistical mechanics and quantum statistics. It describes the probability that a state with energy εᵢ will be occupied by a fermionic particle (e.g., an electron) at a given temperature (T) and chemical potential (μ). For a system in thermal equilibrium, the Fermi-Dirac distribution function is given by: f(ε, μ, T) = 1 / [exp((ε - μ) / (k\_B \* T)) + 1], where k\_B is the Boltzmann constant. This function depends on the energy (ε) and the difference between the energy and the chemical potential (ε - μ) of the system.

4. **Fermi Energy (Fermi Level):** In a solid, electrons occupy the energy levels by the Pauli exclusion principle. At zero temperature, the Fermi-Dirac distribution function becomes a step function, and all states with energy below a certain energy value called the Fermi energy (E\_F) are fully occupied by electrons, while all states above E\_F are unoccupied.

5**. Matrix Elements**: Matrix elements represent the probability amplitudes for transitions between quantum states. In the context of the Kubo formula, the matrix element |⟨ψᵢ|v|ψⱼ⟩|² describes the probability of an electron in state |ψᵢ⟩ transitioning to state |ψⱼ⟩ with the help of the velocity operator.

6**. Probability Current Density:** The magnitude of probability current density represents the flux of the particle that is number of particles passing through per unit area per unit time and the direction of probability current density is along the direction of the flow of the particles. It is given by the formula: 

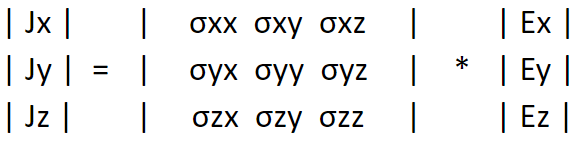
Current density along x axis: Jx = (ħ / 2mi) \* ((ψ\*)(∂ψ/∂x) - (ψ)(∂ψ\*/∂x))

Current density along y axis: Jy = (ħ / 2mi) \* ((ψ\*)(∂ψ/∂y) - (ψ)(∂ψ\*/∂y))

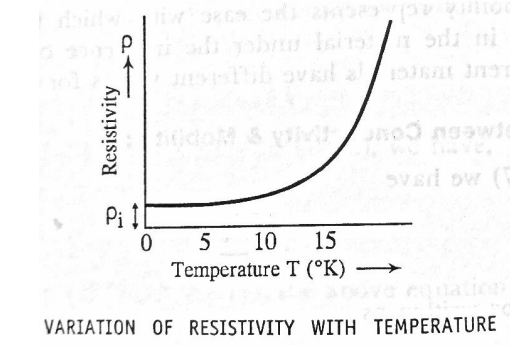
Current density along z axis: Jz = (ħ / 2mi) \* ((ψ\*)(∂ψ/∂z) - (ψ)(∂ψ\*/∂z))

7. **Commutators:** A commutator in quantum mechanics tells us if we can measure two 'observables' at the same time. If the commutator of two 'observables' is zero, then they can be measured at the same time, otherwise there exists an uncertainty relation between the two. For example, Heisenberg Uncertainty principle is a direct consequence of the fact that position and momentum do not commute, therefore we can not precisely determine position and momentum at the same time.

8. **Conductivity tensor in terms of Current Density:** We can express the relationship between current density (J) and electric field (E) using Ohm's law as J = σE, where the conductivity (σ) is represented by a conductivity tensor, which is a 3x3 matrix. This tensor accounts for the three-dimensional nature of both current density and electric field vectors, giving both longitudinal and transverse conductivities. The individual components of the tensor, such as σxx, σyy, and σxy, describe the conductivity in various directions within the three-dimensional space. It is represented as:



9. **Factors affecting Conductivity:** Except for metals that are superconducting, the variation shown in the below graph applies to all conductors. There are two components of resistivity which arises. They are ρ**ph** and ρ**i**.



1) Resistivity ρph due to scattering of electrons by lattice vibrations (phonons) increases with temperature. It is the resistivity exhibited by a pure specimen that is free of all defects, and hence called the ideal resistivity.

2) Resistivity ρi due to scattering of conduction electrons by the presence of impurities, and imperfections. This type of scattering is independent of temperature and contributes to resistivity even at temperature T= 0 kelvin. ρi is called residual resistivity.

If ρ is the total resistivity, then which can be expanded as

At lower temperature, scattering of electrons by lattice ions become less, due to which τph becomes larger and ρph tends to zero, hence at low temperature

When temperature becomes high, the amplitude of lattice vibrations increase, hence resistivity becomes linearly dependant on temperature.

Since, hence the behavior of electrical conductivity with temperature depends on the reciprocal of resistivity.

**ALGORITM USING KUBO FORMULA:**

1. **Input:**

* + Eigenvalues (εᵢ) and corresponding eigenfunctions (|ψᵢ⟩) of the metal system.
  + Fermi energy (E\_F) or Fermi level.
  + Fundamental constants: e (elementary charge) and h (Planck's constant)

2. **Define a small energy threshold E\_th.**

This threshold is used to identify eigenstate pairs with energies close to each other and the Fermi level.

3. **Initialize variables:**

* + Conductivity (σ) = 0

4. **Loop over all pairs of eigenstates (i, j):**

* + Calculate the energy difference: Δε = εᵢ - εⱼ
  + Check if |Δε| ≤ E\_th. If not, skip to the next pair as they don't significantly contribute to the electrical conductivity.
  + Calculate the Fermi-Dirac distribution functions for energies εᵢ and εⱼ: fᵢ = f(εᵢ,ef) and fⱼ = f(εⱼ,ef), where f(ε, ef) is the Fermi-Dirac distribution function.

5. **For each contributing pair (i, j):**

Calculate the matrix element |⟨ψᵢ|v|ψⱼ⟩|², where v is the velocity operator. This matrix element represents the probability amplitude of an electron transitioning from state |ψᵢ⟩ to state |ψⱼ⟩.

6. **Update the conductivity:**

* + Add the contribution of the current pair to the total conductivity:

σ += (e² / h) \* |fᵢ - fⱼ|² \* |⟨ψᵢ|v|ψⱼ⟩|²

7. After processing all contributing pairs, the variable σ will contain the electrical conductivity of the metal.

8. **Output:**

σ: The electrical conductivity of the metal, obtained using the Kubo formula.

**Electrical Conductivity Calculator using Tkinter Application**

import numpy as np

import tkinter as tk

from tkinter import ttk

def fermi\_dirac(energy, fermi\_energy, temperature):

k\_B = 8.617333262145e-5

return 1 / (np.exp((energy - fermi\_energy) / (k\_B \* temperature)) + 1)

def calculate\_conductivity(eigenvalues, eigenfunctions, fermi\_energy, temperature):

conductivity = 0.0

e = 1.602176634e-19

h = 6.62607015e-34

num\_states = len(eigenvalues)

for i in range(num\_states):

for j in range(num\_states):

f\_i = fermi\_dirac(eigenvalues[i], fermi\_energy, temperature)

f\_j = fermi\_dirac(eigenvalues[j], fermi\_energy, temperature)

matrix\_element\_squared = np.abs(np.dot(eigenfunctions[i].conjugate(), eigenfunctions[j]))\*\*2

conductivity += (e\*\*2 / h) \* np.abs(f\_i - f\_j)\*\*2 \* matrix\_element\_squared

return conductivity

def calculate\_and\_display\_conductivity():

eigenvalues\_str = eigenvalues\_entry.get()

eigenvalues = np.array([float(val) for val in eigenvalues\_str.split()])

eigenfunctions\_str = eigenfunctions\_entry.get("1.0", tk.END) # Get text from the Text widget

eigenfunctions\_rows = eigenfunctions\_str.split("\n")

eigenfunctions = []

for row in eigenfunctions\_rows:

eigenfunctions.append(np.array([float(val) for val in row.split()]))

fermi\_energy = float(fermi\_energy\_entry.get())

temperature = float(temperature\_entry.get())

conductivity = calculate\_conductivity(eigenvalues, eigenfunctions, fermi\_energy, temperature)

result\_label.config(text="Electrical conductivity: {:.6e}".format(conductivity))

# Create the main application window

root = tk.Tk()

root.title("Electrical Conductivity Calculator")

# Create and layout GUI elements

frame = ttk.Frame(root, padding=10)

frame.grid(column=0, row=0, sticky=(tk.W, tk.E, tk.N, tk.S))

eigenvalues\_label = ttk.Label(frame, text="Eigenvalues (space-separated):")

eigenvalues\_label.grid(column=0, row=0)

eigenvalues\_entry = ttk.Entry(frame)

eigenvalues\_entry.grid(column=1, row=0)

eigenfunctions\_label = ttk.Label(frame, text="Eigenfunctions (one row per eigenfunction):")

eigenfunctions\_label.grid(column=0, row=1)

eigenfunctions\_entry = tk.Text(frame, width=40, height=4)

eigenfunctions\_entry.grid(column=1, row=1)

fermi\_energy\_label = ttk.Label(frame, text="Fermi Energy:")

fermi\_energy\_label.grid(column=0, row=2)

fermi\_energy\_entry = ttk.Entry(frame)

fermi\_energy\_entry.grid(column=1, row=2)

temperature\_label = ttk.Label(frame, text="Temperature (K):")

temperature\_label.grid(column=0, row=3)

temperature\_entry = ttk.Entry(frame)

temperature\_entry.grid(column=1, row=3)

calculate\_button = ttk.Button(frame, text="Calculate", command=calculate\_and\_display\_conductivity)

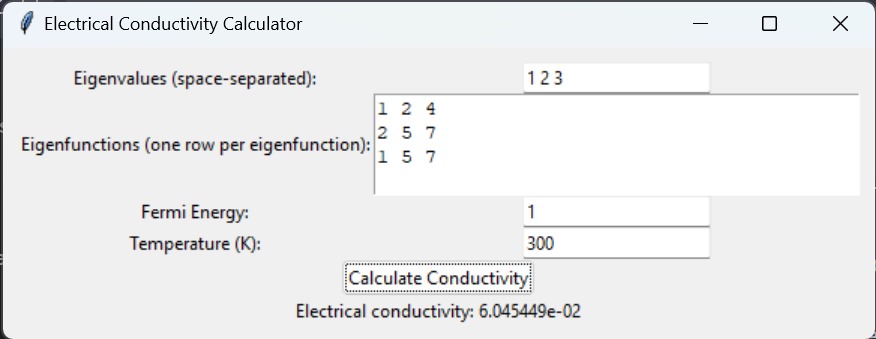
calculate\_button.grid(column=0, row=4, columnspan=2)

result\_label = ttk.Label(frame, text="")

result\_label.grid(column=0, row=5, columnspan=2)

# Start the GUI application

root.mainloop()



**MATLAB CODE:**

% Prompt for the size of the eigenvalues matrix

num\_states = input('Enter the number of eigenvalues: ');

% Prompt for the eigenvalues as a row vector

eigenvalues = zeros(1, num\_states);

for i = 1:num\_states

eigenvalues(i) = input(['Enter eigenvalue ', num2str(i), ': ']);

end

% Prompt for the size of the eigenfunctions matrix

num\_functions = input('Enter the number of eigenfunctions: ');

% Prompt for the eigenfunctions as a matrix

eigenfunctions = zeros(num\_states, num\_functions);

for i = 1:num\_functions

disp(['Enter eigenfunction ', num2str(i), ' (as a column vector): ']);

eigenfunctions(:, i) = input(' ');

end

fermi\_energy = 1.0;

temperature = 300;

conductivity = calculate\_conductivity(eigenvalues, eigenfunctions, fermi\_energy, temperature);

disp(['Electrical conductivity: ', num2str(conductivity)]);

function f = fermi\_dirac(energy, fermi\_energy, temperature)

k\_B = 8.617333262145e-5;

f = 1 / (exp((energy - fermi\_energy) / (k\_B \* temperature)) + 1);

end

function conductivity=

calculate\_conductivity(eigenvalues, eigenfunctions, fermi\_energy, temperature)

conductivity = 0.0;

e = 1.602176634e-19;

h = 6.62607015e-34;

num\_states = length(eigenvalues);

for i = 1:num\_states

for j = 1:num\_states

f\_i = fermi\_dirac(eigenvalues(i), fermi\_energy, temperature);

f\_j = fermi\_dirac(eigenvalues(j), fermi\_energy, temperature);

matrix\_element\_squared = abs(eigenfunctions(:, i)' \* eigenfunctions(:, j))^2;

conductivity = conductivity + (e^2 / h) \* abs(f\_i - f\_j)^2 \* matrix\_element\_squared;

 end

    end

end

**CONDUCTIVITY VS TEMPERATURE MATLAB CODE**

% Define the data points

T = [300.45 317.15 322.15 327.15 332.15 337.15 342.15 347.15 352.15 357.15]; % Temperature in K

C = [45756662.93 42955234.58 42096129.89 41270715.58 40477047.97 40091552.28 38977898.05 38269208.99 37924441.34 37585830.26]; % Conductivity in S/m

% Plot the data points with a line connecting them

figure

plot(T, C, '-o') % Use '-o' to connect with lines and plot data points as circles

xlabel('Temperature (K)')

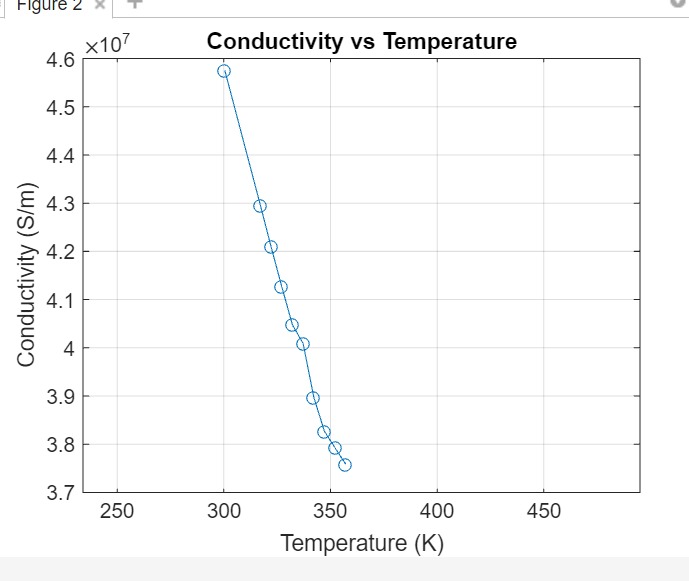
ylabel('Conductivity (S/m)')

title('Conductivity vs Temperature')

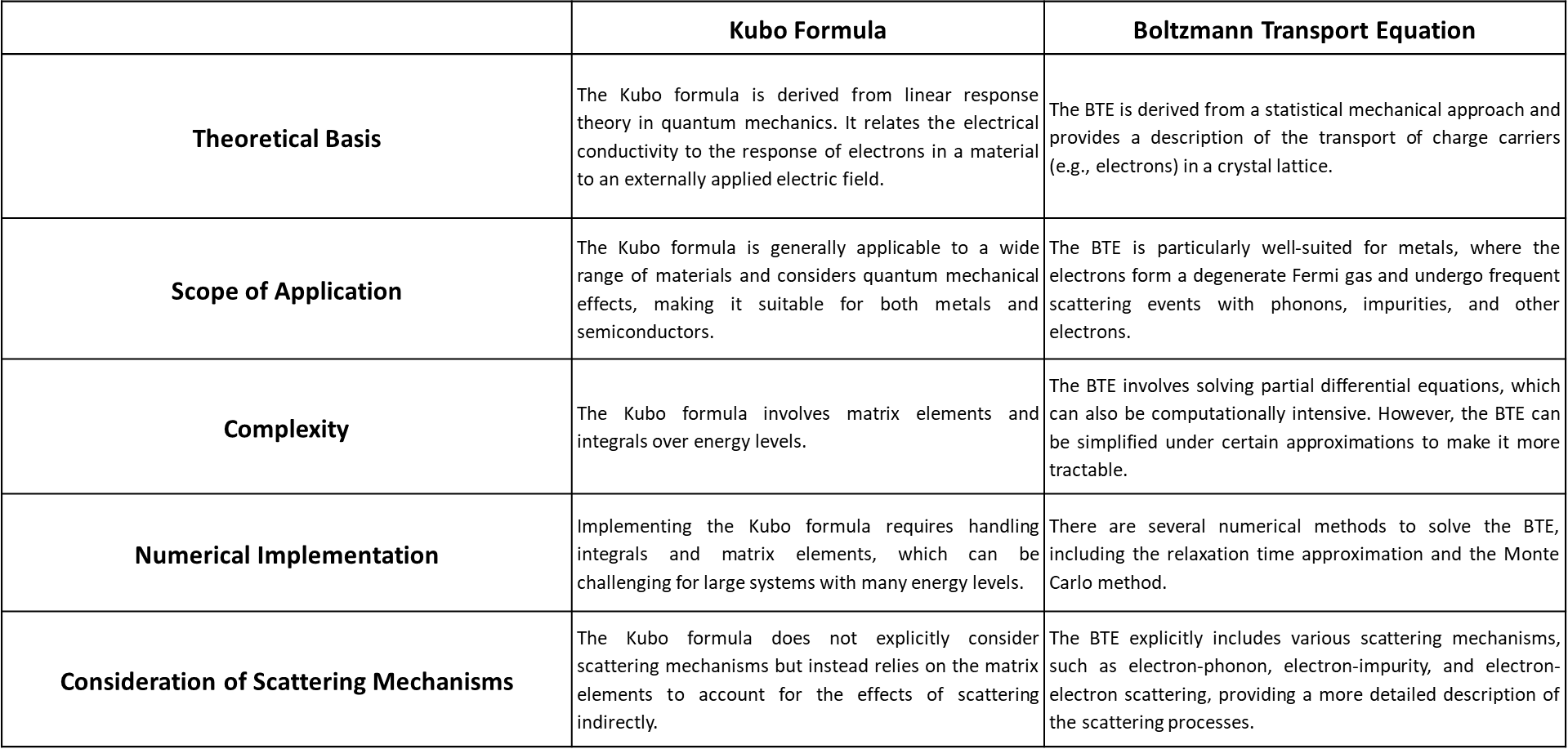
grid on

% To plot from 0 K, you can change the x-axis limits

xlim([0 max(T)])



**DIFFERNCE BETWEEN KUBO FORMULA AND BOLTZMANN TRANSPORT EQUATION**

****

**REFERENCES:**

**[1]** "Theory of the electrical properties of metals" by J. M. Ziman, Reports on Progress in Physics, 1960. DOI: 10.1088/0034-4885/23/1/301.

**[2]** "Transport theory and the Boltzmann equation in the semiclassical approximation" by G. Mahan, Applied Physics Letters, 1967. DOI: 10.1063/1.1754819.

**[3]** "Electronic transport in two-dimensional graphene" by K. S. Novoselov et al., Nature, 2005. DOI: 10.1038/nature04233

**[4]** "Electrical conductivity and Bloch-Boltzmann equation in graphene" by V. I. Fal'ko, European Physical Journal B, 2007. DOI: 10.1140/epjb/e2007-00004.

**[5]** "Electron-phonon interactions and the electrical resistivity of metals" by W. G.Aulbur et al., Journal of Physics: Condensed Matter, 2000. DOI: 10.1088/09538984/12/42/203.

**[6]** "Calculations of electrical resistivity and electrical conductivity via the Kubo method" by M. H. Ko, American Journal of Physics, 1992. DOI: 10.1119/1.17089.

**[7]** "An introduction to the Green-Kubo formula for classical molecular dynamics simulation" by R. Zwanzig, Journal of Statistical Physics, 2001. DOI:10.1023/A:1013690416641.

**[8]** "Comprehensive Semiconductor Science and Technology" by Adibi, A. Agarwal, A. Ahn, Elsevier Science, 2011, ISBN

978-0-444-53153-7.

**[9]** "Analytical Mechanics", L.N. Hand, J.D. Finch, Cambridge University Press, 2008, ISBN 978-0-521-57572-0.

**[10]** "ac conductivity of a one-dimensional site-disordered lattice", R. C. Albcrs, J. E. Gubcrnatis, Theoretical Division, Los Alamos Scientific Laboratory, Los Alamos, Mexico 87545, Physical Review B Volume 17 Number 12, 1978.

**[11]** "Transport properties of random and nonrandom substitutionally disordered alloys. I. Exact numerical calculation of the ac conductivity", M. Hwang, A. Gonis, and A. J. Freeman, Department of Physics and Astronomy and Materials Research Center, Northtvestern University, Evanston, Illinois 60201, Physical Review B Volume 35 Number 17, 1987.