Some ideas for the final project in ECE597/697EN

1. Drunken Landauer

We talked in class about the Landauer approach where a nanoscale system is represented by some collection of channels, each with a given transmission probability. Each channel has associated with it an energy $E$ which is the energy a carrier has to have to fill that channel, and the probability of filling a channel by inserting a carrier in it is given by the distribution function. Then we extended the concept of transmission to have more than one scattering point by imagining each channel is really now a sequence of channels each with its own transmission probability $T_{12}$, $T_{23}$, $T_{34}$, etc.

To simulate such a system, we would treat each channel as a series of steps: a carrier enters on either side and then goes down the channel to the scattering center. At that point it has probability $T_{12}$ to proceed or $R_{12}$ to reflect back. Next it goes to the second scattering center and again has probability of moving forward $T_{23}$ and probability of going back $R_{23}$. We are treating each channel now as a sequence of discrete steps “$i$”, each with a probability of moving one step forward given by $T_{i,i+1}$ and probability of moving one step back given by $R_{i,i+1}$. A sequence of $N$ such scattering centers can produce anywhere between $N$ and infinity steps.

This random process is sometimes jokingly likened to a drunk man walking down the street and making random turns at each corner because he’s too drunk to know the correct way home. The simulation you would perform would be a 1-dimensional random walk. Each time you have to decide whether to move forward or back, you would ask the computer to generate a random number “$R$” uniformly distributed between 0 and 1 (this is something computers do well and have built in routines for, just call rand in Matlab!). Then you compare the random number “$R$” with your probability $T_{i,i+1}$ and if $R < T_{i,i+1}$ then the move is accepted; otherwise you turn around and go back a step. After repeating this process many times for some large number $P_{in}$ of carriers, you would count how many actually came out the other end $P_{out}$ and the actual total transmission would be $P_{out}/P_{in}$. To get the correct carrier statistics, you would have some large number of channels $C$ each with energy $E$, and you would pick the channel into which to put the carrier according to $f(E,T)$, where $f(E,T)$ is either the Fermi-Dirac or the Boltzmann distribution function. Then averaging over all the channels would produce the correct energy distribution of carriers. Inserting carriers from two sides at different temperatures would allow you to count the net transfer and deduce a charge or heat current through the nanoscale system.

2. Here comes Boltzmann again…and again…and again…

The idea here is to solve the Boltzmann Transport equation in a 1D system numerically using direct discretization. What you would do is consider a 1D problem where we have only one real space coordinate ($x$) and one momentum space coordinate ($k$), and possibly time. Therefore we need a 2D array to store the values of the distribution function at each time. Let’s say we pick $Nx$
points in our discretization in space and Nk in momentum. Then we need a Nx by Nk matrix to store f(x,k,t) at each time. We would use simple finite differences to get the derivative of f(x,k,t) with respect to position df/dx, momentum df/dk and time df/dt. Each new f(x,k,t+dt) would be given by dt*(df/dx*v+df/dk*F+df_scat). Here df_scat is the change in the distribution due to scattering. The velocity v can be calculated from the dispersion relationship as \( v = \frac{dE}{dk} \cdot \frac{1}{h_{\text{bar}}} \) for electrons or \( v = \frac{d\omega}{dk} \) for phonons, depending on what you wish to simulate. The scattering can be approximated in the so called “relaxation time” approximation as \( df_{\text{scat}} = \frac{(f(x,k,t) - f_0)}{\tau} \) where \( \tau \) is the relaxation time accounting for all the scattering mechanisms and \( f_0 \) is the equilibrium distribution function at a given temperature or Fermi level, depending on whether we want electrons or phonons. The system is then iterated through many small time steps \( dt \) until it reaches steady state.

3. Converting to the Nanoscale

In lectures we gave definitions for the electrical and thermal conductivity, as well as the thermoelectric Seebeck coefficient \( S \) in terms of integrals over the velocity and density of states (DOS) by defining transport integrals \( L_{11}, L_{12}, L_{21}, \) and \( L_{22} \). A nice project would be to calculate those values for some nanostructure of interest, such as graphene or a silicon nanowire, using the dispersion \( E(k) \), velocity, and DOS we calculated in class or the homeworks. The only missing term in the equations is the lifetime \( \tau(E) \) which can be approximated in various ways—it is sometimes taken to be a constant, more often a power law \( \tau(E) = \tau_0 * E^n \) with power “n” or, more realistically, it follows the shape of the DOS \( \tau(E) = \tau_0 * g(E) \). Using these known components, and numerical integration over energy, you can compute the Seebeck coefficient, electrical, and thermal conductivity from the transport integrals. Finally, you can ask how efficient would the conversion from heat to electric power be by computing the thermoelectric figure of merit \( ZT = S^2 \sigma T / \kappa \) for your nanostructure of choice. You can also vary the doping or the Fermi level to see how charge density affects your result. A typical question you can ask is “what is the optimal doping level which maximizes \( ZT \)?” Calculating the Lorentz number \( L = ZT = \kappa / \sigma T \) for various nanostructures and assumptions for the relaxation time could also be interesting, especially in the ultimate limit where the thermal conductivity only has the electron contribution.

4. More than just Neighbors

Another classic project is to extend the nearest neighbor model of phonons to 3D and to multiple neighbors. Using a 3D version of the techniques we used in class, compute the phonon vibrational frequency for a realistic 3D or 2D solid, such as silicon or graphene, by introducing springs with constants \( K_1 \) between nearest neighbors, \( K_2 \) to second nearest neighbors, \( K_3 \) for 3\textsuperscript{rd}, etc. The number of neighbors usually does not have to be large, typically 4 is used. Note that this approach, called the force constant approach, is actually often used for graphene simply by fitting the spring constants to available experimental data.
Once the dispersion is obtained, you can compute the phonon DOS and from it you can get heat capacity by integrating over vibrational frequency. Note that the size of the problem is not very big---this problem is solved by creating a dynamical matrix which is, for a structure with 2 atoms per basis and 6 vibrational modes, a 6x6 matrix. The vibrational frequencies are then given by the eigenvalues of the dynamical matrix, which are easily computed in Matlab by calling eig(M) on your 6x6 matrix. To get the full dispersion, you have to compute the dynamical matrix and the vibrational frequencies for a large number of wavevectors sampled throughout the 1st Brillouin zone. The code to compute the DOS can be provided to you, or you can get a code to construct the dynamical matrix, and you use it as input to compute the DOS and heat capacity.

5. Gambling on Science

A popular approach to solving the Boltzmann transport equation is called the Monte Carlo method. At the heart of it is the idea that electrons can be approximated to be charged point particles. Then each electron is assigned a “momentum” \( k \), \( a \) and a position \( r \). The energy of each electron is given by \( E(k) \) and velocity is given by the group velocity so if an electron starts out at position \( r \) at time \( t \), then at \( t+dt \) it will have position \( r(t+dt)=r(t)+v(k)*dt \). If there is a force acting on the electron (such as an applied field) then the momentum \( k \) will change according to \( k(t+dt)=k(t)+F*dt/\hbar \). Electrons also scatter at certain time intervals which is done by treating the lifetime as a Poisson process with mean \( \tau(k) \) where \( \tau(k) \) is the lifetime (average time between scattering). Therefore the time to next scattering can be obtained from \( dt_{\text{scat}}=-\tau*\ln(R) \), where \( R \) is a random number drawn uniformly form the interval \([0,1]\) which is something that computers are very good at.

The power of the Monte Carlo approach is that many physical effects can be added to the mix. For example, the external potential can be applied and solved for from the Poisson equation to model semiconductor devices. Various forms of scattering can be put in if we know their respective scattering rates \( \tau(k) \), and boundaries or contacts can be introduced. For the project, though, a simple implementation with just a fixed applied field would do. You can compute useful things like mobility by applying a small electric field and then computing the current density. The ratio of current density and field gives conductance \( \sigma=en\mu \). The approach works in any dimension (1d, 2d, or 3d) as long as you know \( E(k) \) and \( \tau(k) \).

6. Thermal conductivity of semiconductors

Compute the thermal conductivity of a semiconductor using the Debye model (linear dispersion, quadratic DOS) for a semiconductor of choice (what will change is the velocity of sound). The relaxation time is given in the textbook as a sum of three components: isotope, phonon-phonon, and boundary scattering \( \tau(\omega)=A\omega^2+B\omega^3T\exp(-\Theta/3T)+\nu/L \), where A and B are given parameters and L is the size. Compare the results to measured data and see how close you get.