

# PHY480: Project 3 - Due Friday May 1

## Two problems in sparse linear algebra

Solving large sets of linear equations and finding eigenvalues and eigenvectors of matrices are two standard problems arising in all branches of physics and engineering. Often we are interested in large problems involving millions of variables. In that case it is important to take advantage of the sparse property of many matrices arising in science and engineering. This project involves a study of two problems where sparse matrix methods can be used. (i) The first problem is finding the ground state energy of a lattice problem in quantum physics. (ii) The second problem is to solve for the average conductance of a square array of random resistors.

### (i) Lowest energy state of a tight binding Hamiltonian

In this project you will calculate the lowest energy state of an electron in a square lattice of atoms, where there is one energy level available at each lattice site. These energy levels are  $\epsilon_i$  and we are going to consider the case where  $\epsilon_i$  may all be different. We use a tight binding model where we consider the hopping of electrons from one localized orbital to another. The hopping matrix element is given by,  $t_{ij}$ .

We consider a model where there is one orbital per site and where the electrons are not interacting. In that case, we can find the energy levels of the  $L \times L$  square lattice of atoms by considering the tight-binding Hamiltonian,

$$H = \sum_{\langle ij \rangle} t_{ij} |i\rangle\langle j| + \sum_i \epsilon_i |i\rangle\langle i| \quad (1)$$

This Hamiltonian can be used to find the band structure of materials. We are going to consider a different issue, namely the effect of disorder on the lowest energy state of this Hamiltonian. The problem which we consider arises when there are different values of  $\epsilon_i$  in a system. If these values are very different, the electron gets trapped or "localized" at one site. If the energies are the same, the electron moves in an "extended state".

#### What to do:

(i) Write a fortran code to set up the single electron, single orbital, tight binding Hamiltonian matrix for arbitrary  $N = L \times L$  lattices. Consider a fixed value of  $t_{ij} = 1$ , and site energies  $\epsilon_i$  which are random on the interval  $[-W/2 - 1, W/2 - 1]$ .

(ii) Write a fortran program to find the lowest energy eigenvalue  $E_0$  and wavefunction  $\psi_0$  using the iterative power method (see below).

(iii) Plot the value of the lowest energy eigenvalue  $E_0/N$  as a function of  $W$  for  $0 \leq W \leq 100$  and  $L = 50$ .

(iv) Plot the value of  $\sum_i^N (\psi_0^*(i) \psi_0(i))^2$  as a function of  $W$  for  $0 \leq W \leq 100$  and  $L = 50$ . Why is this quantity interesting?

#### Iterative power method

The iterative power method uses some very useful properties of matrices. The first property is that the eigenvalues of a real symmetric matrix are real, and the second is that the eigenfunctions of a real symmetric matrix are orthogonal so we can do an expansion of any vector using the eigenfunctions as a complete basis set. Consider a starting vector (which you choose, e.g. to be random), then

$$|start\rangle = \sum_{l=1}^N a_l |l\rangle \quad (2)$$

where  $a_l = \langle l | start \rangle$  and  $|l\rangle$  is the  $l^{th}$  eigenvector of the real symmetric matrix. Now if we repeatedly act on the starting vector  $|start\rangle$  using the tight binding Hamiltonian (matrix), we get,

$$H^n |start\rangle = \sum_{l=1}^N H^n a_l |l\rangle = \sum_{l=1}^N a_l E_l^n |l\rangle \quad (3)$$

Now if we make sure that the negative eigenvalues are larger in magnitude than the large positive eigenvalues and then write,

$$\sum_{l=1}^N a_l E_l^n |l\rangle = E_0^n (a_0 |0\rangle + a_1 \left(\frac{E_1}{E_0}\right)^n |1\rangle + \dots), \quad (4)$$

then we see that the second term in this expression goes to zero rapidly with the number of iterations  $n$ . The process of acting on an arbitrary state with the Hamiltonian thus "projects out" the ground state of the system. Actually this procedure can also be extended to extract several low lying levels.

**Procedure:**

Find the matrix  $H$ , and define a starting vector  $|start\rangle$  of length  $N$ . Define  $|old\rangle = |start\rangle / \langle start|start\rangle$

Then

- (i) Take the matrix product  $|new\rangle = H|old\rangle$
- (ii) Calculate  $norm = (\langle new|new\rangle)^{1/2}$
- (iii) The current estimate of  $E_0 = norm$
- (iv)  $|old\rangle = |new\rangle / norm$ , the current estimate of  $\psi_0 = |old\rangle$ .
- (v) if converged, exit otherwise return to (i)

**(ii) The resistance of a square array of random resistors**

Random resistor networks are used to model many different situations, ranging from transport in semiconductors to composites and pore networks in geology. Many of these cases have a non-linear response, however even in those cases the numerical methods on the solution depends on solving the linear case repeatedly. Here we concentrate on the linear case where each bond in a square lattice is a resistor that may have a different value from all of the other resistors in the network.

There are many ways to solve the random resistor network. Here we use an iterative method that takes advantage of the sparseness of the linear equations for the voltages in the network. To find the resistance of the network we first have to decide on the boundary conditions and on the way in which we want to set up the random resistor network. Consider placing the network with one of the bonds lying along the x-direction. In that direction, we take the boundary conditions one side of the network to be one volt and on the other side to be zero (grounded). In the y direction we take free boundaries. We consider the network to be made up of resistors of two types, with resistances  $r_1$  and  $r_2$  with a fraction  $p$  of type  $r_1$  and the remaining fraction  $1 - p$  of type  $r_2$ . The resistors are randomly placed with these probabilities on the bonds of the network. Now we want to find the total current in the network and from it we can find the resistance of the network from  $1\text{volt}/\text{current}$ .

To find the currents in the network we write down Kirchoff's laws for the node voltages and solve the large set of linear equations for these voltages. Once we have the node voltages it is easy to find the bond currents and hence the total current flowing through the network. Write a Fortran 90 program to set up the resistor network. Use the Conjugate Gradient method to solve the equations to find the voltage. Look up the Wikipedia page for the method - it has a good description. Your program should allow the possibility of different size networks (i.e.  $L \times L$  square networks). The CG method should be in a separate subroutine that takes as input the matrix, the starting vector, the boundary vector and outputs the solution vector.

Calculate the resistance of the network as a function of  $p$  for  $r_1/r_2 = 2, 20, 200$  and for system sizes  $L = 10, 20, 30$ . What is happening near  $p = 1/2$ ?