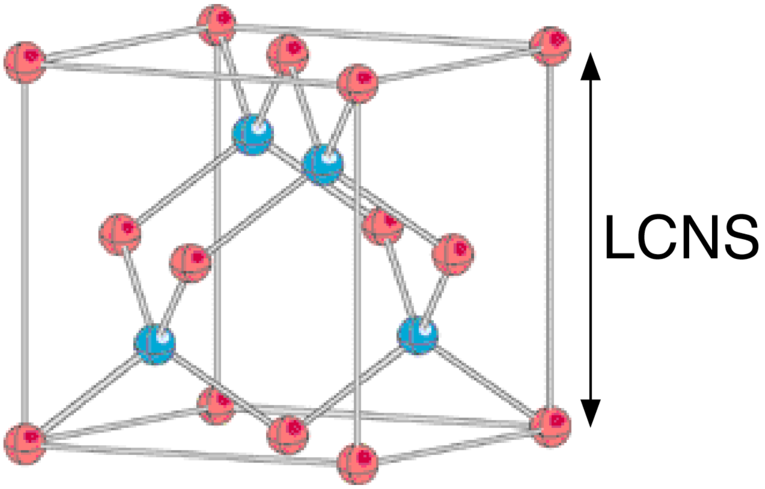
PHYS516 ASSIGNMENT 6
TIGHT BINDING MODEL OF ELECTRONIC STRUCTURES
Due: Friday, March 24, 2023

In this assignment, you will calculate the electronic structure of a small crystal of silicon atoms, based on the tight binding model explained in the lecture.

To set up atomic coordinates in a Si crystal, use the function `InitConf()` in the program `tb_util_angstrom.c` or `tb_util_au.c` at the class homepage (either use angstrom or atomic unit of your choice to specify length). This function reads the numbers of crystalline unit cells, `InitUcell[0]`, `InitUcell[1]` and `InitUcell[2]`, to be repeated in the *x*, *y* and *z* directions, respectively. Since the cubic unit cell of Si crystal (which takes the diamond structure, see the figure below) contains 8 atoms, your crystal will consist of `nAtom` = 8×`InitUcell[0]`×`InitUcell[1]`×`InitUcell[2]` atoms. In the bulk Si crystal, the lattice constant (the edge length of the cubic unit cell) is LCNS = 5.43 Å = 10.2622 atomic unit.



The following lists relevant data structures.

```
int nAtom;                 // # of atoms
double r[NMAX][3];         // r[i][0|1|2] is the x|y|z coordinate of atom i
```

where NMAX is the maximum number of atoms that can be handled with this program.

1. Write a program that sets up a 4`nAtom` × 4`nAtom` Hamiltonian matrix (using the s-p basis set explained in the lecture, 4 orbitals/atom × `nAtom` atoms) and diagonalizes it to obtain 4`nAtom` eigenenergies. (Use the periodic boundary condition, as implemented in the molecular dynamics program, `md.c`, to mimic a bulk crystal.)

   To use matrix diagonalization functions in *Numerical Recipes* (given in `eigen.c` at the class homepage), you need to define a matrix and vector as follows:

   ```
   double **h;             // Hamiltonian matrix
   double *d;              // Eigenvalues
   double *e;              // Work array for matrix diagonalization
   ```

   Then the following code section uses the *Numerical Recipes* utility functions, `dmatrix` and `dvector`, to allocate memory to use matrix, `h[i][j]`, in the range 1 ≤ `i` ≤ `n4` and 1 ≤ `j` ≤ `n4`, and vectors, `d[i]` and `e[i]`, in the range 1 ≤ `i` ≤ `n4`, where `n4` = 4×`nAtom` is the size of the basis set.

```
n4 = 4*nAtom;
h = dmatrix(1,n4,1,n4);
d = dvector(1,n4);
e = dvector(1,n4);
```

After setting up the `n4`×`n4` Hamiltonian matrix, you will diagonalize it with the *Numerical Recipes* functions, `tred2` (which reduces the symmetric Hamiltonian matrix into a tridiagonal form) and `tqli` (which diagonalizes the tridiagonal matrix).

```
tred2(h,n4,d,e);
tqli(d,e,n4,h);
```

2. Choose `InitUcell[0]` = `InitUcell[1]` = `InitUcell[2]` = 1 and diagonalize the resulting 32×32 Hamiltonian matrix for three different values of lattice constant: `LCNS` = 1.8×5.43 Å, 1.4×5.43 Å, and 1×5.43 Å. For each lattice constant, plot the density of states defined as follows:

$$D(\varepsilon) = \sum_{\nu=1}^{n4} \frac{1}{\sqrt{\pi}\sigma} \exp\left(-\frac{(\varepsilon-\varepsilon_\nu)^2}{\sigma^2}\right)$$

where σ = 0.1 eV = 0.003675 au is the energy spread given to each energy eigenvalue, ε_ν, to obtain a smooth density-of-states curve. Plot $D(\varepsilon)$ *vs.* ε and discuss the effect of lattice constant on the density of states.

3. Choose `InitUcell[0]` = `InitUcell[1]` = `InitUcell[2]` = 2 (`nAtom` = 8×2^3 = 64), diagonalize the resulting 256×256 Hamiltonian matrix for `LCNS` = 5.43 Å, and plot the density-of-states $D(\varepsilon)$ *vs.* ε. How does this density of states differ from the one with the same lattice constant calculated above for 8 atoms?

4. Consider the eigenenergies for 64 atoms and the bulk Si lattice constant, `LCNS` = 5.43 Å. Determine the Fermi distribution,

$$f(\varepsilon_\nu) = \frac{2}{\exp\left((\varepsilon_\nu-\mu)/k_\mathrm{B}T\right)+1},$$

for all the 256 eigenenergies, where T = 0.2 eV/k_B (k_B is the Boltzmann constant) = 0.2×11604.5 K = 2320.9 K is the temperature and μ is the chemical potential to satisfy

$$\sum_\nu f(\varepsilon_\nu) = 4N = 256.$$

(Note that there are 4 valence electrons from each of the 64 Si atoms). Plot $f(\varepsilon_\nu)$ *vs.* ε_ν.

Submit your code (#1 above) and all plots (three plots for #2, one plot for #3, and one for #4).