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 the atomic coordinates in a Si cluster, use the function `InitConf()` in the program `tb_util.c` at the class homepage. 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 $n_{\text{Atom}} = 8 \times \text{InitUcell[0]} \times \text{InitUcell[1]} \times \text{InitUcell[2]}$ atoms. In the bulk Si crystal, the lattice constant (the edge length of the cubic unit cell) is $L_{\text{CNS}} = 5.43 \text{ Å} = 10.2622$ atomic unit.

![Si crystal structure](image)

The following lists relevant data structures.

```c
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 $4n_{\text{Atom}} \times 4n_{\text{Atom}}$ Hamiltonian matrix (using the s-p basis set explained in the lecture, 4 orbitals/atom $\times n_{\text{Atom}}$ atoms) and diagonalizes it to obtain $4n_{\text{Atom}}$ 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:

```c
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 $1 \leq i \leq n_4$ and $1 \leq j \leq n_4$, where $n_4 = 4n_{\text{Atom}}$. 
and vectors, \( d[i] \) and \( e[i] \), in the range \( 1 \leq i \leq n_4 \), where \( n_4 = 4 \times n_{\text{Atom}} \) is the size of the basis set.

\[
n_4 = 4 \times n_{\text{Atom}}; \\
h = \text{dmatrix}(1,n_4,1,n_4); \\
d = \text{dvector}(1,n_4); \\
e = \text{dvector}(1,n_4);
\]

After setting up the \( n_4 \times n_4 \) Hamiltonian matrix, you will diagonalize it with the **Numerical Recipes** functions, \texttt{tred2} (which reduces the symmetric Hamiltonian matrix into a tridiagonal form) and \texttt{tqli} (which diagonalizes the tridiagonal matrix).

\[
\text{tred2}(h,n_4,d,e);
\text{tqli}(d,e,n_4,h);
\]

2. Choose \( \text{InitUcell}[0] = \text{InitUcell}[1] = \text{InitUcell}[2] = 1 \) and diagonalize the resulting 32\( \times 32 \) Hamiltonian matrix for three different values of lattice constant: \( \text{LCNS} = 1.8 \times 10.2622 \) au, \( 1.4 \times 10.2622 \) au, and \( 1 \times 10.2622 \) au. For each lattice constant, plot the density of states defined as follows:

\[
D(\epsilon) = \sum_{\nu=1}^{n_4} \frac{1}{\sqrt{\pi} \sigma} \exp\left( -\frac{(\epsilon - \epsilon_\nu)^2}{\sigma^2} \right)
\]

where \( \sigma = 0.1 \) eV = 0.003675 au is the energy spread given to each energy eigenvalue, \( \epsilon_\nu \), to obtain a smooth density-of-states curve. Plot \( D(\epsilon) \) vs. \( \epsilon \) and discuss the effect of lattice constant on the density of states.

3. Choose \( \text{InitUcell}[0] = \text{InitUcell}[1] = \text{InitUcell}[2] = 2 \) (\( n_{\text{Atom}} = 8 \times 2^3 = 64 \)), diagonalize the resulting 256\( \times 256 \) Hamiltonian matrix for \( \text{LCNS} = 10.2622 \) au, and plot the density-of-states \( D(\epsilon) \) vs. \( \epsilon \). 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, \( \text{LCNS} = 1 \times 10.2622 \) au. Determine the Fermi distribution,

\[
f(\epsilon_\nu) = \frac{2}{\exp\left( \frac{(\epsilon_\nu - \mu)}{k_B T} \right) + 1},
\]

for all the 256 eigenenergies, where \( T = 0.2 \) eV/\( k_B \) (\( k_B \) is the Boltzmann constant) = 0.2\times11604.5 K = 2320.9 K is the temperature and \( \mu \) is the chemical potential to satisfy

\[
\sum_\nu f(\epsilon_\nu) = 4N = 256.
\]

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

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