

# Modeling and simulation methods in nanotechnology.

## Practical laboratory

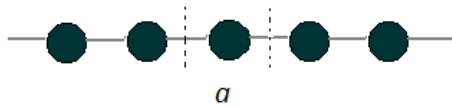
Solving practical tasks in computer laboratory. 6 hours/topic

### Practical application of tight-binding method to numerical calculations of energy bands in selected graphene structures.

#### Task 1.

#### One-dimensional chain of atoms. Construction of molecular orbital and Hamiltonian matrix.

This is arithmetical task. Let us consider a system as show in the figure:



Atoms are represented by Black points and chemical bonds by lines. The lattice constant is assigned as  $a$ . We assume one (valence) orbital  $\chi$  per atom. Although  $\chi$  can depend on three special variables, symbolically we write the dependence only on  $x$ , since this is the direction of periodicity. We assume  $N$  atoms in this chain.  $\chi(x-na)$  is an orbital situated on  $n$ -th atom in the chain. We create a molecular (crystal) orbital fulfilling the Bloch theorem, as

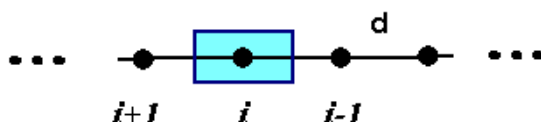
$$\phi(\mathbf{r}) = \frac{1}{\sqrt{N}} \sum_n e^{ikna} \chi(x-na)$$

The energy of electron in such a chain (precisely the energy expectation value in a state described by function  $\phi$ , i.e. matrix element of  $\mathbf{H}$  is

$$E = \langle \phi | \mathbf{H} | \phi \rangle$$

Since each  $\phi$  is a sum of many  $\chi$ , therefore, applying the nearest neighbors approximation, i.e.

assuming that different than zero are only those  $\langle \chi_i | \mathbf{H} | \chi_j \rangle$  elements, where  $j=i$  and when  $j$  differs from  $i$  by only one position (the nearest neighbor).



Next, putting  $\langle \chi_i | \mathbf{H} | \chi_i \rangle = 0$  (i.e. the energy scaling) we finally get

$$E = \langle \chi_0 | \mathbf{H} | \chi_1 \rangle e^{ikd} + \langle \chi_0 | \mathbf{H} | \chi_{-1} \rangle e^{-ikd} = t2 \cos(kd) \quad (1)$$

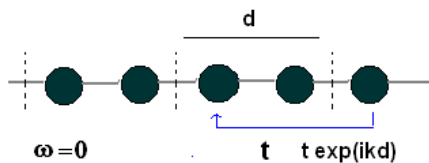
that means a single band  $E(k)$ .

Students have to derive the formula (1) and draw  $E(k)$  in any graphical tool.

### **Task 2.**

#### One-dimensional chain as in task.1, but now unit cell contains two atoms.

The aims of this exercise are: (1) derive formula for Hamiltonian matrix elements, (2) understanding that the resulting bands, come out are energetically equivalent to the band obtained in task.1, (3) to show that the bands can be obtained by folding BZ from the task.1. Now, we have the same chain but the unit cell is defined in a different way.



Now, the dimension of Hamiltonian matrix is 2 (two atoms in a unit cell). Setting diagonal elements equal zero we get

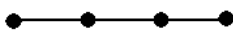
$$H = \begin{bmatrix} 0 & t(1 + e^{ikd}) \\ t(1 + e^{-ikd}) & 0 \end{bmatrix}$$

The energy spectrum is the same as in task.1, although now built of two bands. It is important to notice the difference in the size of BZ in both cases. Matrix can be diagonalized "by hand" or one can write a code, which will construct  $H$  and call appropriate procedure (diagonalization).

### **Task 3.**

#### Construction of the Hamiltonian matrix for a few-atomic molecule.

For example, "molecule"  $C_4$

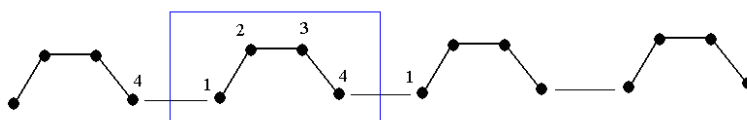


$H_{ii} = 0$ ,  $i=1,2,3,4$  - as diagonal elements,  $H_{12} = H_{21} = H_{23} = H_{32} = H_{34} = H_{43} = t$  [ = -2.7 eV ] All the others  $H_{kl} = 0$ ; here  $H$  doesn't depend on  $k$

### **Task4.**

#### Construction of the Hamiltonian matrix for the simplest (the most narrow) armchair graphene Ribbon.

First, one has to enumerate all the nodes (atoms) in the unit cell (also separately in the neighbor unit cells).

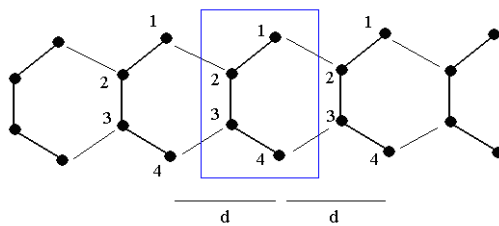


$$H = \begin{bmatrix} 0 & t & 0 & t(e^{-ikd}) \\ t & 0 & t & 0 \\ 0 & t & 0 & t \\ t(e^{ikd}) & 0 & t & 0 \end{bmatrix}$$

**Task 5.**

Construction of the Hamiltonian matrix for the simplest (the most narrow) zigzag graphene Ribbon.

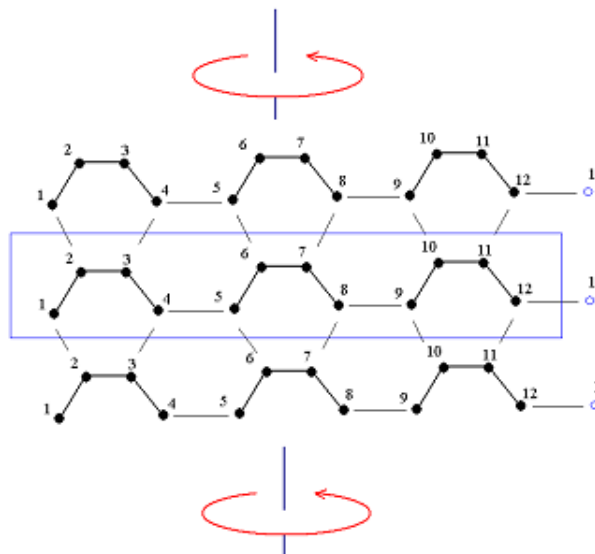
First, one has to enumerate all the nodes (atoms) in t hunt cell (also separately in the neighbor unit cells).



$$H = \begin{bmatrix} 0 & t(1 + e^{ikd}) & 0 & 0 \\ t(1 + e^{-ikd}) & 0 & t & 0 \\ 0 & t & 0 & t(1 + e^{-ikd}) \\ 0 & 0 & t(1 + e^{ikd}) & 0 \end{bmatrix}$$

**Task 6.**

Construction of the Hamiltonian matrix for carbon nanotube (3,3)



And matrix H (dimension = 12)

$$\begin{bmatrix}
 0 & t(1+e^{ikd}) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t \\
 t(1+e^{-ikd}) & 0 & t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & t & 0 & t(1+e^{-ikd}) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & t(1+e^{ikd}) & 0 & t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & t & 0 & t(1+e^{ikd}) & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & t(1+e^{-ikd}) & 0 & t & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & t & 0 & t(1+e^{-ikd}) & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & t(1+e^{ikd}) & 0 & t & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & t & & & & \\
 & & & & & & & & & & & \text{etc...}
 \end{bmatrix}$$

### **Final classification task.**

#### **Construction and execution of a computer code for calculating energy structure of any graphene systems, periodic and non-periodic ones.**

General structure of the code:

- Determination of the number of nodes (atoms) in the unit cell,  $N = \dim H$ ,
- Definition of bonds (connections) within a given unit cell; determination of elements, which are  $H_{ij} = t$ ,
- Definition of connections (bonds) between atoms belonging to neighbor unit cells, ( $t e^{ik}$ ), or  $t[1+e^{ik}]$ ),
- Loop over all  $k$  values (in BZ)
  - diagonalization of  $H$  for each  $k$ , using any appropriate method, depending on the programming language or package and platform chosen,
- Allow for different strengths of Bond ( $t$ ) in different regions,
- Allow for unit cell modification by adding or removing atoms,
- Write the results (energy bands) down to a file
- Make diagrams of the calculated energy bands.

Students can choose any language or computational package they wish (Fortran, C, C++, Csharp, Matlab, Python, etc..). They can utilize ready-to-use libraries and procedures for diagonalization of complex, hermitian matrices.

**For classification, students must demonstrate they know their codes in detail. They must execute the code for a system chosen by the teacher (e.g., carbon tube (6,6), carbon tube (5,0), graphene zigzag Ribbon with one edge modified by different number of Klein nodes).**

### **Resonance states in nanostructures**

Three exercise problems (Tasks 7-9) associated with resonances in nanostructures are given here. They should be done by single students within the exercise hours in a computer laboratory or at home. One of them should be chosen by the teacher for a group of students or for a given course. (A student is expected to do one of the exercises). The job is to prepare some analytical formulas, to encode them, to compute the results and to present them graphically. Students choose which programming language and what programming environments they want to use.

### **Task 7.**

#### **Resonance states of symmetric system of one well and two barriers**

The system consists of three layers of semiconductors. We consider one electron whose interaction with the structure is modeled by a one-dimensional potential of rectangular well and barriers:

$$V(x) = \begin{cases} 0, & x < -b \\ V_0, & -b \leq x \leq -a \\ 0, & |x| < a \\ V_0, & a \leq x \leq b \\ 0, & x \geq b \end{cases}$$

where  $V_0 > 0$ .

The Hamiltonian is:

$$\hat{H} = -\frac{1}{2m} \frac{d^2}{dx^2} + V(x).$$

We have to solve the Schroedinger equation for energies of the range between 0 and  $V_0$  in a way analogous to that described in the lecture chapter "Resonance states of the potential well-barrier system". The symmetry of the system should be used to simplify the problem. Since the system does not change under inversion of coordinate  $x \rightarrow -x$ , i.e. ,  $V(-x) = V(x)$ , the probability distribution also has to be even function of  $x$ , i.e.  $|\psi(-x)|^2 = |\psi(x)|^2$ . Functions themselves can be either even,  $\psi(-x) = \psi(x)$ , or odd,  $\psi(-x) = -\psi(x)$ . Eigenfunctions of these two types can be found separately. Thus we assume proper relations between coefficient present in the functions on the left and right sides. This reduces the number of unknowns and makes the problem easier.

The results should be presented in a graphical way. Especially the resonance character of solutions should be shown like in the lecture chapter "Resonance states of the potential well-barrier system". New ideas on how to present results are welcome.

The computer code should offer possibility of changing the parameters of the potential  $a, b, V_0$  and effective mass of the electron  $m$ .

### **Task 8.**

#### **Resonance states of a symmetric nanolayer-structure modeled with an analytical potential**

The nanostructure under consideration is similar to that of previous exercise but modeled with an analytical (continuous) potential

$$V(x) = (ax^2 - b)e^{-qx^2},$$

where  $a, b$  i  $q$  are nonnegative real parameters.

The Hamiltonian is

$$\hat{H} = -\frac{1}{2m} \frac{d^2}{dx^2} + V(x).$$

The exercise relies in application of the **stabilization method** to find resonance states in the energy range from the potential asymptote (energy level equal to  $\lim_{x \rightarrow \infty} V(x) = 0$ ) up to the maximum of the potential and bound states of energies between the minimum of the potential ( $-b$ ) and the asymptote.

The symmetry of the system ( $V(-x) = V(x)$ ) should be used; the even and odd solutions should be searched for separately. For the even states a basis of even functions

$$\varphi_i(x) = e^{-\beta_i x^2}, i = 1, \dots, N,$$

and for the odd states a basis of odd functions

$$\varphi_i(x) = x e^{-\beta_i x^2}, i = 1, \dots, N$$

should be used. In both cases the values of  $\beta_i$  parameters are chosen so that they form a geometric sequence. The first and the last values,  $\beta_1$  and  $\beta_N$  need to be given.

Formulas should be derived for the matrix elements:  $H_{ij}(\alpha) = \langle \varphi_i^\alpha | \hat{H} | \varphi_j^\alpha \rangle$  i  $S_{ij}(\alpha) = \langle \varphi_i^\alpha | \varphi_j^\alpha \rangle$ .

The following relations are worth paying attention to

$$S_{ij}(\alpha) = \langle \varphi_i^\alpha | \varphi_j^\alpha \rangle = \int_{-\infty}^{\infty} \varphi_i(\alpha x) \varphi_j(\alpha x) dx = \alpha^{-1} \int_{-\infty}^{\infty} \varphi_i(x) \varphi_j(x) dx = 2\alpha^{-1} \int_0^{\infty} \varphi_i(x) \varphi_j(x) dx$$

$$\begin{aligned} T_{ij}(\alpha) &= \left\langle \varphi_i^\alpha \left| -\frac{1}{2m} \frac{d^2}{dx^2} \right| \varphi_j^\alpha \right\rangle = -\frac{1}{2m} \int_{-\infty}^{\infty} \varphi_i(\alpha x) \frac{d^2}{dx^2} \varphi_j(\alpha x) dx = -\frac{\alpha}{2m} \int_{-\infty}^{\infty} \varphi_i(x) \frac{d^2}{dx^2} \varphi_j(x) dx \\ &= \frac{\alpha}{2m} \int_{-\infty}^{\infty} \left( \frac{d}{dx} \varphi_i(x) \right) \left( \frac{d}{dx} \varphi_j(x) \right) dx = \frac{\alpha}{m} \int_0^{\infty} \left( \frac{d}{dx} \varphi_i(x) \right) \left( \frac{d}{dx} \varphi_j(x) \right) dx \end{aligned}$$

( $T_{ij}(\alpha)$  is a matrix element of the kinetic part of the Hamiltonian)

$$S_{ij} = S_{ji}, H_{ij} = H_{ji}.$$

They should be used when deriving the formulas and then when coding them so that to avoid multiple computation of the same matrix elements. In particular, the matrices **S** and **T** may be computed only once and then when  $\alpha$  changes multiplied by appropriate factors.

For solving

$$\mathbf{H}(\alpha) \mathbf{C}^{\text{opt}}(\alpha) = E(\alpha) \mathbf{S}(\alpha) \mathbf{C}^{\text{opt}}(\alpha),$$

a standard procedure should be used which can be found in LAPACK library.

The input data should include the physical parameters of the potential  $a, b, q$  and the effective electron mass as well as data considering the method: the number of basis functions  $N$ , exponents  $\beta_1$  and  $\beta_N$ , and the number and range of the scaling parameter values.

The results should be presented in the form of a stabilization graph (Fig. 2.11 of the lecture materials) and histograms of the density of roots obtained in the vicinities of resonance levels together with the Lorentz curves fitted to the histograms data (Fig. 2.12). The final results: the resonance energy positions,  $E_r$ , and widths,  $\Gamma$ , should be given.

## **Task 9.**

### **Resonance states of spherically symmetric quantum dot**

A spherical quantum dot considered here is a ball made of a semiconductor surrounded by a spherically symmetric layer of another semiconductor and put into a third material extending to infinity. Such a nanostructure influences an electron put into it via spherically symmetric potential

$$V(r) = (ar - b)e^{-qr},$$

where  $a, b$  and  $q$  are nonnegative real parameters.

The issue, although in principle three-dimensional, can be reduced to one-dimensional radial problem (dependent on the radial coordinate  $r$  only) with Hamiltonian

$$\hat{H} = -\frac{1}{2m} \frac{d^2}{dr^2} + \frac{l(l+1)}{2mr^2} + V(r),$$

where  $l(l+1)$  is a value of angular momentum of the electron,  $l = 0, 1, 2 \dots$ .

The exercise relies in application of the complex coordinate rotation method to find resonance states of the electron in the quantum dot in the energy range between the asymptote of the potential (level 0) and the maximum of the potential.

Basis set of  $N$  functions should be chosen as

$$\varphi_i(r) = r^n e^{-\beta_i r},$$

where  $n$  is the same for all functions and parameters  $\beta_i$  form a geometric sequence defined by two extreme elements:  $\beta_1$  and  $\beta_N$ .

Before the computer code is written the formulas for the matrix elements

$$H_{ij}(\theta) = \langle \varphi_i | \hat{H}(e^{i\theta} r) | \varphi_j \rangle = \int_0^{\infty} \varphi_i(r) \hat{H}(e^{i\theta} r) \varphi_j(r) dr$$

and

$$S_{ij} = \langle \varphi_i | \varphi_j \rangle = \int_0^{\infty} \varphi_i(r) \varphi_j(r) dr$$

should be derived. Let us note that elements  $S_{ij}$  do not depend on  $\theta$ . Similarly the  $\theta$  dependence of the kinetic part of the Hamiltonian can be factorized

$$\begin{aligned}
T_{ij}(\theta) &= \left\langle \varphi_i \left| -\frac{1}{2m} \frac{d^2}{d(e^{i\theta}r)^2} \right| \varphi_j \right\rangle = -\frac{e^{-2i\theta}}{2m} \int_0^\infty \varphi_i(r) \frac{d^2}{dr^2} \varphi_j(r) dr = \\
&= \frac{e^{-2i\theta}}{2m} \int_0^\infty \left( \frac{d}{dr} \varphi_i(r) \right) \left( \frac{d}{dr} \varphi_j(r) \right) dr = e^{-2i\theta} T_{ij}(\theta = 0).
\end{aligned}$$

Computation of the elements  $S_{ij}$  and  $T_{ij}(\theta = 0)$  can be performed just once and then, when changing  $\theta$ , the elements  $T_{ij}(\theta = 0)$  need to be multiplied by factor  $e^{-2i\theta}$ . This cannot be done for the potential part of the Hamiltonian

$$V_{ij}(\theta) = \langle \varphi_i | V(e^{i\theta}r) | \varphi_j \rangle = \int_0^\infty \varphi_i(r) V(e^{i\theta}r) \varphi_j(r) dr.$$

All the integrals needed are of one type:

$$\int_0^\infty r^m e^{-\gamma r} dr = \frac{m!}{\gamma^{m+1}}.$$

The generalized eigenvalue problem

$$\mathbf{H}(\theta)\mathbf{C}^{\text{opt}}(\theta) = E(\theta)\mathbf{S}\mathbf{C}^{\text{opt}}(\theta)$$

should be solved by using a standard procedure which can be taken from LAPACK library.

The input data should include the physical parameters, i.e. the potential parameters  $a, b, q$ , the effective mass of the electron  $m$  and the number  $l$  determining the angular momentum value, and the defining the method: the number of basis functions  $N$ , their parameters  $n, \beta_1$  and  $\beta_N$ , and the range and number of values of the rotation parameter  $\theta$ .

The results should be presented graphically in the form of  $\theta$ -trajectories (see Fig. 42 of the lecture notes). The final result, i.e. the optimal approximate resonance energies, should be reported.

## Application of gradient descent method for calculation of strain in one-dimensional chain of atoms

### Task 10.

#### Gradient descent method application for a simple analytical function.

Write a computer program implementing gradient descent method for finding minimum of a two-dimensional function:  $f(x,y) = x^2 - 6x + 9 + 2y^2 - 8y + 8$ , for which the gradient is given by a simple analytical formula.



### **Task 11.**

#### Gradient descent method applied to N-dimensional function.

Write a computer program implementing gradient descent method for any N-dimensional ( $N \leq 10$ ) user defined function. Consider two cases:

- A. gradient given by an analytical formula
- B. gradient calculated using numerical approximation:

$$f = f(x_1, x_2, \dots, x_n) \quad \nabla f = \left[ \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right]$$
$$\frac{\partial f(x_1, x_2, \dots, x_n)}{\partial x_k} = \lim_{h \rightarrow 0} \frac{f(x_1, x_2, \dots, x_k + h, \dots, x_n) - f(x_1, x_2, \dots, x_k, \dots, x_n)}{h} \approx$$
$$\approx \frac{f(x_1, x_2, \dots, x_k + \Delta x, \dots, x_n) - f(x_1, x_2, \dots, x_k - \Delta x, \dots, x_n)}{2\Delta x}$$

### **Task12. Final classification task.**

#### Construction and execution of a computer code for minimizing strain energy in one-dimensional chain of atoms.

Apply a gradient descent method and write a computer program finding elastic energy minimum for a one-dimensional chain of N atoms:

$$E = \sum_{i=0}^{N-1} \sum_{\substack{j=i-1, i+1 \\ j \geq 0, j \leq N-1}} A_{ij} \left[ (x_i - x_j)^2 - d_{ij}^2 \right]^2,$$

where  $x_i$  are atomic positions.  $d_{ij}$  is a bond length between  $i$  and  $j$  atoms,  $A_{ij}$  are force constants related to bond squeezing or stretching.

The picture below shows the example input data for a chain, where atoms are coded by color (Red, Green, Blue). The equilibrium bond lengths and force constants are also given in the input. Just like in case of real semiconductor compounds (e.g. InAs) the bonds are only allowed between atoms of specific colors (kinds), i.e. B-G and B-R.

**Important notice: the edge atoms positions are not modified (are frozen) during the optimization process!!!**

$$d_{BR} \quad 1.5$$

$$A_{BR} \quad 2.0$$

$$d_{BG} \quad 1.75$$

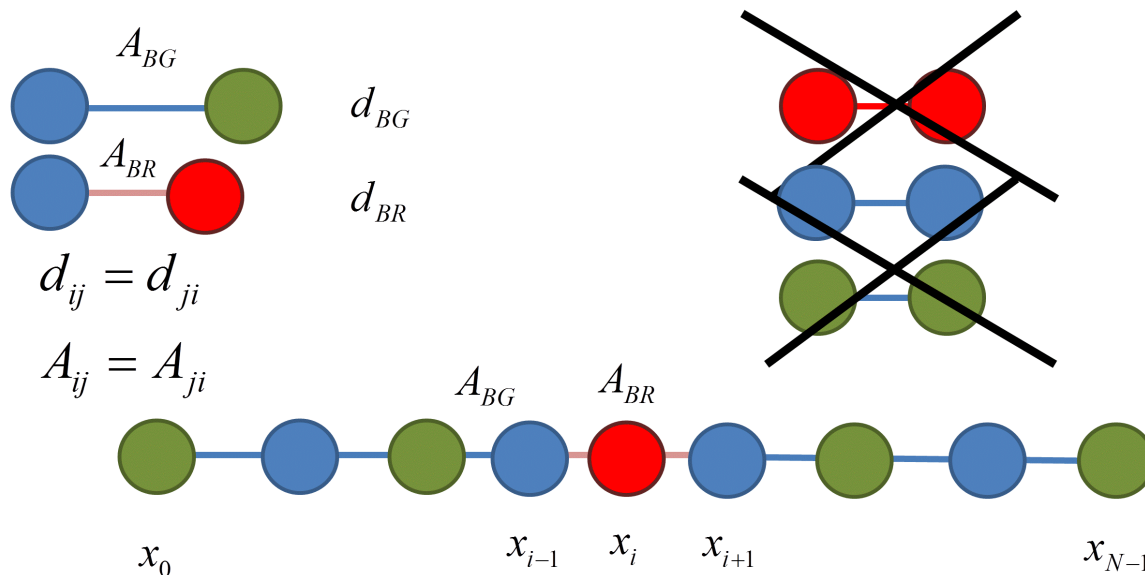
$$A_{BG} \quad 2.5$$

$$x_0 \quad 0$$

$$x_8 \quad 18.0$$

$$N \quad 9$$

$$G \quad B \quad G \quad B \quad R \quad B \quad G \quad B \quad G$$



Program output should include optimized atomic position and strain values for each chain atom:

$$\varepsilon(x_i) = \frac{(x_{i+1} - x_i) - d_{i,i+1} + (x_i - x_{i-1}) - d_{i,i-1}}{d_{i,i+1} + d_{i,i-1}} = \frac{(x_{i+1} - x_{i-1}) - (d_{i,i+1} + d_{i,i-1})}{d_{i,i+1} + d_{i,i-1}}$$

for  $i = 1, \dots, N - 2$

Program should also return total number of iterations, gradient length and elastic energy value calculated during each iteration.

Students can choose any language or computational package they wish (Fortran, C, C++, C#, Matlab, Python, etc.).

**For classification, students must demonstrate they know their codes in detail. They must execute the code for a system (input data) chosen by the teacher.**

## Modeling of electrical and mechanical properties of bionanosystems

### **Task 13.**

PDB i VMD – computer visualisation of nanosystems (2h)

1. Protein Data Bank database - basics.
  - 1.1. Open the WWW page of the Protein Data Bank and answer (in your report) the following questions:
    - 1.1.1 How many structures are available in the PDB today?
    - 1.1.2 How many structures of nanotubes may be found in the PDB?
    - 1.1.3 Enlist at least 3 living organisms for which 3-D structures of hemoglobin proteins are stored in the PDB. Register PDB codes of these hemoglobins.
    - 1.1.4 How many proteins are linked to the word: „biotechnology” in PDB?
    - 1.1.5 What molecule was nominated „A molecule of the month” in PDB as of the day of our exercises?

- 1.1.6 Download from the PDB (to a local folder) a file in a PDF format having 2AHJ structure (or any structure related to fibroin ).
2. VMD graphical environment – an introduction.
  - 2.1 Find in the Internet the Visual Molecular Dynamics code for Windows. Get acquainted with a manual of this code. Determine the current version number and hardware requirements.
  - 2.2 Run the VMD code (it should be installed on PKV computer). If you prefer to work with your own notebook, register to the VMD code as academic user and install it on your computer.
  - 2.3 Use the command „Load molecule” to load a protein stored before on a local HD (point 1.1.6). Learn how to navigate the molecule model using a computer mouse (R, T, S, keys, scroll). Measure a distance between selected atoms.
3. Protein visualization.
  - 3.1 Check at least 4 options for a protein representation and colouring available in the VMD.
  - 3.2 Learn how to select amino acids and display them in a selected colour.
  - 3.3 Select all tryptophanes (Trp) and display them in “red” using a licorice representation.
  - 3.4 Learn how to save a graphical representation (“save state” command). Learn how to load the saved state.
  - 3.5 Learn how to change the background colour in the Main display window.
4. Modeling of a nanotube using VMD.
  - 1.1 Use an Internet to review possible graphical representations of carbon nanotubes.
  - 1.2 Find in the Internet coordinates of any nanotube that may be loaded into VMD (or ArgusLab). Download these coordinates.
  - 1.3 Use VMD or ArgusLab (Pymol??) to create your own the most beautiful picture of a nanotube. The points will be awarded for the clarity, originality, and effectiveness in structural information display technique. Save this Picture and send it to your email mailbox. This Picture should be inserted into a report from this exercise (formats \*.doc , \*.docx or pdf are accepted).

*1.4 Hints:*

(a) Nanotube Quicktime movies: see.: <http://www.ipt.arc.nasa.gov/gallery.html>

(b) Nanotube generator, use notepad to copy data, write down the output with the name having extension \*.pdb,

Read-in this file into VMD or Arguslab code (version 1.0 is ok)

<http://www.ugr.es/~gmdm/java/contub/contub.html>

(c) Learn how to use a plugin to VMD - nanotube builder:

<http://www.ks.uiuc.edu/Research/vmd/plugins/nanotube/>

5. Your short report (1-2 pages) from this exercise - Task 13 (wn#1) should contain:

A. Answers to questions 1.1.1-1.1.5

B. A nice picture (prepared by yourself!) of a carbon nanotube (SW or MWNT) drawn using (preferably) the VMD code. You may pretend that this graphics will be used for a book cover. Give some information explaining what features of a nanotube are presented in this particular graphical model.

### **Task 14.**

#### Quantum modeling of drugs – structures and electrostatic potentials (2h)

1. Learning ArgusLab package.
  - 1.1 Dig out in the Internet the newest version of ArgusLab (academic licence is free). Instal the code on your computer (it has been already installed on PKV PCs)
  - 1.2 Learn „nuts-and-bolds” of pull-down Menu and „Builder” options. Learn how to measure interatomic distances and bond angles in a molecule.
  - 1.3 Built a model of water.
  - 1.4 Optimize the structure of water molecule. Use AM1, PM3, MNDO Hamiltonians. In a separate calculation find an optimal (the lowest energy) geometry using molecular mechanics („pliers icon”). Write down H-O-H angles and O-H distances into a table. Compare your results with experimental data (Wiki). Decide which computer method would you recommend for that type of calculations. Select visualization of orbitals HOMO and LUMO for water , for example, in the AM1 method. Select an option: calculate the electron density. Calculate an electrostatic potential (AM1) for H<sub>2</sub>O molecule. Project the electrostatic potential on that surface. Note how positive and negative regions of MEP are graphically represented. Do you see that water molecule has a dipole moment?
2. Building a model of a drug molecule.
  - 2.1 Find a structural formula of Aspirin (wiki).
  - 2.2 Built a model of aspirin (ArgusLab)
  - 2.3 Optimize a geometry of aspirin using MM and AM1 methods, find in the output a value of heat of formation (kcal/mol) by the AM1 method.
  - 2.4 Calculate MEP (charge distribution) in aspirin.
3. Calculation of torsional potential in aspirin (Arguslab).
  - 3.1 Select amino-acetyl group and measure a torsional angle of this group with respect to phenyl ring plane. Sample this angle entry 30 deg, in the range of 0-360 deg, Please make 12 calculations of Total energy. (Here only a single point (one conformation) energy evaluation will be performed, don't make geometry optimization.
  - 3.2 Prepare a plot: Energy (kcal/mol) vs. torsional angle (deg.) Where a minimum on this plot is located? Compare your result with those obtained by classmates. Are the results identical. Explain possible differences.
4. Have a glimpse on NAMD Internet tutorial and local tutorial (separate instruction will be provided by WN).
5. The report (1 page.) from Task 14 contains:
  - A. A Table with results of geometry optimization of water (MM, AM1, PM3, MNDO, exp)
  - B. The torsional potential plot (aspirin)
  - C. The energy of the lowest conformer of aspirin.

## **Task 15.**

### A study of nanomechanics of nanocomposite biomaterial (2h)

Students prepare inputs, perform simulations (MD – locally, SMD on a designated cluster) using molecular dynamics simulation, it is a part of a real scientific practical project. They analyze results of simulations as well. Students construct a model of a *Caddisfly* fibroin fragment (2 beta strands). They learn how to study a strength of nanomaterial. They study the effect of phosphorylation and the presence of  $\text{Ca}^{+2}$  ions. They have to prepare own inputs and scripts. They optimize geometry of a protein (programs used: VMD i NAMD).

1. On a local PC (PKV, Windows): Using a separate instruction prepare \*.bat files and input files. Ask instructor for hints what velocity of pulling should be used and what force constant of a virtual spring is recommended.
2. On a local PC (PKV, Windows): Perform D0 step of MD simulations (minimization, optimization of a water shell). Perform D1 step (MD – equilibration of the whole system).
3. Learn details how to log in into a remote Linux cluster. Transfer appropriate input files, note that sometimes conversion of formats may be needed (Small/big Endian problem etc..). Learn how to submit computational job to a cluster. Submit D2 (SMD) job to the queue. Please, control validity of k\_SMD and Vel, and frequency of storing trajectory frames and writing out data into the output file. (wrong parameters will lead to a fast Crash of your tests). Monitor an output file (\*.out). Sometimes one have to repeat D2 calculations several times before decent results are obtained. Learn how to transfer output file into a local computer. Learn how to check the state of a Job on a cluster (qstat -b).
4. Write your own scripts (or use the scripts prepared by dr L. Peplowski) to extract from \*.out data on a value of applied virtual force and protein extension induced by this SMD force.
5. Prepare a plot (from D2) force – time (or better (force – extension). Check what units on these plots should be inserted.
6. The final report (1-2 pages, EX wn#3) should contain information on parameters used in the simulation (a copy of an input file is ok) and the plot force-time (or extension). The e-mail address for sending the PDF report file will be given during exercises.