

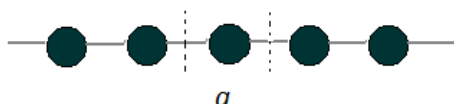
Metody symulacji w nanotechnologii - ćwiczenia

Ćwiczenia w laboratorium komputerowym

Praktyczne zastosowanie metody ciasnego wiązania do numerycznych obliczeń pasm energetycznych wybranych struktur grafenowych

Ćwiczenie 1: Jednowymiarowy łańcuch atomowy. Konstrukcja orbitalu molekularnego i macierzy hamiltonianu

Ćwiczenie jest rachunkowe. Zajmujemy się układem zbudowanym jak na poniższym rysunku.



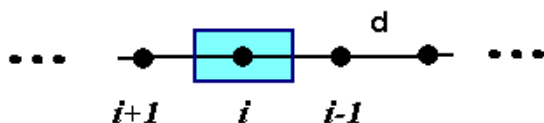
Atomy reprezentowane są przez czarne punkty a wiązania przez łączące je kreski. Wielkość a – to stała sieci. Zakładamy jeden (1) orbital na każdym atomie, oznaczmy go χ i choć może zależeć od 3 współrzędnych przestrzennych symbolicznie przypisujemy mu zależność tylko od x gdyż tylko w tym kierunku łańcuch jest periodyczny. Zakładamy skończoną liczbę węzłów w łańcuchu – N . $\chi(x-na)$ to orbital atomowy centrowany na n -tym atomie w łańcuchu (licząc od początku łańcucha). Tworzymy orbital molekularny (krystaliczny) spełniający twierdzenie Blocha, jako

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

Energia, a właściwie wartość średnia energii w stanie opisywanym funkcją ϕ , czyli tu element macierzy hamiltonianu H , będzie równa

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

Ponieważ każda ϕ jest sumą wielu χ zatem stosując przybliżenie najbliższych sąsiadów (patrz wykład), tzn. zakładając, że różne od zera $\langle \chi_i | \mathbf{H} | \chi_j \rangle$ są tylko dla $j=i$ oraz j różniącym się od i o 1 (najbliższy sąsiad)



oraz kładąc $\langle \chi_i | \mathbf{H} | \chi_i \rangle = 0$ (skalowanie energii) dostaniemy ostatecznie

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

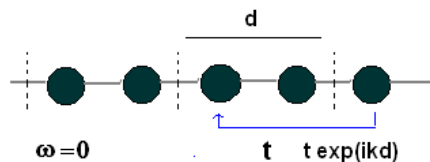
tzn. jedno pasmo $E(k)$.

Należy wyprowadzić wzór (1), narysować go w dowolnym programie graficznym i zinterpretować.

Ćwiczenie 2: Łańcuch jednowymiarowy taki jak w ćwiczeniu 1, ale z komórką elementarną 2 razy większą (dwa atomy w komórce)

Cele ćwiczenia: (1) wyprowadzenie wzorów na elementy macierzowe hamiltonianu, (2) zrozumienie, że powstaną dwa pasma energetycznie równoważne wynikowi ćwiczenia 1, (3) pokazać, że pasma energetyczne można uzyskać z rezultatów ćwiczenia 1 poprzez „złożenie” pasma do 2-krotnie mniejszej strefy Brillouina ($d=2a$).

Tym razem mamy łańcuch jednowymiarowy z dwoma atomami w komórce elementarnej



Macierz hamiltonianu ma wymiar 2×2 (bo dwa atomy w komórce elementarnej), zakładając, że elementy diagonalne kładziemy = 0, macierz H ma postać:

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

Widmo energii jest takie samo, choć teraz formalnie złożone z 2 pasm (bo dwie wartości własne H). Ważne jest aby zauważyć różnicę w wielkości strefy Brillouina. Macierz można zdiagnozować „ręcznie” lub już na tym etapie napisać program, który będzie albo wczytywał macierz hamiltonianu, albo ją konstruował i wywoływał procedurę diagonalizacyjną.

Ćwiczenie 3: Konstrukcja macierzy H dla kilkuatomowej „molekuły” węgla, np. dla „molekuły” C_4



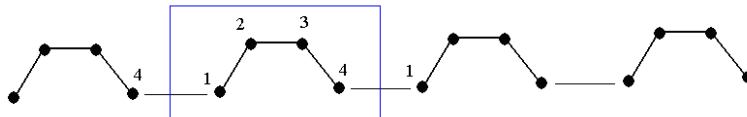
$H_{ii} = 0$, $i=1,2,3,4$ - jako diagonalne elementy macierzowe, $H_{12} = H_{21} = H_{23} = H_{32} = H_{34} = H_{43} = t$ [$t = -2.7$ eV] wszystkie pozostałe $H_{kl} = 0$; tu H nie zależy od k

Ćwiczenie 4: Konstrukcja macierzy H dla najprostszej wstęgi grafenowej typu fotelowego

Uwaga.

Wymiar macierzy hamiltonianu równy jest liczbie węzłów (atomów węgla) w komórce elementarnej. Należy najpierw ponumerować węzły w wybranej komórce elementarnej i w sąsiednich komórkach tak jak to jest pokazane na poniższym rysunku. Tworząc macierz hamiltonianu należy pamiętać, że

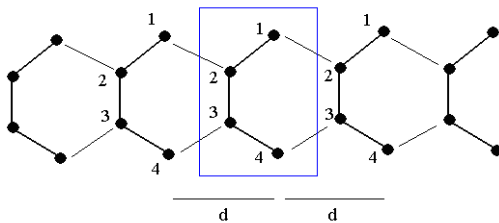
elementy nawęzłowe (diagonalne) kładziemy równe zero (jako wyraz dowolności wyboru zera na skali energii). Elementy H odpowiadające połączeniom (wiązaniom chemicznym π -elektronowym) pomiędzy sąsiednimi węzłami tylko w jednej komórce elementarnej kładziemy równe t . Połączenie węzła „i” w danej komórce z węzłem „j” sąsiedniej komórki realizowane jest przez $t \cdot \exp(ikd)$ [i-jednostka urojona], lub $t \cdot \exp(-ikd)$, w zależności od tego czy komórka z węzłem „i” leży na prawo czy na lewo od danej-wybranej komórki elementarnej. Jeśli jednak węzeł „i” połączony jest z węzłem „j” z tej samej komórki oraz z węzłem „j” z sąsiedniej komórki, to odpowiedni element H wynosi $t[(1+\exp(ikd))]$ lub $t[(1+\exp(-ikd))]$. Sprawdzenie poprawności znaków w eksponencie dokonuje się przez sprawdzenie hermitowskości macierzy H.



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

Ćwiczenie 5: Konstrukcja macierzy H dla najprostszej wstęgi grafenowej typu zygzak

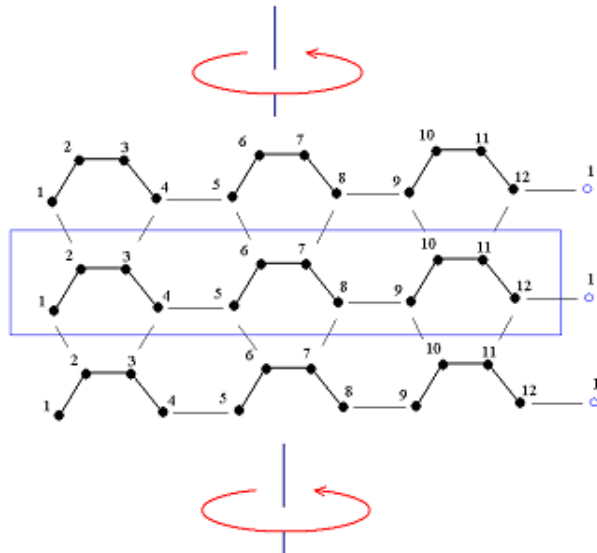
Uwaga – czytaj z ćwiczenia 4.



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

Ćwiczenie 6: Konstrukcja macierzy H dla nanorurki (3,3).

Uwaga: czytaj z ćwiczenia 4. Dodatkowo należy pamiętać, że w komórce elementarnej nanorurki wszystkie węzły mają trzech sąsiadów.



A macierz H (wymiar = 12):

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

Ćwiczenie zaliczeniowe

Budowa, testowanie i uruchomienie programu komputerowego pozwalającego obliczać strukturę energetyczną dowolnych układów grafenowych, periodycznych i nieperiodycznych

Ogólna struktura programu powinna być następująca

- Określenia liczby węzłów (atomów węgla) w komórce elementarnej, $N = \dim H$,
- definicji wiązań wewnątrz komórki elementarnej (liczba elementów które $H_{ij} = t$),
- definicji połączeń pomiędzy komórkami elementarnymi ($t e^{ik}$, lub $t[1+e^{ik}]$),
- pętli po wartościach k (w strefie Brillouina),
 - diagonalizacji macierzy H dla każdego k , wybraną metodą, w zależności od wybranego języka programowania lub pakietu obliczeniowego,
- możliwości symulacji siły wiązania (t) w różnych obszarach,
- możliwości modyfikacji komórki elementarnej poprzez dodawanie lub odejmowanie węzłów,
- zapisu danych do pliku;
- wykonanie wykresów pasm energetycznych (w ramach kodu) lub wykorzystując zewnętrzne narzędzia graficzne w oparciu o dane zapisane do pliku

Studenci mają prawo wybrać język programowania (Fortran, C, C++, C#, Matlab, Python, itd.) Mogą korzystać z gotowych bibliotek zawierających procedury do diagonalizacji.

Na zaliczenie konieczne jest wykazanie się znajomością kodu oraz uruchomienie programu dla wybranych struktur (np. nanorurka (6,6), nanorurka (5,0), nanowstęga grafenowa zygzak z jednym brzegiem modyfikowanym przez dodatnie tzw. węzłów Kleina).

Stany rezonansowe cząstek w nanostrukturach

Mirosław Bylicki

Przedstawiamy trzy zadania dotyczące stanów rezonansowych, przeznaczone do wykonania w czasie ćwiczeń w laboratorium komputerowym i pracy własnej. Są to zadania do wyboru przez prowadzącego dla danej grupy studentów lub w danym cyklu kształcenia. (Każdy student wykonuje tylko jedno z nich). Zadania polegają na przeprowadzeniu wstępnych przygotowań analitycznych (na papierze), przygotowaniu programu do obliczeń i graficznej prezentacji wyników. Student sam decyduje o wyborze używanego języka programowania i środowiska, w którym chce pracować.

Ćwiczenie 7: Stany rezonansowe symetrycznego układu studni i dwóch barier

Układ składa się z trzech warstw półprzewodnikowych. Zachowanie elektronu w tym układzie modelowane jest jednowymiarowym potencjałem typu prostokątne bariery i studnia:

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

gdzie $V_0 > 0$.

Hamiltonian jest następujący:

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

Należy rozwiązać równanie Schroedingera dla energii z zakresu od 0 do V_0 w sposób analogiczny do przedstawionego na wykładzie, w rozdziale „Stany rezonansowe układu studnia bariera”. Należy wykorzystać symetrię układu – ponieważ układ nie zmienia się przy inwersji $x \rightarrow -x$, gęstość prawdopodobieństwa powinna być funkcją parzystą, tzn. $|\psi(-x)|^2 = |\psi(x)|^2$. Same funkcje mogą być parzyste, $\psi(-x) = \psi(x)$, lub nieparzyste, $\psi(-x) = -\psi(x)$. Wykorzystanie symetrii polega na znajdowaniu oddzielnie parzystych i nieparzystych rozwiązań. Zakładamy zatem odpowiednie związki między współczynnikami występującymi w funkcji z lewej i prawej strony. Zmniejsza to liczbę niewiadomych i czyni problem nieco łatwiejszym.

Wyniki należy przedstawić w postaci graficznej ze szczególnym uwzględnieniem rezonansowego charakteru rozwiązań, podobnie jak w rozdziale „Stany rezonansowe układu studnia bariera”. Mile widziane własne pomysły dotyczące sposobu przedstawienia wyników.

Program ma zapewniać możliwość zmiany parametrów fizycznych układu dotyczących potencjału a, b, V_0 i masy efektywnej elektronu m .

Ćwiczenie 8: Stany rezonansowe w symetrycznej strukturze warstwowej modelowanej potencjałem analitycznym

Struktura podobna jak w poprzednim zadaniu modelowana jest ciągłym potencjałem

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

gdzie a, b i q są nieujemnymi parametrami rzeczywistymi.

Przyjmujemy hamiltonian w postaci

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

Zadanie polega na **zastosowaniu metody stabilizacji** do znalezienia stanów rezonansowych w zakresie energii od asymptoty potencjału (poziomej energii równy 0; $\lim_{x \rightarrow \infty} V(x) = 0$) do maksymalnej wartości potencjału (wysokości bariery) i stanów związanych, w zakresie energetycznym pomiędzy minimum potencjału ($-b$) a zerem (poziomym asymptotycznym).

Należy wykorzystać symetrię układu ($V(-x) = V(x)$) poszukując oddzielnie rozwiązań parzystych i nieparzystych. Dla stanów parzystych stosujemy bazę funkcji parzystych

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

a dla stanów nieparzystych – bazę funkcji nieparzystych

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

Wartości parametrów β_i w obydwu przypadkach ustalamy tak, aby tworzyły ciąg geometryczny. Zadajemy tylko wartości β_1 i β_N .

Przed przystąpieniem do programowania należy wyprowadzić wzory na elementy macierzowe $H_{ij}(\alpha) = \langle \varphi_i^\alpha | \hat{H} | \varphi_j^\alpha \rangle$ i $S_{ij}(\alpha) = \langle \varphi_i^\alpha | \varphi_j^\alpha \rangle$.

Warto przy tym zwrócić uwagę na następujące zależności

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

($T_{ij}(\alpha)$ są elementami kinetycznej części macierzy hamiltonianu)

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

Należy zależności te wykorzystać przy wyprowadzaniu wzorów a następnie przy konstruowaniu programu tak, aby uniknąć wielokrotnego obliczania tych samych elementów macierzowych. W szczególności elementy macierzy \mathbf{S} i \mathbf{T} mogą być obliczone raz a przy zmianie wartości α tylko mnożone przez odpowiednie czynniki.

Do rozwiązania równania

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

należy użyć gotowej procedury, którą można znaleźć np. w bibliotece LAPACK.

Wśród danych wejściowych do programu mają znaleźć się dane charakteryzujące układ: parametry potencjału a, b, q i masa efektywna elektronu m oraz dane dotyczące metody, tzn. używanej bazy: liczba funkcji bazowych N i wykładniki β_1 i β_N .

Wyniki powinny być przedstawione graficznie w postaci grafu stabilizacyjnego (Rys. 39 w wykładzie) i histogramu przedstawiającego gęstość pierwiastków otrzymanych w pobliżu poziomu rezonansowego wraz z dopasowaną krzywą Lorentza (Rys. 40). Ostatecznym wynikiem mają być wartości położenia poziomu rezonansowego, E_r , i jego szerokości, Γ , uzyskane z dopasowania profilu Lorentza.

Ćwiczenie 9: Stany rezonansowe w sferycznie symetrycznej kropce kwantowej

Rozważana tu sferycznie symetryczna kropka kwantowa jest kulką z jednego półprzewodnika otoczoną sferyczną warstwą drugiego i zatopiona w materiale tworzącym nieskończone otoczenie. Oddziaływanie tej struktury na znajdujący się w niej elektron modelujemy sferycznie symetrycznym potencjałem

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

gdzie a, b i q są parametrami o rzeczywistych wartościach dodatnich.

Problem, choć w zasadzie trójwymiarowy sprowadza się do zagadnienia radialnego (zależnego tylko od współrzędnej r) z hamiltonianem

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

gdzie $l(l+1)$ jest wartością momentu pędu cząstki; $l = 0, 1, 2 \dots$.

Zadanie polega na **zastosowaniu metody obrotu zespolonego współrzędnych** do znalezienia stanów rezonansowych elektronu w kropce w zakresie energii od asymptoty potencjału (poziom energii równy 0) do maksymalnej wartości potencjału (wysokości bariery).

Należy zastosować bazę N funkcji

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

gdzie n jest takie samo dla wszystkich funkcji a parametry β_i tworzą ciąg geometryczny, zadany przez podanie skrajnych elementów: β_1 i β_N .

Przed przystąpieniem do pisania programu należy przygotować wzory na elementy macierzowe

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

oraz

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

Zwróćmy uwagę na to, że elementy S_{ij} nie zależą od kąta obrotu θ . Podobnie zależność od θ części kinetycznej macierzy hamiltonianu da się wyłączyć w postaci stałego czynnika

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

Obliczanie elementów S_{ij} i $T_{ij}(\theta = 0)$ będzie można w programie wyłączyć poza pętlę, w której zmieniamy θ . Nie da się tego zrobić dla części potencjalnej macierzy hamiltonianu

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

Uwaga: wszystkie występujące tu całki sprowadzają się do całek typu

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

Do rozwiązania uogólnionego problemu własnego

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

należy użyć odpowiedniej procedury z biblioteki LAPACK.

Dane wejściowe do programu muszą zawierać parametry potencjału a, b, q , masę efektywną elektronu m i liczbę l określającą wartość momentu pędu elektronu oraz dane dotyczące bazy funkcyjnej: liczbę funkcji N i ich parametry n, β_1 i β_N . Wyniki powinny być przedstawione w postaci graficznej, tj. w postaci θ -trajektorii (patrz rys. 42). Ostatecznym wynikiem jest optymalna wartość zespolonej energii rezonansu.

Zastosowanie metody gradientu prostego do numerycznych obliczeń odkształceń w jednowymiarowym łańcuchu atomów

Michał Zieliński

Ćwiczenie 10: Implementacja metody gradientu prostego dla prostej funkcji w postaci analitycznej

Napisz program do obliczeń metodą gradientu prostego oraz wyznacz minimum funkcji dwuwymiarowej funkcji: $f(x,y) = x^2 - 6x + 9 + 2y^2 - 8y + 8$ dla której gradient (wektor pochodnych cząstkowych) można w prosty sposób policzyć analitycznie.

Ćwiczenie 2: Implementacja metody gradientu prostego dla funkcji wielowymiarowej

Napisz program do obliczeń metodą gradientu prostego dowolnej funkcji N-wymiarowej ($N \leq 10$) zdefiniowanej przez użytkownika. Rozpatrz dwa warianty:

- A. gradient można wyznaczyć analitycznie
- B. gradient wyznaczony numerycznie, np.:

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

Ćwiczenie zaliczeniowe

Budowa, testowanie i uruchomienie programu komputerowego pozwalającego obliczać położenia atomów minimalizujące energię odkształceń w jednowymiarowym łańcuchu atomów

Wykorzystaj metodę gradientu prostego oraz napisz programu znajdujący minimum energii odkształceń dla jednowymiarowego łańcucha N atomów w postaci:

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

gdzie kolejne x_i oznaczają położenia atomów, d_{ij} długość wiązania między atomem i oraz j , A_{ij} stała siłowa związana z rozciąganiem/ściskaniem wiązań międzyatomowych.

Poniższy rysunek pokazuje przykładowy format danych wejściowych dla łańcucha, gdzie rodzaj atomów kodowany jest za pomocą koloru (czerwony, zielony, niebieski, ang. Red, Green, Blue). Podane są też długości wiązań i odpowiadające im stałe siłowe. Podobnie jak dla prawdziwych

materiałów półprzewodnikowych (np. InAs) możliwe są wiązania pomiędzy atomami określonego koloru (rodzaju), tj. B-G oraz B-R.

Uwaga: położenia krańcowych atomów (x_0 oraz x_{N-1}) nie zmieniają się (są zamrożone) w czasie minimalizacji!!!

$$d_{BR} \quad 1.5$$

$$A_{BR} \quad 2.0$$

$$d_{BG} \quad 1.75$$

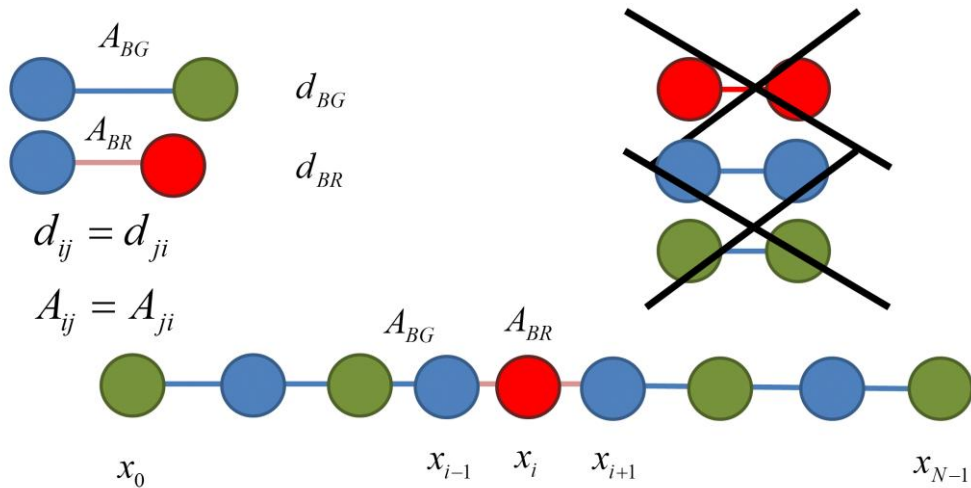
$$A_{BG} \quad 2.5$$

$$x_0 \quad 0$$

$$x_8 \quad 18.0$$

$$N \quad 9$$

G B G B R B G B G



Wyniki programu powinny być zaprezentowane w postaci położenia kolejnych atomów oraz odkształcenia na każdym atomie z wnętrza łańcucha:

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

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

Program powinien również podać łączną liczbę iteracji, a także wypisać długości wektora gradientu i wartości energii odkształceń dla każdej z iteracji.

Studenci mają prawo wybrać język programowania (Fortran, C, C++, C#, Matlab, Phyton, itd.).

Na zaliczenie konieczne jest wykazanie się znajomością kodu oraz uruchomienie programu dla wybranych struktur (danych wejściowych).

Modelowanie metodą SMD nanomechaniki biomateriału

Ćwiczenie ma na celu zapoznanie Państwa z programem do obliczeń dynamiki molekularnej NAMD (<http://www.ks.uiuc.edu/Research/namd/>) oraz programem analizującym wyniki symulacji VMD (<http://www.ks.uiuc.edu/Research/vmd/>). Obie te aplikacje są dostępne za darmo dla użytkowników akademickich na podanych stronach. Programy mogą być uruchamiane pod Linuksem i w środowisku Windows.

Poniższy tutorial jest pisany z myślą o użytkowniku korzystającym z systemu Windows.

Przedmiotem ćwiczeń jest fragment białka Fib H (*Fibroin heavy chain*), składający się z 31 aminokwasów. Białko to jest jednym ze składników oprzędu produkowanego przez pewne owady - chrzączki. Jest ono bogate w seryny. Przygotujemy model tego białka, w którym seryny będą fosforylowane. Oprócz tego w symulowanym układzie będą występować jony Ca^{2+} .

Należy samodzielnie wykonać podstawowe symulacje (dalej nazwane D0 i D1) metodą Dynamiki Molekularnej (MD) oraz metodą Sterowanej Dynamiki Molekularnej (SMD), w której poprzez przyłożenie zewnętrznej siły do wybranej grupy atomów należy zbadać wytrzymałość tego biomateriału (krok D2).

VMD Visual Molecular Dynamics – podstawy

1. Proszę wejść do katalogu **NAMD_symulacje**, który znajduje się na dysku C. Znajduje się tam plik o nazwie **beta_sp2.pdb**. Jest to plik ze strukturą przestrzenną fragmentu białka Fib H. Fragment ten został utworzony wcześniej w odpowiednim programie do modelowania (KM), dlatego też nie można go pobrać ze strony: <http://www.pdb.org/pdb/home/home.do> (Protein Data Bank, **PDB**). Jak już wiemy (ćw. wn#1), PDB jest biologiczną bazą danych, na której gromadzi się informacje o strukturze przestrzennej białek i w pewnym stopniu DNA. Na stronie PDB oprócz samych struktur białek można dodatkowo znaleźć inne użyteczne informacje: sekwencja białka, struktura drugorzędowa, metody wykorzystane do określenia struktury danego białka (NMR, krystalografia, mikroskop elektronowy), a także odnośniki do innych stron WWW związanych z analizowanym białkiem.
2. Obsługa VMD: Proszę obejrzeć strukturę naszego fragmentu białka Fib-H używając programu **VMD**. Kliknij dwa razy na plik **beta_sp2.pdb** z białkiem. Program VMD zostanie uruchomiony, a w nim powinna pojawić się struktura przestrzenna naszego modelu białka. Należy „poobracać” białko używając myszki i przycisków. Domyślnie włączona jest opcja *rotate* (*R* – skrót na klawiaturze, przy aktywnym oknie z wizualizacją białka). Jeśli chcemy powiększyć białko wciskamy *S* na klawiaturze lub używamy scrolla. Klawisz *T* (*translate*) służy do zmiany położenia białka w oknie do wizualizacji. Można je dowolnie zmieniać w trakcie pracy w programie.

Wyświetlamy białko w innej reprezentacji: *Menu Main -> Graphics -> Representations -> Drawing Method*. Proszę wypróbować kilka reprezentacji modelu np. Licorice, CPK, VDW, NewCarton. Można również zmienić barwy stosując *Coloring Method*.

Proszę utworzyć nową reprezentację białka, w której wyświetlone będą wiązania wodorowe: *Create Rep*, należy kliknąć na nowo powstałą reprezentację i zmienić *Drawing method* na

HBonds. Proszę zmienić wartości *Distance Cutoff* (odległość) oraz *Angle Cutoff* (kąt). W ten sposób nauczymy się ustawiać pożądane filtry na strukturze modelu.

Wyświetlmy aminokwas *SER25* w reprezentacji CPK (*Create Rep* -> w *selected atoms* wpisujemy: *resid 25*). Teraz wyświetlmy wszystkie seryny w białku. Podobnie jak poprzednio tworzymy nową reprezentację i w *selected atoms* wpisujemy tym razem *resname SER*. Wybieramy reprezentację *Licorice*. Spróbujmy wyświetlić również jony poleceniem *ion*.

Wykorzystując klawisze numeryczne można sprawdzić nazwę atomu (1), odległość między wybranymi atomami (2), kąt (3) oraz kąt dwuścienny (4). W *Menu Main* -> *Graphics* -> *Labels* można dodatkowo manipulować parametrami, które zaznaczyliśmy.

NAMD Steered Molecular Dynamics

3. Uruchamianie symulacji w programie NAMD.

A. Tworzenie pliku psf.

Do wykonania symulacji niezbędne są dwa pliki: plik ze współrzędnymi atomów *.pdb oraz plik ze strukturą (topologią) białka *.psf (PSF = *protein structure file*, należy samemu utworzyć ten plik).

Do utworzenia pliku *.psf w pakiecie NAMD wykorzystujemy program *psfgen*. Program ten znajduje się w katalogu NAMD na dysku C. By posłużyć się tym programem potrzebne nam będą dwa pliki skryptowe: *psfgen_build.bat*, który korzysta z informacji zawartych w pliku PDB oraz z plików z topologią (*top_all_27_prot_na.inp* oraz *toppar_prot_na_all.str*). Drugim plikiem jest plik *psfgen.bat*, w którym zamieszczona jest ścieżka do programu *psfgen* z pakietu NAMD. Pliki *.bat można edytować np. w notatniku. Plik z topologią *top_all_27_prot_na.inp* jest wspólny dla wszystkich białek, a plik *toppar_prot_na_all.str* zawiera dodatkowe informacje dotyczące aminokwasów niestandardowych (po modyfikacjach postranslacyjnych, np. fosforylacji). Tworzymy odpowiedni skrypt, mający postać:

psfgen.bat:

```
"C:\NAMD\psfgen.exe"
```

psfgen_build.bat:

```
topology top_all27_prot_na.inp          # plik z topologią aminokwasów
topology toppar_prot_na_all.str        # jest potrzebny ponieważ chcemy by seryna
                                        była fosforylowana

alias residue CA CAL                   # aliasy PDB-NAMD
alias atom CAL CA CAL

segment CHR {                          # budowanie segmentów
pdb beta_sp2.pdb
}

patch SP2 CHR:3                        # konieczne ponieważ chcemy seryny fosforylowane
patch SP2 CHR:5
patch SP2 CHR:7
patch SP2 CHR:9
patch SP2 CHR:23
patch SP2 CHR:25
patch SP2 CHR:27
patch SP2 CHR:29
regenerate angles dihedrals
coordpdb beta_sp2.pdb CHR

segment CAL {                          # budowanie segment dla jonów Ca2+
first none
```

```

last none
pdb calcium.pdb
}
coordpdb calcium.pdb CAL

writepsf chrust_ca_sp2.psf          #zapis PSF
guesscoord
writepdb chrust_ca_sp2.pdb          #zapis PDB

```

Polecenia:

```

Start -> Uruchom -> polecenie: cmd
W konsoli Windowsa:
cd C:\NAMD_symulacje
psfgen.bat < psfgen_build.bat

```

Często zdarza się, że białka posiadają kilka oddzielnych łańcuchów, wtedy należy każdy oddzielny łańcuch umieścić w oddzielnym pliku i stworzyć kilka segmentów. W plikach pobranych z PDB może również wystąpić woda tzw. „krystalograficzna”. Dla niej również powinien zostać stworzony oddzielny segment. W naszym przypadku konieczne jest utworzenie dwóch segmentów. Jeden dla białka, a drugi dla jonów Ca^{2+} .

- B. Przygotowanie białka do symulacji. Po stworzeniu pliku psf można przejść do przygotowania białka do pierwszej symulacji. Żeby białko było w możliwie zbliżonym środowisku do naturalnego należy umieścić je w otoczce wodnej. Możemy to zrobić wykorzystując program VMD. Wczytujemy do programu VMD plik pdb (*VMD Main -> File -> New Molecule*). Poprzez przycisk *Browse* wybieramy plik **chrust_ca_sp2.pdb** i wczytujemy go do programu (*Load*). Plik ten można również otworzyć w programie VMD poprzez podwójne kliknięcie. Do wykonania kolejnych czynności trzeba również wskazać plik **chrust_ca_sp2.psf**. W tym celu w oknie *VMD Main* należy zaznaczyć wczytany wcześniej plik pdb i prawym przyciskiem wybrać opcję: *Load Data Into Molecule*. Wybieramy plik **chrust_ca_sp2.psf**.

Aby dodać wodę do naszego układu chodzimy w *Extension -> TkConsole*. Ukazuje nam się konsola Tk. Wpisujemy w niej polecenie (**Uwaga: Należy zachować kolejność!**):

```
solvate chrust_ca_sp2.psf chrust_ca_sp2.pdb -t 5 -rotate -o chrust_ca_sp2_s
```

„Skrzynka” wody (model TIP3P, Jeśli stosujemy jak tutaj pole siłowe CHARMM) będzie miała 5Å w każdą stronę od skrajnych atomów białka. Po tej wykonaniu tego polecenia zostaną utworzone dwa nowe pliki o nazwie chrust_ca_sp2_s.pdb i chrust_ca_sp2_s.psf. Następnie centrujemy skrzynkę, którą stworzyliśmy, poleceniem (w *TkConsole*):

```
play ctrbox.tcl oraz
```

```
ctrbox chrust_ca_sp2_s.psf chrust_ca_sp2_s.pdb chrust_ca_sp2_sc
```

- C. Symulacja dynamiki ekwilibracji (zrównoważanie) wody i podgrzewania układu. Po wykonaniu powyższych czynności mamy przygotowany układ do symulacji ekwilibracji wody. Symulacja ta jest przeznaczona przede wszystkim dla wody. Białko na czas symulacji zostaje „zamrożone”. Dodając wodę w programie można zauważyć, że jest ona ułożona w bardzo sztuczny sposób, co nie jest zgodne z przyrodą. Wykonując ekwilibrację wody mamy przede wszystkim dwa cele: „zaburzenie regularności” wody oraz ułożenie wody wokół białka. Skrypt do ekwilibracji wody:

Nazwa skryptu: NAMD_D0.inp

```

structure      chrust_ca_sp2_sc.psf
coordinates    chrust_ca_sp2_sc.pdb

```

```

paratypecharm on
parameters      par_all27_prot_na.inp      # plik z parametrami dla wszystkich białek
parameters      toppar_prot_na_all.str
exclude         scaled1-4
1-4scaling      1.0

switching       on                        # parametry związane z oddziaływaniami van der
switchdist      8                        Waalsa i elektrostatycznymi
cutoff          12
pairlistdist    14
margin          2.5
stepspercycle   20

wrapWater       on
rigidBonds      water

timestep 1.0                    # krok czasowy (1 fs)

outputenergies  200
outputtiming     200
binaryoutput    yes
outputname      N_beta_ca_sp2_D0         # nazwa pliku z wynikami (*.dcd)
dcdfreq        200

temperature     300

langevin        on                    # Dynamika Langevina (regulacja temperatury)
langevinDamping 5
langevinTemp    300
langevinHydrogen no

useFlexibleCell yes
useGroupPressure yes

LangevinPiston on                    # Dynamika Langevina (regulacja ciśnienia)
LangevinPistonTarget 1.01325
LangevinPistonPeriod 200
LangevinPistonDecay 100
LangevinPistonTemp 300

cellBasisVector1 27.980998993          0.0    0.0          # wektory bazowe należy
cellBasisVector2      0.0    62.9489974976  0.0          skopiować w to miejsce
cellBasisVector3      0.0    0.0    23.3190002442       z pliku chrust_ca_sp2_sc.pdb
cellOrigin            0.0    0.0    0.0

fixedAtoms        on
fixedAtomsFile    beta_ca_sp2_fix.pdb   # plik wskazuje, które atomy mają być "zamrożone"
fixedAtomsCol    0                      podczas ekwilibracji wody - jest nim całe białko

minimize 100                    # minimalizacja
run 1000                        # czas symulacji 1000 fs (1 ps)

```

Wartości wektorów bazowych `cellBasisVector1` brane są z pliku PDB, który jest wynikiem działania skryptu `ctrbox.tcl`. Są to wektory definiujące główną skrzynkę symulacyjną, wg tych danych utworzone będą skrzynki obrazowe przy stosowaniu periodycznych warunków brzegowych.

Plik `beta_ca_sp2_fix.pdb`, który zostanie użyty w symulacji D0 został wcześniej utworzony wykorzystując skrypt `fixowanie_białka.tcl`. Skrypt ten „zamraża białko” poprzez umieszczenie „1” w kolumnie *Occupancy* w pliku `chrust_ca_sp2_sc.pdb` i zapisuje plik pod nową nazwą.

Polecenie:

Skrypt uruchamiamy w konsoli Windows poleceniem:

```
run_namd_D0.bat
```

Następną symulacją, do której trzeba przygotować plik uruchamiający jest podgrzewanie układu. W symulacji tej wykonuje się minimalizację całego układu oraz stopniowe podgrzewanie układu od 0K do 300K. Te czynności zapobiegają gwałtownym zmianom prędkości atomów w białku. Po wykonaniu tej symulacji białko gotowe jest do tzw. „właściwej symulacji”, czyli symulacji MD lub SMD.

Skrypt do symulacji podgrzewania układu (NAMD_D1.inp):

```
structure      chrust_ca_sp2_sc.psf
coordinates    chrust_ca_sp2_sc.pdb
bincoordinates N_beta_ca_sp2_D0.coor      # plik wynikowy z symulacji D0

paratypecharm on
parameters    par_all27_prot_na.inp
parameters    toppar_prot_na_all.str
exclude       scaled1-4
1-4scaling    1.0

switching     on
switchdist    8
cutoff        12
pairlistdist  14
margin        0
stepspcycle   20

wrapWater     on
rigidBonds    water

timestep 1.0      # krok czasowy w fs

outputenergies 200
outputtiming    200
binaryoutput   yes
outputname     N_beta_ca_sp2_D1
dcdfreq       200

temperature    0      # temperatura początkowa

reassignFreq   1000    # blok związany ze stopniowym podgrzewaniem układu
reassignIncr   25      co 1000 kroków podnoszona jest temperatura o 25K
reassignHold   300    aż do osiągnięcia 300K.

extendedSystem N_beta_ca_sp2_D0.xsc # plik wynikowy z symulacji D0

minimize 500      # liczba kroków minimalizacji
run 20000         # liczba kroków symulacji (=20ps)
```

Polecenie:

Skrypt uruchamiamy w konsoli Windows poleceniem:

```
run_namd_D1.bat
```

- D. „Właściwa symulacja – SMD”. Symulacja SMD jest metodą pozwalającą na naśladowanie eksperymentu wykonywanego Mikroskopem Sił Atomowych (AFM). Służy do rozciągania molekuł. W metodzie tej można dowolnie określić kierunek oraz prędkość ruchu „atomu

wirtualnego” do rozciągania molekuly oraz „sztywnosc” wirtualnej sprężyny, przy pomocy której przykładamy siłę zewnętrzną do wybranej grupy atomów (lub atomu).

Przykładowy skrypt do symulacji SMD:

```
structure      chrust_ca_sp2_sc.psf
coordinates    chrust_ca_sp2_sc.pdb
bincoordinates N_beta_ca_sp2_D1.coor      # plik wynikowy z symulacji D1

paratypecharm on
parameters    par_all127_prot_na.inp
parameters    toppar_prot_na_all.str
exclude       scaled1-4
1-4scaling    1.0

switching     on
switchdist    8
cutoff        12
pairlistdist  14
margin        0
stepspercycle 20

wrapWater     on
rigidBonds    water

timestep      2.0                      # krok czasowy w fs

nonbondedFreq 2
fullElectFrequency 4

outputenergies 200
outputtiming    200
binaryoutput   yes
outputname     N_beta_ca_sp2_smd
dcdfreq       200

binvelocities  N_beta_ca_sp2_D1.vel      # plik wynikowy z symulacji D1

langevin      on
langevinDamping 5
langevinTemp  300                      # kontrola temperatury tu 300K
langevinHydrogen no

SMD on                                               # blok związany z SMD
SMDfile beta_ca_sp2_doSMD.pdb                     # plik z zamrożonymi atomami (do smd)
SMDk 4                                             # stała sprężystości sprężyny zewnętrznej
SMDVel 0.0025                                     # prędkość „atomu holownika”
SMDDir 4.821 -0.812 -1.203                       # wektor kier. siły rozciągaj (TRZEBA WYZNACZYĆ!)
SMDOutputFreq 200

fixedAtoms on
fixedAtomsFile beta_sp2_ca_FIX_doSMD.pdb          # plik z zamrożonymi atomami
fixedAtomsCol 0

run 2000000 (2ns)
```

Linie wyróżnione kolorem wymagają indywidualnej uwagi Studenta. Tymi parametrami „manipujemy” chcąc uzyskać jak najlepsze wyniki modelowania tego fragmentu białka.

W pliku N_beta_ca_sp2_D1.vel są zachowane prędkości atomów po zakończeniu podgrzewania układu. W pliku fixedAtomsFile określamy atomy, które mają nie zmieniać pozycji podczas trwania symulacji. Natomiast w pliku SMDfile znajdują się atomy, do których przykładamy siłę.

Wektor siły (rozciągającej) w symulacji SMD trzeba obliczyć własnoręcznie. Należy wczytać pliki: `chrust_ca_sp2_sc.pdb` i `chrust_ca_sp2_sc.psf` do programu VMD oraz wyznaczyć współrzędne atomu C α dla pierwszego i ostatniego aminokwasu. Wciskamy „1” na klawiaturze i lewym przyciskiem myszy klikamy na wybrany atom. Współrzędne atomu pojawią się w konsoli programu. Dalej należy odjąć od siebie poszczególne współrzędne atomów wg schematu: $x_2 - x_1$, $y_2 - y_1$, $z_2 - z_1$. Współrzędne z indeksem 1 będą należały do atomu C α LYS1, ponieważ to lizyna zostanie w tej symulacji SMD zamrożona (zostaje to zaznaczone w pliku `beta_sp2_ca_FIX_doSMD.pdb`). Natomiast współrzędne z indeksem 2 będą należały do atomu C α GLU31 (plik `beta_ca_sp2_doSMD`). GLU31 będzie „ciągniętym” atomem. Do niego zostanie przyłożona zewnętrzna siła, dzięki której możliwe będzie rozciągnięcie białka.

Polecenie:

```
Skrypt uruchamiamy w konsoli Windowsa poleceniem:  
run_namd_SMD.bat
```

E. Pliki wynikowe.

Z każdej symulacji wykonanej programem NAMD otrzymuje się co najmniej pięć plików wyjściowych. DCD – plik zawierający trajektorie wszystkich atomów w kolejnych klatkach symulacji, COOR – współrzędne atomów z ostatniej ramki symulacji, VEL – wektory prędkości wszystkich atomów z ostatniej ramki symulacji oraz XSC – dane dotyczące okresowych warunków brzegowych. Plik *dcd* jest plikiem, który służy do analizy wyników, które możemy obejrzeć w programie VMD. Plik *out* zawiera informacje o zmianach energii w czasie trwania symulacji, ciśnieniu, temperaturze itp.

F. Analiza wyników

Na ćwiczeniach będzie podany adres WWW, z którego będzie można pobrać przykładowy skrypt do wydobywania danych z pliku *out* związanych z wytrzymałością białka. Należy przy jego pomocy (lub przy pomocy własnego skryptu) wydobyć zestawy par siła rozciągająca – czas (w pikosekundach) symulacji. Wykres siła-czas należy przedstawić na zaliczenie ćwiczenia. Warunki symulacji (czas symulacji D2, realistyczna prędkość) należy dobrać eksperymentalnie lub w oparciu o doświadczenia komputerowe kolegów. Obliczenia D2 są czasochłonne (raczej dni niż godziny dla (np. 2 ns) trajektorii SMD ..) więc będą wykonywane zdalnie, albo na klastrze w UCI albo na lokalnym klastrze linuxowym „huncwot” w COK, zależnie od aktualnego stanu sprzętu. Informacje poda Wiesław Nowak.