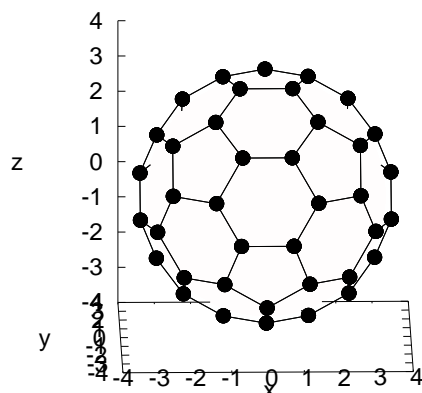


Monte Carlo: modelowanie struktury fullerenów metodą symulowanego wyżarzania

25 kwietnia 2022

1 Wstęp



Rysunek 1: Struktura przestrzenna fullerenu C60 - widok z przodu, tylna część przesłonięta. Położenia atomów węgla zaznaczone są kropkami, a wiązania pomiędzy najbliższymi sąsiadami - czarnymi liniami. Skala na osiach xyz w angstromach. Średnica C60 wynosi 7.04Å.

Na zajęciach wykorzystamy metodę symulowanego wyżarzania w celu określenia struktury geometrycznej fullerenów czyli klasterów zbudowanych z atomów węgla. W strukturach 3D (struktura diamentu) orbitale atomowe hybrydują w konfiguracji sp^3 (tetraedr - 4 najbliższych sąsiadów) natomiast w strukturach powierzchniowych (grafen, fullereny) w konfiguracji sp^2 (3 najbliższych sąsiadów) - podobnie jak w graficie, gdzie powierzchnie grafenu oddziałują siłami van der Waalsa. Modelowanie wykonamy używając potencjału Brennera, poprawnie opisuje on konfiguracje przestrzenne 2D i 3D atomów węgla. Potencjał i jego parametryzacja przedstawione są w pracy: **D.W. Brenner, Phys. Rev. B 42, 9458(1990)**. Więcej o innych typach potencjałów stosowanych w modelowaniu molekularnym można dowiedzieć się z książki dostępnej w wersji elektronicznej(wyszukiwarka pod adresem springerlink.com): **I.A. Solovyov, A.V. Korol, A.V. Solovyov, "Multiscale modelling of complex molecular structure and dynamics with MBN Explorer", Springer**

1.1 Potencjał Brennera

Potencjał Brennera generalnie opisuje oddziaływanie dwuciałowe (podobnie jak potencjał Lennarda-Jonesa), ale zawiera także informację o liczbie aktualnie utworzonych wiązań po to aby ich ilość nie przekraczała 4. Całkowitą energię potencjalną układu opisuje wyrażenie

$$V_{tot} = \frac{1}{2} \sum_{i=1}^n V_i \quad (1)$$

gdzie V_i to energia oddziaływania i -tego atomu z najbliższymi sąsiadami

$$V_i = \sum_{\substack{j=1 \\ j \neq i}}^n f_{cut}(r_{ij}) \left[V_R(r_{ij}) - \bar{B}_{ij} V_A(r_{ij}) \right] \quad (2)$$

Funkcja $f_{cut}(r)$ ogranicza zasięg przestrzenny tylko do lokalnego otoczenia atomu

$$f_{cut}(r) = \begin{cases} 1, & r \leq R_1 \\ \frac{1}{2} \left[1 + \cos \left(\frac{r-R_1}{R_2-R_1} \pi \right) \right], & R_1 < r \leq R_2 \\ 0, & r > R_2 \end{cases} \quad (3)$$

Wyraz $U_R(r)$ określa wielkość potencjału odpychania dwóch atomów, a wyraz $U_A(r)$ to energia ich przyciągania

$$V_R(r) = \frac{D_e}{S-1} \exp \left[-\sqrt{2S} \lambda (r - R_0) \right] \quad (4)$$

$$V_A(r) = \frac{D_e S}{S-1} \exp \left[-\sqrt{\frac{2}{S}} \lambda (r - R_0) \right] \quad (5)$$

Czynnik skalujący potencjał przyciągania \bar{B}_{ij} liczymy jako średnią wyrazów opisujących krotność wiązania

$$\bar{B}_{ij} = \frac{B_{ij} + B_{ji}}{2} \quad (6)$$

Wyraz B_{ij} obliczamy następująco

$$B_{ij} = (1 + \zeta_{ij})^{-\delta} \quad (7)$$

$$\zeta_{ij} = \sum_{\substack{k=1 \\ k \neq i, j}}^n f_{cut}(r_{ik}) g(\theta_{ijk}) \quad (8)$$

$$g(\theta_{ijk}) = a_0 \left[1 + \frac{c_0^2}{d_0^2} - \frac{c_0^2}{d_0^2 + (1 + \cos \theta_{ijk})^2} \right] \quad (9)$$

B_{ji} liczymy identycznie ale odwracamy wskaźniki. θ_{ijk} to kąt pomiędzy wektorami $\vec{r}_{ij} = \vec{r}_j - \vec{r}_i$ oraz $\vec{r}_{ik} = \vec{r}_k - \vec{r}_i$, który liczymy następująco

$$\cos \theta_{ijk} = \frac{\vec{r}_{ij} \cdot \vec{r}_{ik}}{r_{ij} r_{ik}} \quad (10)$$

$$\begin{aligned}
 R_0 &= 1.315 \text{ \AA} \\
 R_1 &= 1.70 \text{ \AA} \\
 R_2 &= 2.00 \text{ \AA} \\
 D_e &= 6.325 \text{ eV} \\
 S &= 1.29 \\
 \lambda &= 1.5 \text{ \AA}^{-1} \\
 \delta &= 0.80469 \\
 a_0 &= 0.011304 \\
 c_0 &= 19 \\
 d_0 &= 2.5
 \end{aligned}$$

1.2 Funkcja korelacji par (Pair Correlation Function)

W modelowaniu molekularnym często stosuje się w przypadku płynów tzw. funkcję korelacji par, która określa gęstość prawdopodobieństwa znalezienia innego atomu w odległości r od atomu położonego w początku układu współrzędnych. Mimo iż nasz problem dotyczy formowania się wieloatomowej cząsteczki, użyjemy tej funkcji do zwizualizowania zmiany korelacji przestrzennych pomiędzy atomami w trakcie symulacji. Wzór opisujący funkcję (PCF)

$$PCF(r) = \frac{\frac{2\Omega}{n^2} \sum_{i=1}^n \sum_{j>i}^n \delta(r - r_{ij})}{d\Omega} \quad (11)$$

gdzie: $\delta(r)$ to delta Diraca, Ω to objętość obszaru symulacji. Nasz problem jest w zasadzie dwuwymiarowy więc powierzchnia sfery/fullerenu (r_{sr} - średnia odległość atomów od środka układu)

$$\Omega = 4\pi r_{sr}^2 \quad (12)$$

natomiast element $d\Omega$ wyraża pole powierzchni w pierścieniu o promieniu r i szerokości dr

$$d\Omega = 2\pi r dr \quad (13)$$

Wzór (11) jest dobry do rozważań teoretycznych, ale w symulacjach musimy pozbyć się delty Diraca i PCF zdefiniować w postaci histogramu. W tym celu określamy średnią odległość atomów od środka

$$r_{sr} = \frac{1}{n} \sum_{i=1}^n r_i \quad (14)$$

maksymalny zakres histogramu (odległość między dwoma atomami)

$$r_{max} = 2.5 \cdot r_{sr} \quad (15)$$

ilość podprzedziałów M w histogramie i ich szerokość (niech będzie stała)

$$\Delta r = \frac{r_{max}}{M} \quad (16)$$

Teraz funkcję PCF możemy wyrazić w postaci tablicy 1D, której m -ty element obliczamy następująco

$$pcf[m] = \frac{2\Omega}{n^2} \frac{\sum_{i=1}^n \sum_{j>i}^n \delta_{m,k}}{2\pi r_m \Delta r} \quad (17)$$

$$\delta_{m,k} = \begin{cases} 1, & \text{gdy } k = m \\ 0, & \text{gdy } m \neq k \end{cases} \quad (18)$$

$$k = \text{floor} \left(\frac{r}{\Delta r} \right) \quad (19)$$

$$r_m = \left(m + \frac{1}{2} \right) \Delta r, \quad m = 0, 1, 2, \dots, M - 1 \quad (20)$$

$$(21)$$

Algorytm wyznaczania funkcji korelacji par

definiujemy: M , $pcf[M]$

$$r_{sr} = \frac{1}{n} \sum_i r_i$$

$$r_{max} = 2.5 \cdot r_{sr}$$

$$\Delta r = \frac{r_{max}}{M}$$

```
for(i=1; i<=n; i++){
  for(j=i+1; j<=n; j++){
    r = rij
    m = floor( r / Δr )
    if(m < M) pcf[m] = pcf[m] + 2·4πrsr2 / (n2·2πr·Δr)
  }
}
```

Uwaga: w algorytmie dodajemy zabezpieczenie w postaci instrukcji $if(m < M)$ ponieważ nie zawsze nasz fulleren będzie sferyczny, jeśli jakaś grupa atomów bardziej oddali się od środka układu wówczas może pojawić się wartość $m > M$ co spowoduje wyjście poza tablicę (często określaną jako *segmentation fault*).

1.3 Algorytm symulowanego wyżarzania

1.3.1 Położenia atomów

Interesują nas struktury zbliżone kształtem do sfery, więc położenia atomów (i przesunięcia) wygodnie opisywać używając współrzędnych sferycznych (r, ϕ, θ - kąt θ liczymy od osi z), a do liczenia odległości (potencjał oddziaływania) współrzędnych kartezjańskich (x, y, z). Dane całego układu (klastera) trzymamy w tablicy, aby były łatwo dostępne

r_1	ϕ_1	θ_1	x_1	y_1	z_1	(22)
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	
r_n	ϕ_n	θ_n	x_n	y_n	z_n	

Jeśli przesuniemy atom to modyfikujemy cały wiersz. Transformacja: sferyczne \rightarrow kartezjańskie

$$x_i = r_i \sin \theta_i \cos \phi_i \quad (23)$$

$$y_i = r_i \sin \theta_i \sin \phi_i \quad (24)$$

$$z_i = r_i \cos \theta_i \quad (25)$$

1.3.2 Losowe przesunięcia atomów

W metodzie SA jedna iteracja polega na próbie losowej zmiany położenia wszystkich atomów w klastrze. W celu określenia prawdopodobieństwa akceptacji nowego położenia i -tego atomu wykorzystamy algorytm Metropolisisa

wybieramy indeks cząstki: i

$$U_1, U_2, U_3 \sim U(0, 1)$$

$$\Delta r = r_i (2U_1 - 1) w_r$$

$$\Delta \phi = \phi_i (2U_2 - 1) w_\phi$$

$$\Delta \theta = \theta_i (2U_3 - 1) w_\theta$$

$$r_i^{new} = r_i + \Delta r$$

$$\phi_i^{new} = \phi_i + \Delta \phi$$

$$\theta_i^{new} = \theta_i + \Delta \theta$$

$$\text{if } (\phi_i^{new} < 0) \phi_i^{new} \leftarrow \phi_i^{new} + 2\pi$$

$$\text{if } (\phi_i^{new} > 2\pi) \phi_i^{new} \leftarrow \phi_i^{new} - 2\pi$$

$$\text{if } (\theta_i^{new} < 0) \theta_i^{new} \leftarrow \theta_i$$

$$\text{if } (\theta_i^{new} > \pi) \theta_i^{new} \leftarrow \theta_i$$

$$\vec{r}_i^{new} = [x_i^{new}, y_i^{new}, z_i^{new}]$$

liczymy starą i nową wartość potencjału atomu: V_i^{old}, V_i^{new}

$$p_{acc} = \min \left\{ 1, \exp \left[-\beta (V_i^{new} - V_i^{old}) \right] \right\}$$

$$U_4 \sim U(0, 1)$$

if ($U_4 \leq p_{acc}$) {

akceptujemy nowe położenie: $r_i \leftarrow \vec{r}_i^{new}$

}else{

nie akceptujemy przesunięcia - atom pozostaje w starym położeniu

}

Powyższy algorytm jest znacznym uproszczeniem, ponieważ po zmianie położenia atomu sprawdzamy tylko zmianę jego energii, wiadomo jednak że może to wpłynąć na energie sąsiadów poprzez czynnik \bar{B}_{ij} . Sprawdzanie zmian w energii sąsiadów znacznie wydłużyłoby jednak proces wyżarzania, czego chcemy uniknąć, więc użyjemy algorytmu SA w obecnej wersji.

1.3.3 Losowa (globalna) zmiana promienia sfery

Ponieważ nie wiemy jaki promień powinna mieć sfera na której rozłożone będą atomy, więc promień startowy inicjalizujemy małą wartością. Indywidualne zmiany **współrzędnej radialnej** r_i są mało wydajne, lepiej jest w każdej iteracji spróbować zmienić ją dla wszystkich cząstek jednocześnie - jeśli nowa konfiguracja będzie miała mniejszą energię to zostanie ona zaakceptowana. Szybkość zmian kontrolujemy parametrem W_{all} . Tu też skorzystamy z algorytmu Metropolisisa

inicjalizacja: $W_{all} \ll 1$

$$U_1 \sim U(0, 1)$$

for ($i=1$; $i \leq n$; $i++$) {

$$r_i^{new} = r_i [1 + W_{all} \cdot (2U_1 - 1)]$$

}

$$E_{tot}^{old} = E_{tot}(\vec{r}_1, \vec{r}_2, \dots, \vec{r}_n)$$

$$E_{tot}^{new} = E_{tot}(\vec{r}_1^{new}, \vec{r}_2^{new}, \dots, \vec{r}_n^{new})$$

```

pacc = min { 1, exp [ -β(Etotnew - Etotold) ] }
U2 ~ U(0,1)
if ( U2 ≤ pacc ) {
    ri ← rinew,   i = 1,2,...,n - akceptujemy nowe wartości radialne
} else {
    ri ← ri,   i = 1,2,...,n - pozostawiamy stare bez zmian
}

```

1.3.4 Parametr β

W klasycznej metodzie parametr β jest związany z temperaturą $\beta = 1/(kT)$. Mała wartość β oznacza duże prawdopodobieństwo akceptacji próbných przesunięć atomów, a duża wartość znacząco je ogranicza. Proces doboru wartości β możemy zautomatyzować wiążąc go z numerem iteracji

$$\beta = \beta_{min} + \left(\frac{it}{it_{max}} \right)^p (\beta_{max} - \beta_{min}) \quad (26)$$

gdzie: it - aktualny numer iteracji, it_{max} - maksymalna liczba iteracji, β_{min} i β_{max} to wartości minimalna i maksymalna parametru, $p > 0$ - wykładnik określający szybkość zmiany wartości β .

1.3.5 Algorytm SA

Mamy potrzebne informacje dotyczące poszczególnych elementów algorytmu więc możemy teraz zapisać sam algorytm SA

```

inicjalizacja: n, itmax, m, wr, wφ, wθ, Wall
               losowe położenia  $\vec{r}_1, \dots, \vec{r}_n$ 

for(int it=1; it<=itmax; it++){
    oblicz β(it) = wzór(26)

    for(int i=1; i<=n; i++){
        dokonaj próby zmiany położenia atomu i-tego (rozdział 1.3.2)
    }

    dokonaj próby globalnej zmiany promienia sfery (rozdział 1.3.3)

    if(it%100 == 0){
        wyświetl wartość Vtot
        zapisz do pliku:
                               aktualną konfigurację wiązań
                               tablicę pcf
    }
}

```

Uwaga: w algorytmie co 100 iteracji (przykładowo) zapisujemy aktualne dane do plików w celu szybkiej weryfikacji struktury przestrzennej. Jeśli w trakcie wykonywania symulacji zauważymy, że konfiguracja położenia atomów zaczyna "znacznie" odbiegać od naszych oczekiwań tj. od kształtu sferycznego, wówczas należy zatrzymać obliczenia, zmienić parametry startowe i wykonać symulację jeszcze raz.

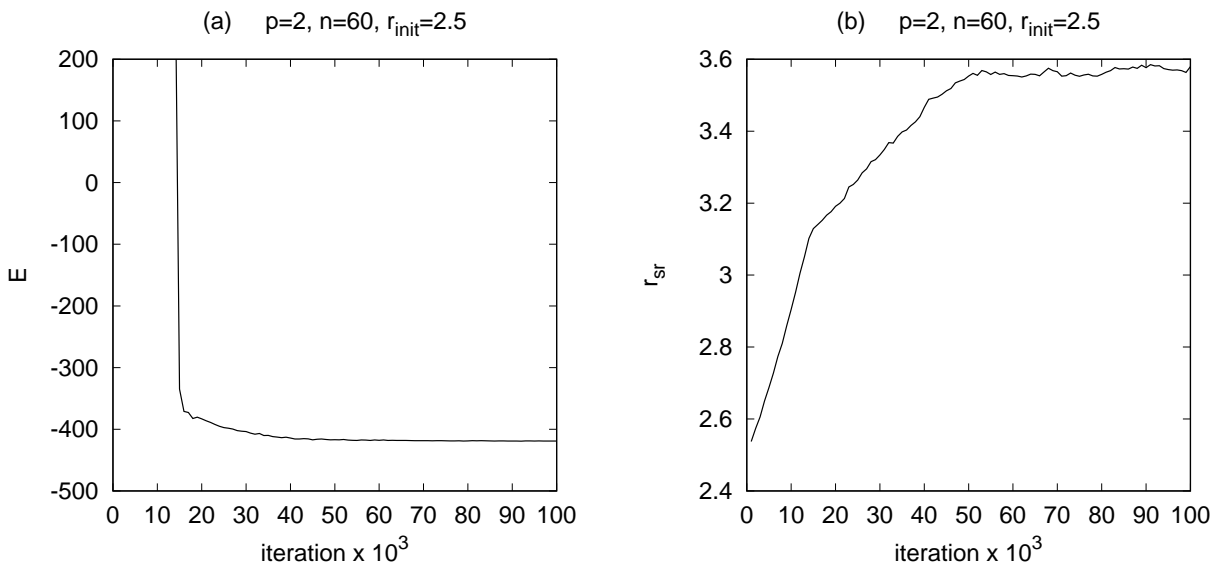
2 Zadania do wykonania

1. Zaprogramować metodę symulowanego wyżarzania z potencjałem Brennera.
2. Wykonać test sprawdzający poprawność liczenia potencjałów, w tym celu wczytać położenia atomów fullereny C_{60} ($n = 60$) z pliku "atoms_positions.c60.dat" (format danych: x_i, y_i, z_i) i obliczyć energię układu, powinna wynosić $V_{tot} = -421.6 eV$ co daje energię wiązania na atom równą $E_b = V_{tot}/n = 7.027 eV$. Odległości wszystkich atomów od środka układu są identyczne i wynoszą $r_i = r_{sr} = 3.52 \text{Å}$ Narysować strukturę przestrzenną fullereny C_{60} (np. w sposób opisany w sekcji 3)
3. Wykonać symulację dla parametrów: $n = 60$, $\beta_{min} = 1.0$, $\beta_{max} = 100$, wykładnik $p = 2$, $it_{max} = 10^5$, $w_r = 10^{-4}$, $w_\phi = 0.05$, $w_\theta = 0.05$, $W_{all} = 10^{-4}$. Jako położenia startowe atomów przyjąć: $r_i = 3.5 \text{Å}$, kąty $\phi_i \in (0, 2\pi)$ i $\theta_i \in (0, \pi)$ wylosować z rozkładu jednorodnego. Narysować uzyskaną strukturę oraz histogram pcf dla $M = 100$ podprzedziałów. Ile wiązań może utworzyć atom węgla? Narysować wykres zmian energii w trakcie symulacji wraz ze zmianami parametru β . Jaką uzyskamy energię układu i jaki r_{sr} ?
4. W zwykłej postaci potencjał Brennera preferuje tworzenie przez atom 4 wiązań z najbliższymi sąsiadami, tymczasem w C_{60} atomy mają tylko po 3 sąsiadów - musimy mu zabronić tworzenia 4 wiązań i pozwolić na tworzenie 3. W tym celu dokonujemy prostej modyfikacji parametru ζ_{ij} w procedurze liczącej współczynnik B_{ij} (wzory 7 i 8)

$$\text{if } (\cos \theta_{ijk} > 0) \quad \zeta_{ij} = 10$$

Innymi słowy: wprowadzamy karę do potencjału przyciągania (zmniejszamy jego amplitudę), gdy wykryjemy że atom chce utworzyć dwa wiązania i kąt między nimi jest bliski $\pi/2$ - to dawałoby właśnie możliwość utworzenia 4 takich wiązań. Kara spowoduje odepchnięcie 4 sąsiada. Powtórzyć obliczenia dla $n = 60$ ze zmodyfikowanym potencjałem. Narysować strukturę i pcf oraz określić energię układu i średni promień r_{sr} . Końcowa energia powinna być bliska $V_{tot} \approx -415 eV$ (lub trochę niższa - przykład na rysunku 2). Narysować wykres zmian energii w trakcie symulacji wraz ze zmianami parametru β . Z wykresu pcf odczytać preferowaną odległość między najbliższymi sąsiadami.

5. Powtórzyć symulację ($n = 60$) ze zmodyfikowanym potencjałem startując od $r_i = 2.5 \text{Å}$. Jaka będzie końcowa średnia odległość atomów od środka układu?
6. Przeprowadzić symulacje ($n = 60$) dla innych wartości: β_{min} , β_{max} , p , w_r , w_ϕ , w_θ . Które kombinacje parametrów dadzą lepszy rezultat końcowy a które gorszy? Najlepiej jest porównywać energie wiązania na jeden atom (np. dla wybranego $n = 30, 40$ lub 60), ponieważ dla różnych n zmieniają się w niewielkim zakresie.
7. Wykonać serię symulacji dla $n = 30, 31, \dots, 60$, za każdym razem startując od losowego rozkładu atomów na sferze. Jako promień startowy przyjąć $r_i = 2.5$ - sfera w trakcie symulacji powinna automatycznie dostosować swoje rozmiary. Dla każdego przypadku obliczyć energię wiązania na jeden atom $E_b = V_{tot}/n$. Wykonać wykres E_b w funkcji n . Energia wiązania jest miarą stabilności układu, czy na podstawie wykresu $E_b(n)$ można określić dla jakich n fullereny będą bardziej stabilne od pozostałych?



Rysunek 2: Przykładowe wyniki: (a) zmiany energii, (b) średni promień klastra dla parametrów $n = 60$, $r_{init} = 2.5$, $\beta_{min} = 1.0$, $\beta_{max} = 100$, $p = 2$, $it_{max} = 10^5$, $w_r = 10^{-2}$, $w_\phi = 0.01$, $w_\theta = 0.01$, $W_{all} = 10^{-4}$.

3 Informacje dodatkowe

3.1 Wizualizacja struktury przestrzennej fullereu w Gnuplocie

3.2 Metoda wyszukiwania i rysowania zamkniętych wielokątów - zalecana

W Gnuplocie istnieje możliwość rysowania wielokątów w 3D, co jest dla nas bardzo przydatne, gdyż to z nich jest zbudowana powierzchnia fullereu. Do znalezienia wielokątów można wykorzystać funkcję napisaną w `C++`

```
void write_polygons_from_atoms( double rmax,
                               int n,
                               vector<vector<double>> atom,
                               const char * plik )
```

Należy jej przekazać:

- $r_{max} \approx 1.3 \cdot r_{NN}$, gdzie r_{NN} jest odległością do najbliższego sąsiada odczytaną z histogramu PCF (wysokość pierwszego piku)
- n to liczba atomów
- tablica 2D $atom$ zawiera położenia (x_i, y_i, z_i) wszystkich n atomów
- $plik$ to nazwa pliku do którego zostaną zapisane wielokąty

Do pliku zostaną wpisane bloki danych zawierające położenia wierzchołków wielokątów, bloki oddzielone są dwiema pustymi liniami aby można było z danych wydobyć informacje dotyczące pojedynczego wielokąta. Rysunek klastra w Gnuplocie wykonujemy przy użyciu poniższych instrukcji

```
set xyplane 0
set view equal xyz
set pm3d depthorder border lw 2
set style fill transparent solid 0.3
splot 'plik' u 1:2:3 w polygons fc "gold"
```

Jeśli interesuje nas pojedynczy wielokąt to używamy opcji **index nr** do wyselekcjonowania bloku danych, bloki danych indeksowane są od 0, np.:

```
splot 'plik' index 0 u 1:2:3 w polygons fc "gold"
```

3.3 poprzednia metoda, prymitywna i pracochłonna - nie zalecana

Aby pokazać strukturę przestrzenną fullereny należy narysować sieć połączeń między najbliższymi sąsiadami - wiązania między atomami węgla. Odległość pomiędzy najbliższymi sąsiadami zawiera się w granicach $r_{NN} = 1.4 - 1.5\text{\AA}$. Wartość r_{NN} można też odczytać z wykresu funkcji korelacji par $pcf[m]$ jako położenie najwyższego pików. Zatem należy wyszukać pary atomów, których odległość jest mniejsza niż $r_{bind} \approx 1.2 \cdot r_{NN} = 1.8$ i zapisać do pliku, poniżej algorytm

```
for(int i=1;i<=n;i++){
  for(int j=i+1;j<=n;j++){
    xi=dane[i][3];
    yi=dane[i][4];
    zi=dane[i][5];
    xj=dane[j][3];
    yj=dane[j][4];
    zj=dane[j][5];
    r = rij
    if(r<1.2*rNN){
      fprintf(fp,"%15.5E %15.5E %15.5E\n",xi,yi,zi); // początek wiązania
      fprintf(fp,"%15.5E %15.5E %15.5E\n",xj,yj,zj); // koniec wiązania
      fprintf(fp,"\n\n"); // pusta linia, aby Gnuplot nie łączył wiązania
    }
  }
}
```

Narysowanie struktury przestrzennej zbudowanej tylko z wiązań jest już proste, ale na rysunku nakładająby się 'przednia' i 'tylnia' ścianka obiektu. Aby tego uniknąć, należałoby wyrysować obiekt którego ścianki byłyby zdefiniowane jako wielokąty. Automatyczne wykrycie krawędzi pięcio- i sześciokątów jest kłopotliwe (czasochłonne) więc ominiemy problem umieszczając w środku sferę która przesłoni 'tylną' ściankę. Trzeba tylko wygenerować dane sfery i zapisać do pliku, można to zrobić przy użyciu pomocniczego programu, algorytm jak poniżej

```
nf=50; //liczba wezłów w kierunku azymutalnym
nt=25; // liczba wezłów w kierunku biegunowym
r=3.3; // promień sfery (mniejszy niż promień fullereny)
```

```

for(int i=0;i<=nf;i++){
  for(int j=0;j<=nt;j++){
    fi=(2.*M_PI*i)/nf;
    teta=(M_PI*j)/nt;
    x=r*sin(teta)*cos(fi);
    y=r*sin(teta)*sin(fi);
    z=r*cos(teta);
    color=1.;
    fprintf(fp,"%15.5E   %15.5E   %15.5E   %15.5E   \n",x,y,z,color);
  }
  fprintf(fp,"\n"); // pusta linia dla Gnuplota - ułatwia gridowanie siatki
}

```

Mając pliki z danymi: wiązaniami oraz sferą, możemy użyć Gnuplota do narysowania struktury fullerenu

```

set view equal xyz          // ustawiamy identyczne skale na osiach
set hidden3d front         // ukrywamy to co jest za sferą
set style fill transparent solid 0.01 // skalujemy intensywność koloru sfery
unset key
unset colorbox
set view 76,359            // ustawiamy widok na obiekt

// wiązania+atomy+sfera w środku
splot 'cluster.dat' u 1:2:3 w lp ps 2 , 'sphere_3.3.dat' u 1:2:3:4 w pm3d

```