

Dynamika molekularna – sprawozdanie

Wybrany język: C++14

Program do wizualizacji: jmol

Program do sporządzenia wykresów: MATLAB R2018b

Struktura programu (stworzone klasy i funkcje):

1. argon.h:

- generator – statyczny silnik do generowania liczb pseudolosowych
- Vector – klasa reprezentująca wektor w trójwymiarowej przestrzeni; stworzono konstruktor domyślny, który tworzy obiekt o współrzędnych (0, 0, 0) oraz konstruktor z parametrami, przeciężono operatory: dodawania, przypisania dodawania, odejmowania, przypisania odejmowania oraz mnożenia i dzielenia przez skalar, oprócz tego stworzono metodę vectorModule, która zwraca długość wektora oraz gettery i settery pól klasy
- Parameters – klasa zawierająca dane o eksperymencie wczytane z pliku, stworzono konstruktor z parametrem przyjmującym nazwę pliku z danymi, dane z pliku są przypisywane polom klasy; klasa jest zaprzyjaźniona z klasą State, aby klasa State miała dostęp do pól prywatnych klasy Parameters
- State – klasa zawierająca informacje o stanie układu: energii potencjalnej, energii całkowitej, temperaturze i ciśnieniu, utworzone zostały dynamiczne tablice obiektów klasy Vector reprezentujące położenia, pędy i siły poszczególnych atomów w układzie, stworzono metody: ComputeInitialPositions, ComputeInitialMomenta, ComputeForces, UpdateState, SavePositions i SaveData.

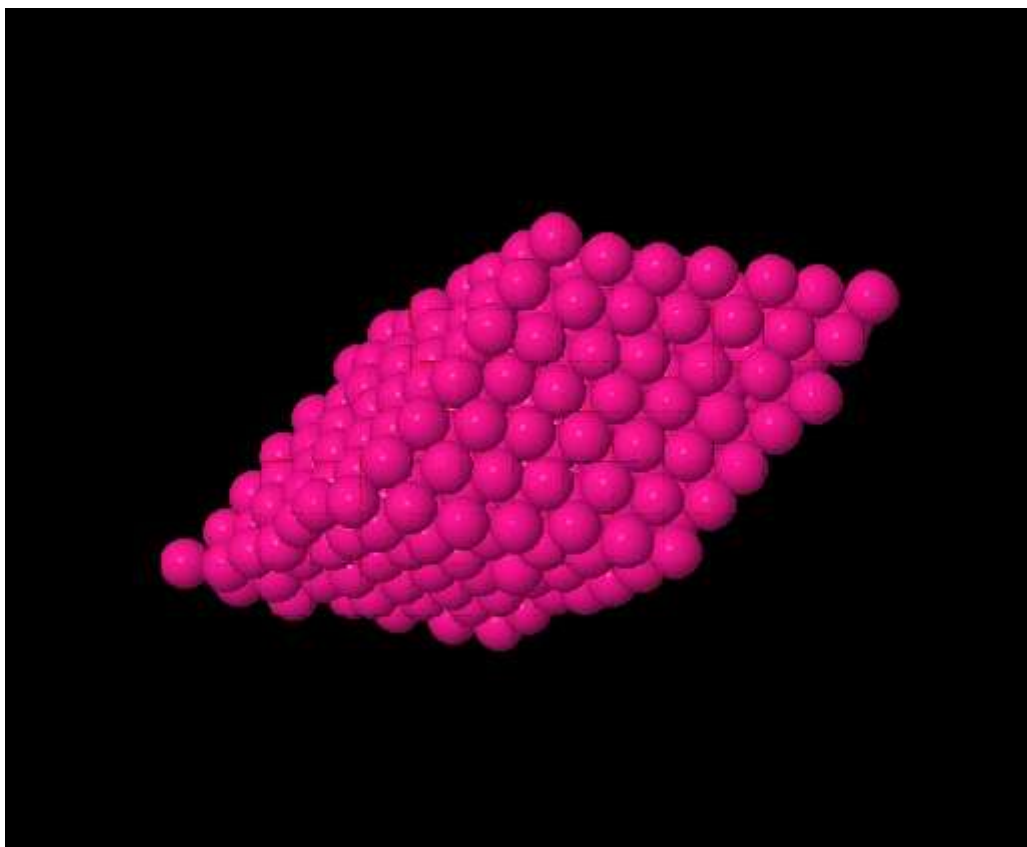
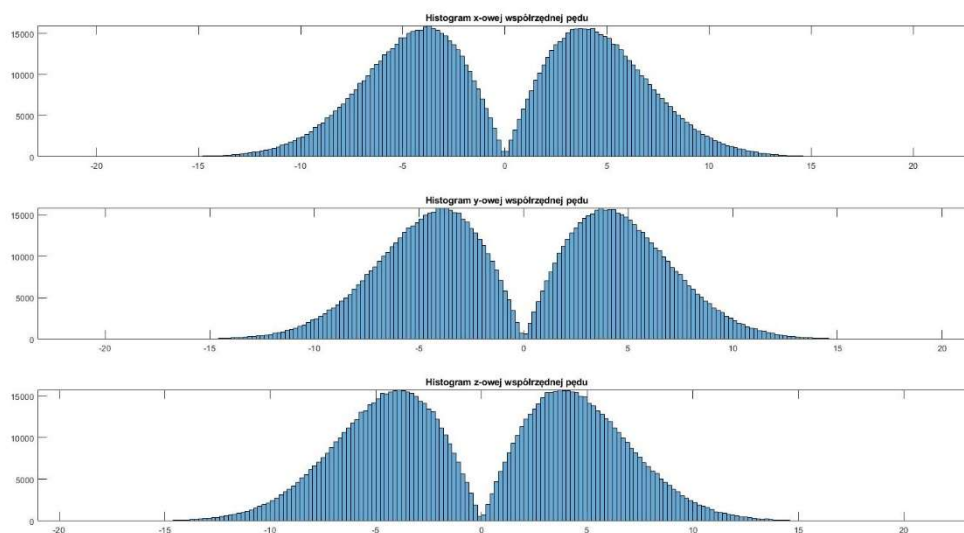
2. argon.cpp:

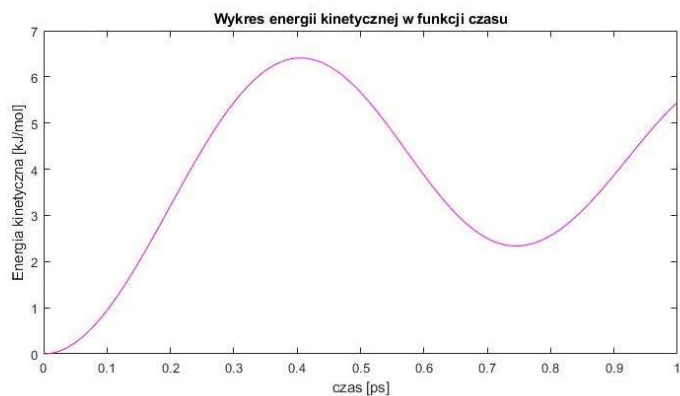
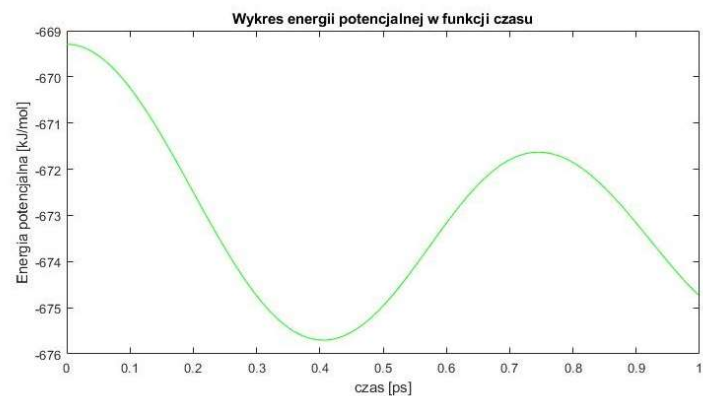
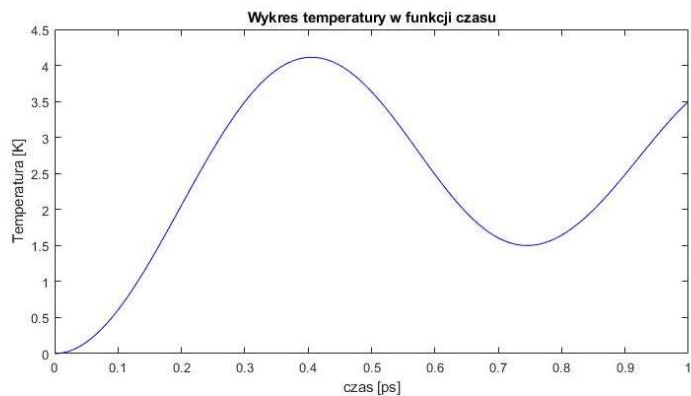
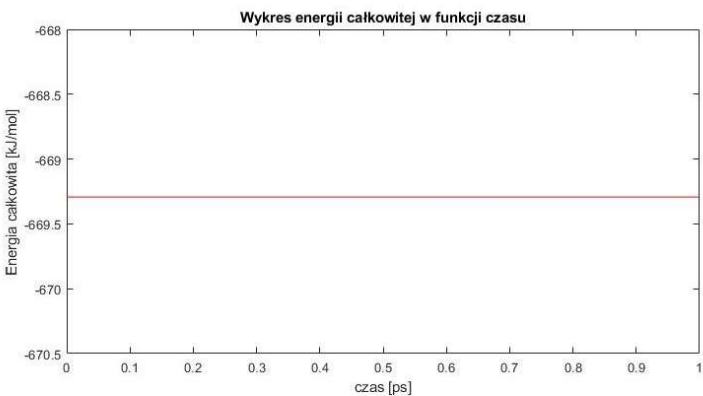
- ComputeInitialPositions – metoda obliczająca początkowe położenia atomów w kryształach
- ComputeInitialMomenta – metoda obliczająca początkowe pędy atomów, numberDistribution losuje liczby z rozkładu jednostajnego od 0 do 1 potrzebne do obliczenia energii kinetycznej, a signDistribution losuje liczby z rozkładu jednostajnego od -1 do 1, jeżeli wylosowana liczba jest mniejsza od 0 to liczba przyjmuje wartość -1, w przeciwnym wypadku przyjmuje wartość 1, co jest potrzebne do określenia znaku pędów
- ComputeForces – metoda oblicza siły van der Waalsa działające między atomami kryształu oraz siły między atomami a ścianką naczynia, oprócz tego obliczany jest też potencjał i ciśnienie układu
- UpdateState – metoda wykonująca krok czasowy i licząca kolejne położenia, pędy i siły atomów, oprócz tego obliczana jest temperatura i całkowita energia układu
- SavePositions – metoda zapisująca położenia atomów w każdym kroku do pliku
- SaveData – metoda zapisująca stan układu w każdym kroku do pliku
- Main – główna funkcja programu, na początku tworzone są obiekty klas Parameters i State, obliczane są początkowe położenia, pędy i siły, otwierane są pliki do zapisu danych, następnie w pętli aktualizowany jest stan układu i następuje zapis danych, na końcu obliczane są średnie parametry układu, które zostają wypisane na st. Wyjście

Poprawność symulacji stwierdzono na podstawie wartości pędów, które spełniają rozkład Maxwella oraz stałej wartości energii całkowitej, biorąc po uwagę fakt, że symulowany jest układ mikrokanoniczny.

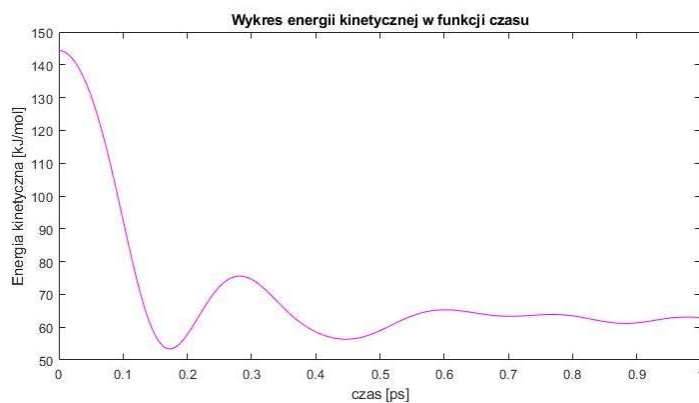
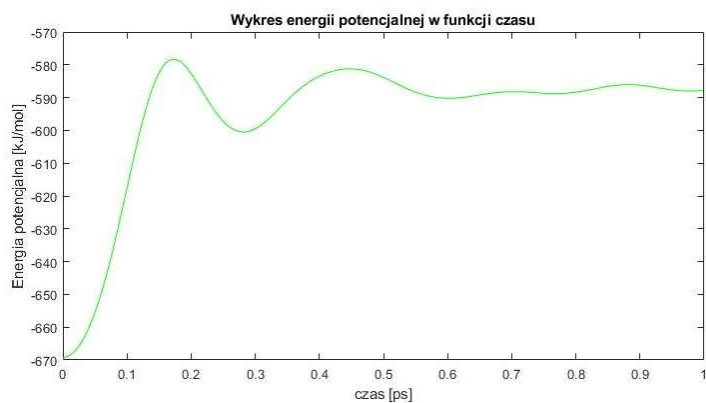
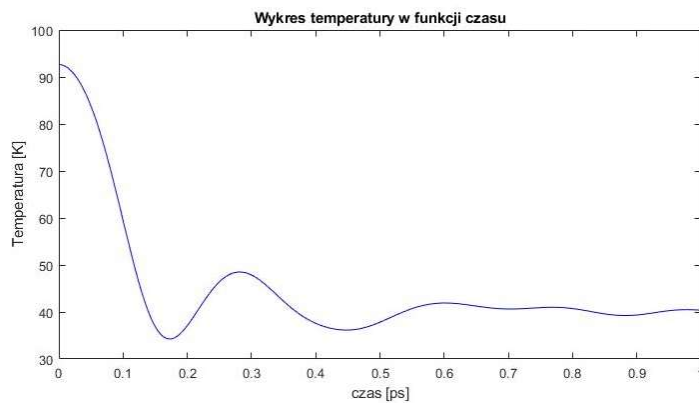
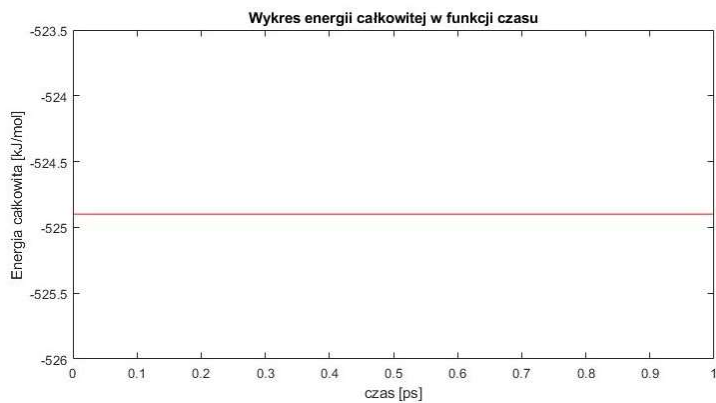
Parametry układu z symulacji zapisywane są w pliku state.dat i na ich podstawie utworzone zostały wykresy, natomiast położenia atomów zapisywane są w pliku avs.dat i wczytuje się je do programu jmol, aby wizualizować zachowanie kryształu w symulacji.

Temperatura topnienia kryształu mieści się w temp. 80 – 90 K, co jest zgodne z rzeczywistością (83 K).

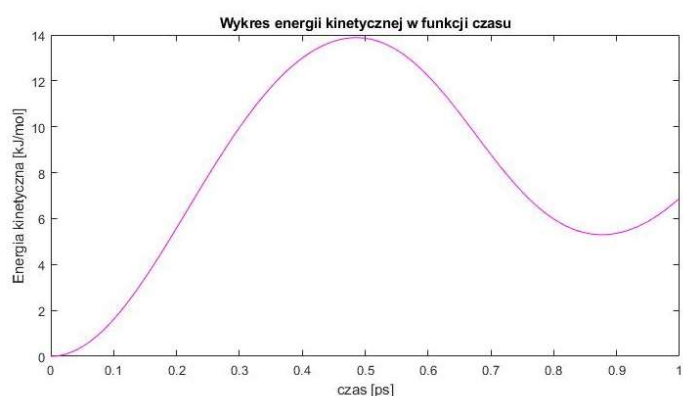
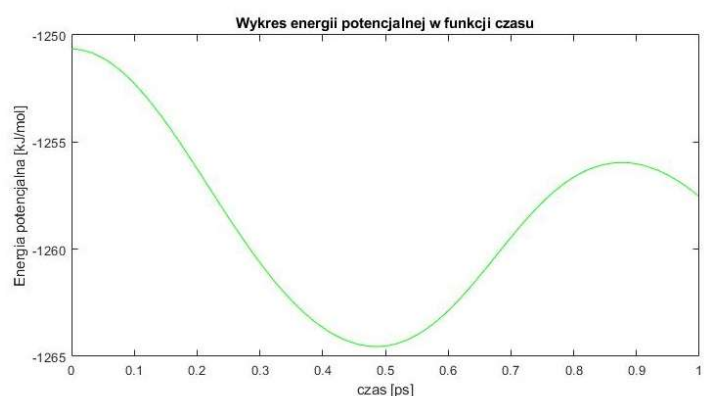
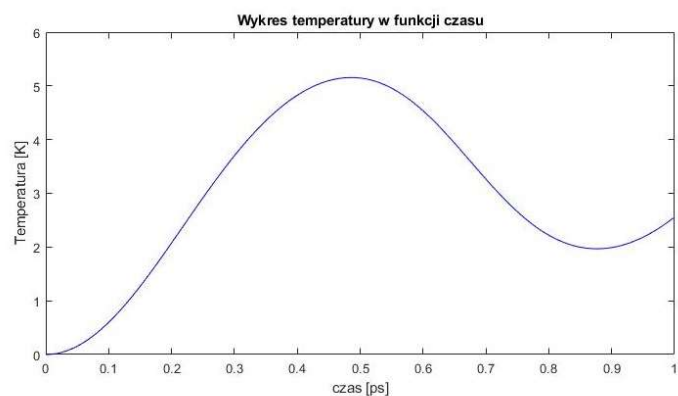
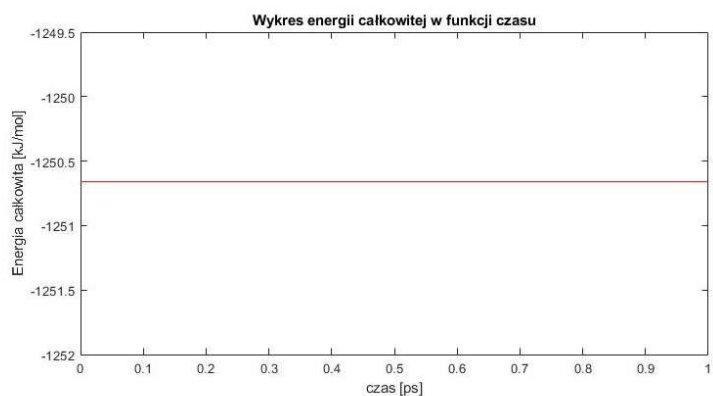




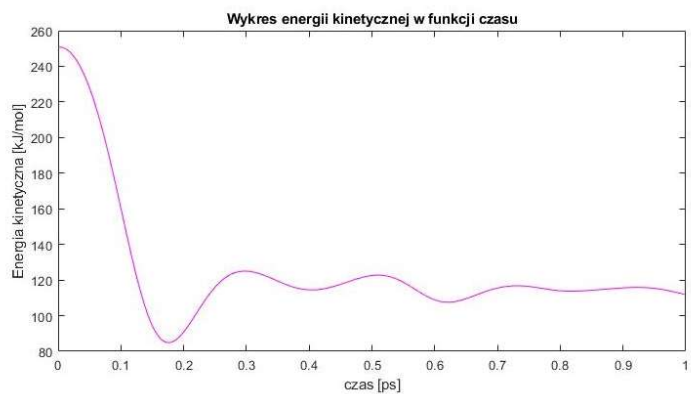
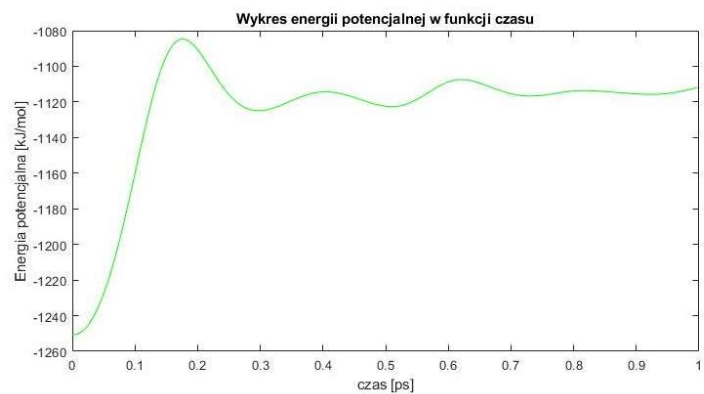
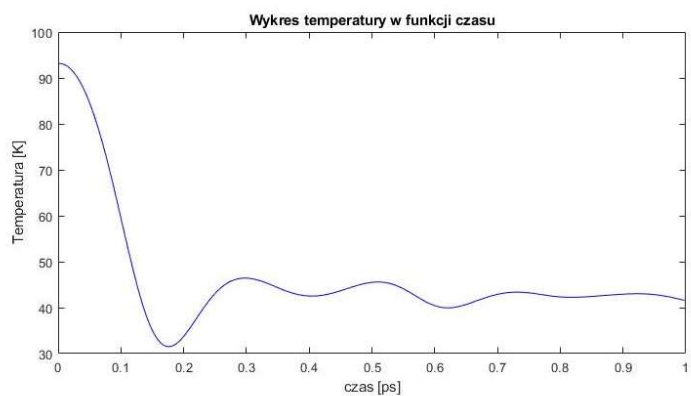
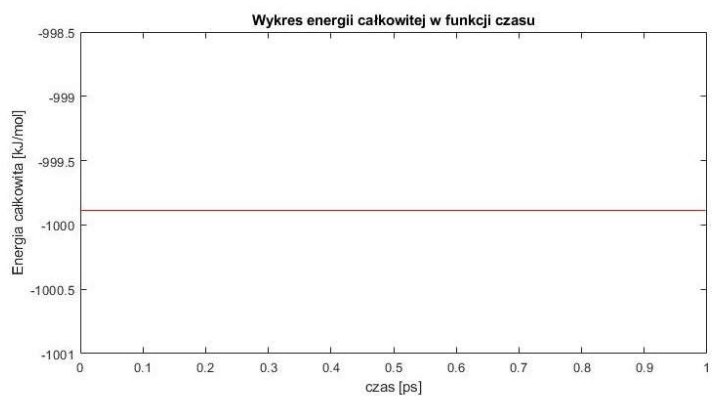
n = 5, T = 90 K



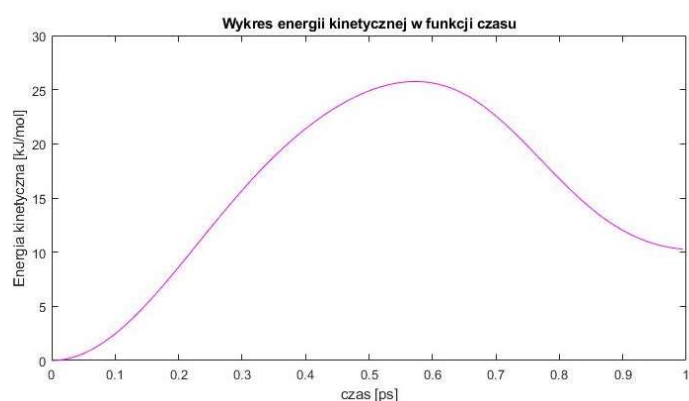
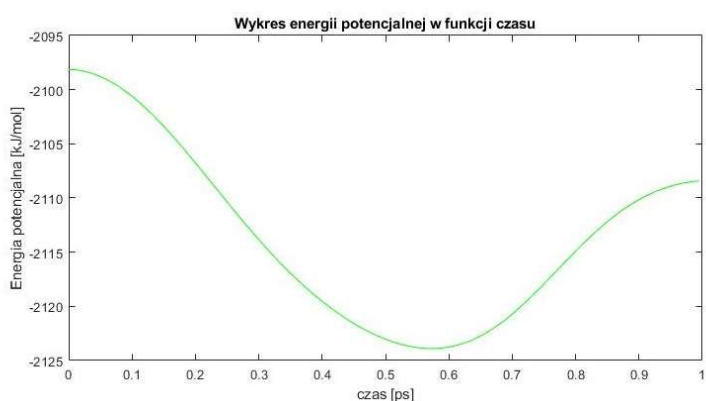
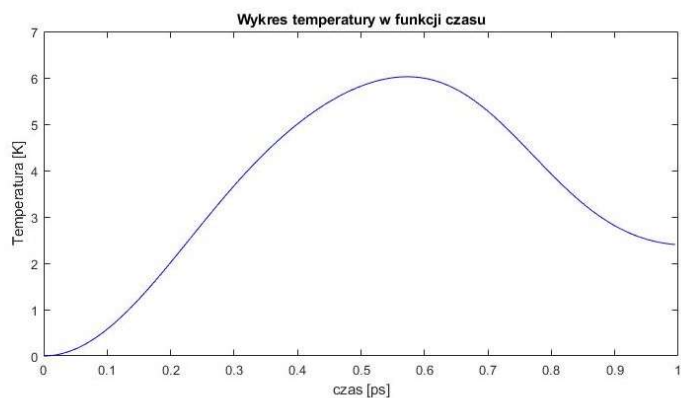
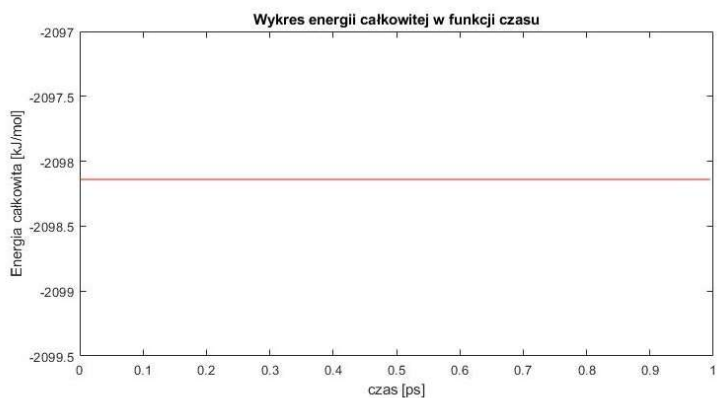
$n = 6, T = 0 \text{ K}$



$n = 6, T = 90 \text{ K}$



$n = 7, T = 0 \text{ K}$



$n = 7, T = 90 \text{ K}$

