

Wstęp do programowania w naukach przyrodniczych

Zadania opracował: dr hab. Dominik Gront¹
11 marca 2019

Część I

Zadania z programowania

1 Proste zadania

1.1 Suma szeregu

Napisz program obliczający 30 pierwszych wyrazów szeregu:

(a)

$$1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \frac{1}{5^2} + \dots + \frac{1}{n^2}$$

(b)

$$1 - \frac{1}{2^2} + \frac{1}{3^2} - \frac{1}{4^2} + \frac{1}{5^2} + \dots + (-1)^{n-1} \frac{1}{n^2}$$

(c)

$$1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \frac{1}{4!} + \dots + (-1)^n \frac{1}{n!}$$

(d)

$$\frac{1}{1 \cdot 2 \cdot 3} + \frac{1}{2 \cdot 3 \cdot 4} + \frac{1}{3 \cdot 4 \cdot 5} + \frac{1}{4 \cdot 5 \cdot 6} + \dots + \frac{1}{n \cdot (n+1) \cdot (n+2)}$$

(e)

$$1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots + \frac{x^n}{n!}$$

(f)

$$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \dots$$

(g)

$$1 + x \ln a + \frac{(x \ln a)^2}{2!} + \frac{(x \ln a)^3}{3!} + \frac{(x \ln a)^4}{4!} + \dots + \frac{(x \ln a)^n}{n!}$$

(h)

$$\frac{(x-1)}{x} + \frac{1}{2} \frac{(x-1)^2}{x^2} + \frac{1}{3} \frac{(x-1)^3}{x^3} + \dots + \frac{1}{n} \frac{(x-1)^n}{x^n}$$

Jeżeli potrzeba, zadeklaruj zmienną x i nadaj jej wartość 0.3.

¹dgront@chem.uw.edu.pl

1.2 Liczby Fibonacciego

Ciąg Fibonacciego składa się z liczb naturalnych określonych rekurencyjnie następująco:

$$F_n := \begin{cases} 0 & \text{dla } n = 0; \\ 1 & \text{dla } n = 1; \\ F_{n-1} + F_{n-2} & \text{dla } n > 1. \end{cases}$$

Napisz program, który wypisze na ekranie liczby Fibonacciego mniejsze od 100.

1.3 Schemat Hornera

Naiwna implementacja obliczania wartości wielomianu:

$$W(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n-1}x^{n-1} + a_nx^n$$

dla zadanej wartości x wymaga $n(n+1)/2$ mnożeń oraz n dodawań, gdzie n to stopień wielomianu. Stosując schemat Hornera, wykonamy jedynie n mnożeń oraz n dodawań, wystarczy jedynie nieco przekształcić wzór na wielomian:

$$W(x) = a_0 + x(a_1 + x(a_2 + \dots + x(a_{n-1} + xa_n) \dots))$$

(a) Napisz program, który dla zadanego x oblicza wartość dystrybuanty rozkładu normalnego $\mathcal{N}(0, 1)$, którą można zdefiniować w oparciu o tzw *funkcję błędu* erf:

$$F_X(x) = P(X \geq x) = 1 - \frac{1}{2} \operatorname{erf}\left(-\frac{x}{\sqrt{2}}\right)$$

Funkcja erf co prawda nie ma znanej postaci analitycznej, ale można ją przybliżyć za pomocą wielomianu², wyrażonego w postaci Hornera następująco:

$$\begin{aligned} \operatorname{erf}(x) &\approx 1 - (a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5)e^{-x^2} \\ &= 1 - (((((a_5t + a_4)t) + a_3)t + a_2)t + a_1)te^{-x^2} \end{aligned}$$

gdzie:

$$t = \frac{1}{1 + p * x}$$

a wartości współczynników p oraz a_i wynoszą:

$$\begin{array}{ll} a_1 = 0.25482 & a_2 = -0.28449 \\ a_3 = 1.42141 & a_4 = -1.45315 \\ a_5 = 1.06140 & p = 0.32759 \end{array}$$

(b) Napisz program, który wczytuje stopień wielomianu n oraz wartości jego $n + 1$ współczynników a następnie tablicuje wielomian w zadanym przedziale.

²Abramowitz and Stegun, „*Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*” (dostępna w Internecie) - wzór **7.1.26**

1.4 Układ równań

Napisz program rozwiązujący układ dwóch równań z podanymi współczynnikami. Program powinien pobierać 6 liczb: $a_1, b_1, c_1, a_2, b_2, c_2$ i drukować parę liczb x, y będącą rozwiązaniem układu:

$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases} \quad (1)$$

o ile takie rozwiązanie istnieje. W przeciwnym przypadku powinien wypisywać odpowiedni komunikat.

1.5 Generator liczb losowych

Jednym z prostszych sposobów generowania liczb (pseudo)losowych³ z rozkładu płaskiego to obliczanie reszty z dzielenia (operator *modulo*). Generator taki nazywa się liniowym (Linear Congruential Generators - LCG) i definiuje się następująco:

$$x_{i+1} = (ax_i + c) \bmod m \quad (2)$$

gdzie x_i to i -ta liczba pseudolosowa, a - mnożnik, c - przyrost, m - moduł. Napisz program, generujący 30 liczb pseudolosowych, korzystając z powyższego wzoru. Przyjmij następujące wartości parametrów:

a) $m = 233280$ $a = 9301$ $c = 49297$ $x \in [0 \dots 233279]$

b) $m = 714025$ $a = 4096$ $c = 150889$ $x \in [0 \dots 714024]$

Wartość pierwszej liczby losowej (konieczna do wygenerowania kolejnej wartości) może być dowolną liczbą mniejszą od stałej m . Opisany tu generator zwraca liczby całkowite nieujemne. Aby uzyskać generator liczb rzeczywistych z przedziału $[0, 1)$, należy generowane wartości podzielić przez m .

1.6 Proste statystyki

Wygeneruj 1000 pseudolosowych liczb rzeczywistych (np. wykorzystując generator opisany w zadaniu 1.5) z przedziału $[0 \dots 1)$. Oblicz wartość średnią i odchylenie standardowe. Znajdź też wartość najmniejszą i największą.

1.7 Losowe punkty na kole

(a) Wylosuj 10 000 punktów na płaszczyźnie tak, że zarówno współrzędna x jak i y każdego z punktów pochodzi z przedziału $[-1, 1]$. Wydrukuj współrzędne tylko tych punktów, które zawierają się w okręgu o promieniu $r_0 = 1.0$.

³Przy ustalonych warunkach początkowych, np. stałe x_0, a, c oraz m we wzorze 2 wynik "losowania" jest zawsze taki sam, dlatego liczb tych nie można nazwać losowymi. Jednakże kolejne generowane liczby zachowują podstawowe właściwości liczb losowych

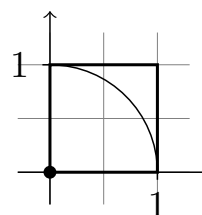
(b) Wylosuj współrzędne radialne (r, ϕ) punktów leżących na kole o promieniu $r_0 = 1.0$. Następnie przelicz te współrzędne na kartezjańskie korzystając z zależności:

$$\begin{cases} x = r \cos \phi \\ y = r \sin \phi \end{cases} \quad (3)$$

Porównaj wykresy punktów wylosowanych wg poleceń (a) i (b) i porównaj je.

1.8 Wyznaczanie wartości π metodą Monte Carlo

Metoda Monte Carlo umożliwia łatwe policzenie dowolnej całki, np. pola figury. Zauważ, że pole $\frac{1}{4}$ okręgu o promieniu $r = 1.0$ wynosi $\frac{\pi}{4}$. Losując punkty z kwadratu o boku 1.0 (czyli o współrzędnych x oraz $y \in [-1, 1]$) możemy trafić wewnątrz wycinka koła z prawdopodobieństwem równym tej wartości pola (czyli $\frac{\pi}{4}$).



Wylosuj N punktów, policz ile z nich trafiło do wnętrza koła i oblicz na tej podstawie wartość liczby π . Oszacuj w ten sposób π dla $N = 100, 1\ 000, 10\ 000$ oraz $100\ 000$.

1.9 Równanie kwadratowe

Program powinien pobierać trzy liczby: a, b, c i znajdować pierwiastki równania kwadratowego w postaci:

$$ax^2 + bx + c = 0 \quad (4)$$

Jeżeli wartość wyznacznika równania kwadratowego jest ujemna (tzn. $\Delta < 0$), program powinien wypisać odpowiedni komunikat.

1.10 Tablicowanie wartości funkcji dwóch zmiennych

Napisz program, drukujący trzy kolumny liczb: x, y oraz wartości funkcji:

$$f(x, y) = \sin(x + y) \frac{x}{x^2 + 1} \quad (5)$$

Wartości zmiennej x powinny zmieniać się od -2 do 2 co 0.1 a wartości zmiennej y powinny zmieniać się od -5 do 5 co 0.1 .

1.11 Tablicowanie pochodnej funkcji

Wartość pochodną dowolnej funkcji $f(x)$ w punkcie x_0 można oszacować wg poniższego wzoru:

$$f'(x_0) \sim \frac{f(x_0 + h) - f(x_0 - h)}{2h}$$

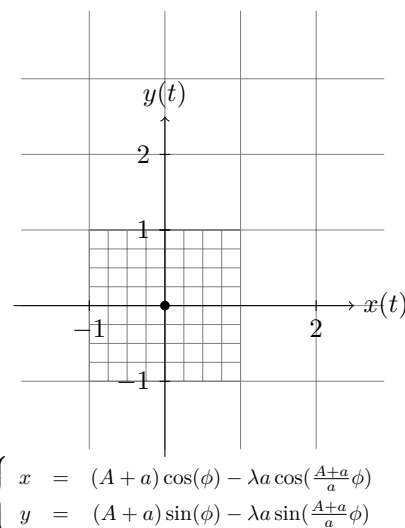
Napisz program, drukujący trzy kolumny liczb: x , $f(x)$ oraz pochodnej $f'(x)$ dla kilku prostych funkcji np $\sin(x)$ czy $\arctan(x)$.

1.12 Epitrochoida

Napisz program, tablicujący punkty leżące na epitrochoidzie. Krzywa ta zadana jest następującym równaniem parametrycznym:

$$\begin{cases} x = (A + a) \cos(\phi) - \lambda a \cos(\frac{A+a}{a}\phi) \\ y = (A + a) \sin(\phi) - \lambda a \sin(\frac{A+a}{a}\phi) \end{cases} \quad (6)$$

Wielkości A , a , oraz λ to dowolne parametry a ϕ to zmienna niezależna. Program powinien drukować trzy kolumny liczb: ϕ x y , gdzie ϕ zmienia się od 0.0 do 10.0 co 0.01 a x i y policzone są wg powyższych wzorów. Przykładowe wartości parametrów: $A = 1.0$, $a = 0.4$, $\lambda = 1.4$.



1.13 Model Lotki-Volterry

Stosując podany poniżej układ równań różniczkowych⁴ można opisać prosty ekosystem ofiar z (np. zajęcy) i drapieżników w (np. wilków) w funkcji czasu t :

$$\begin{cases} \frac{dz}{dt} = (\alpha - \beta w)z \\ \frac{dw}{dt} = (\gamma z - \delta)w \end{cases} \quad \begin{cases} z = z + \delta t(\alpha - \beta w)z \\ w = w + \delta t(\gamma z - \delta)w \end{cases}$$

gdzie α , β , γ i δ to parametry oznaczające:

α - współczynnik przyrostu ofiar

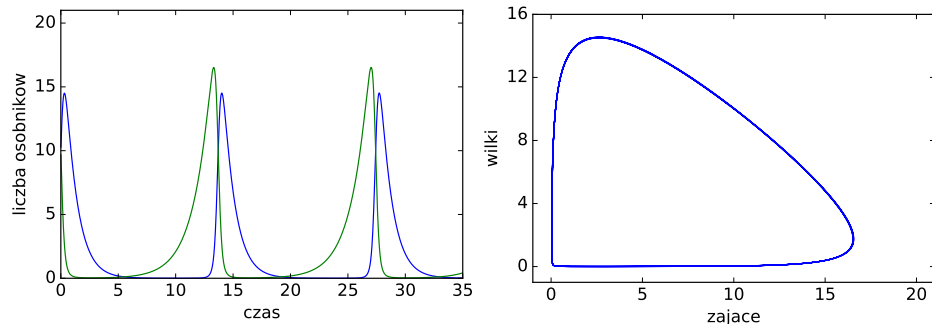
β - skuteczność drapieżników

γ - współczynnik przyrostu drapieżników

δ -współczynnik ubywania drapieżników

Zaimplementuj układ równań różnicowych (dyskretnych) podanych po prawej stronie i narysuj wykres liczby zajęcy oraz wilków w zależności od czasu (jak poniżej po lewej) a także wykres punktów (z, w) dla różnych t (czyli tzw. wykres fazowy - jak poniżej po prawej).

⁴opisany niezależnie przez Vito Volterrę oraz Alfreda Lotkę



Orientacyjne wartości parametrów: $\alpha = 0.66$, $\beta = 1.0$, $\gamma = 0.8$, $\delta = 1.0$, $w_0 = 10.0$, $z_0 = 10.0$; krok czasowy $\delta t = 0.0001$
Uwaga: Jeżeli krok czasowy jest zbyt duży, symulacja jest rozbieżna.

1.14 Zmienna losowa z rozkładu normalnego: Centralne Twierdzenie Graniczne

Losowanie z rozkładu normalnego zrealizować można jako wielokrotne losowanie z rozkładu jednostajnego⁵. Aby policzyć **jedną** wartość zmiennej losowej x_N z rozkładu $N(0.0, 1.0)$:

- wylosuj 12 liczb z rozkładu jednostajnego⁶ : $x_1, x_2, x_3, \dots, x_{12} \in [0.0, 1.0)$
- policz ich sumę $s = x_1 + x_2 + x_3 + \dots + x_{12}$
- $x_N = \frac{s-6}{12}$

Wydrukuj na ekranie takich 30 liczb losowych.

1.15 Losowanie z rozkładu normalnego: transformacja Boxa-Mullera i rozkład Maxwella

Inna metoda na losowanie z rozkładu normalnego polega na wylosowaniu dwóch niezależnych liczb losowych z rozkładu jednostajnego $x_1, x_2 \in [0.0, 1.0)$. Następnie stosuje się następującą transformację:

$$\begin{cases} n_1 = \sqrt{-2 \log x_1} \cos(2\pi x_2) \\ n_2 = \sqrt{-2 \log x_1} \sin(2\pi x_2) \end{cases} \quad (7)$$

w wyniku której otrzymuje się **dwie** losowo niezależne liczby n_1, n_2 z rozkładu normalnego o zerowej wartości oczekiwanej i jednostkowej wariancji. Aby zmienić parametry rozkładu na inne, np wartość czekiwaną na μ a wariancję na σ należy pomnożyć wynik losowania przez σ i dodać μ , tzn:

$$\mathcal{N}(\mu, \sigma) = \sigma \mathcal{N}(0, 1) + \mu$$

⁵Problem ten jest doskonałym przykładem na działanie Centralnego Twierdzenia Granicznego. Nie jest to jednak najlepszy generator liczb losowych $\mathcal{N}(0, 1)$

⁶im więcej liczb losowych uśrednimy, tym bardziej zbliżony do normalnego będzie nasz rozkład. 12 to wystarczająco dobre przybliżenie.

W powyższym równaniu $\mathcal{N}(\mu, \sigma)$ oznacza zmienną losową o rozkładzie normalnym z wartością oczekiwaną μ i wariancją σ .

Zgodnie z rozkładem Maxwella, każda składowa (x , y albo z) wektora prędkości \vec{v} jest zmienną losową z rozkładu normalnego o zerowej wartości oczekiwanej i o wariancji $\sqrt{\frac{k_B T}{m}}$, gdzie m to masa cząsteczki, T to temperatura w Kelvinach a k_B - stała Boltzmannna. Wylosuj 10 000 wektorów prędkości dla cząsteczki azotu N_2 w temperaturze 300K. Oblicz średnią długość tych wektorów, a więc średnią wartość prędkości cząsteczki azotu, korzystając ze wzoru:

$$\bar{v} = \frac{1}{N} \sum_{i=1}^N \sqrt{v_{x_i} * v_{x_i} + v_{y_i} * v_{y_i} + v_{z_i} * v_{z_i}}$$

1.16 Równanie kwadratowe

Program powinien pobierać trzy liczby: a , b , c i znajdować pierwiastki równania kwadratowego w postaci:

$$ax^2 + bx + c = 0 \tag{8}$$

Jeżeli wartość wyznacznika równania kwadratowego jest ujemna (tzn. $\Delta < 0$), program powinien wypisać odpowiedni komunikat.

1.17 Tablicowanie wartości funkcji dwóch zmiennych

Napisz program, drukujący trzy kolumny liczb: x , y oraz wartości funkcji:

$$f(x, y) = \sin(x + y) \frac{x}{x^2 + 1} \tag{9}$$

Wartości zmiennej x powinny zmieniać się od -2 do 2 co 0.1 a wartości zmiennej y powinny zmieniać się od -5 do 5 co 0.1 .

1.18 Tablicowanie pochodnej funkcji

Wartość pochodną dowolnej funkcji $f(x)$ w punkcie x_0 można oszacować wg poniższego wzoru:

$$f'(x_0) \sim \frac{f(x_0 + h) - f(x_0 - h)}{2h}$$

Napisz program, drukujący trzy kolumny liczb: x , $f(x)$ oraz pochodnej $f'(x)$ dla kilku prostych funkcji np $\sin(x)$ czy $\arctan(x)$.

1.19 Rysunki w terminalu tekstowym

Napisz program, wypisujący w terminalu poniższy kształt:

(a)	(b)	(c)
#	#	#
##	###	####
###	####	#####
####	#####	#####
#####	#####	#####

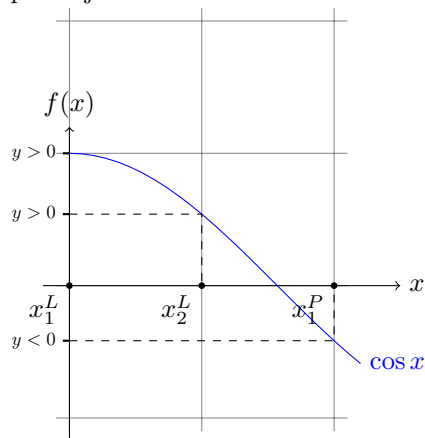
1.20 Liczby pierwsze

Napisz program znajdujący liczby pierwsze. Dla każdej sprawdzanej liczby naturalnej n program powinien sprawdzić, czy dzieli się ona bez reszty przez wszystkie liczby od 2 do $n - 1$ ⁷. *Wariant trudniejszy.* Program powinien zapamiętywać w tablicy już znalezione liczby pierwsze i sprawdzać podzielność tylko przez nie.

1.21 Metoda połowienia przedziału

Napisz program, który znajduje miejsce zerowe zadanej funkcji w określonym przedziale, np. $\cos(x)$ gdzie $x \in [0.0, 2.0]$. Program powinien wykonać zadaną liczbę (np. 30) iteracji. Krótki opis tej metody podano poniżej.

- Mamy daną funkcję $f(x)$ (tu: $\cos(x)$) i dwa punkty: lewy ($x_1^L = 0.0$) i prawy ($x_1^P = 2.0$). Musi być spełniony warunek: $f(x^L) \times f(x^P) < 0$
- W każdym i -tym kroku algorytmu wybieramy nowy punkt x_{i+1}^N leżący pośrodku punktu prawego i lewego. W tym przykładzie nowy punkt $x_2 = 1.0$ leży pomiędzy $x_1^L = 0.0$ i $x_1^P = 2.0$. Następnie wybieramy jedną z trzech możliwości:



- jeżeli $f(x_{i+1}^N) \times f(x_i^P) = 0$ to znaleziono miejsce zerowe w punkcie x_{i+1}^N .
- jeżeli $f(x_i^L) \times f(x_{i+1}^N) < 0$ to oznacza, że miejsce zerowe leży gdzieś pomiędzy punktem lewym a nowym. Wtedy x_i^L pozostaje punktem lewym w kolejnej iteracji jako x_{i+1}^L a x_{i+1}^N staje się nowym punktem prawym (x_{i+1}^P).
- jeżeli $f(x_{i+1}^N) \times f(x_i^P) < 0$ to wtedy x_{i+1}^N staje się punktem lewym (x_{i+1}^L) a x_i^P pozostaje punktem prawym w kolejnej iteracji jako x_{i+1}^P .

W tym przykładzie $\cos(1.0) \times \cos(2.0) < 0$ co oznacza, że miejsce zerowe leży pomiędzy punktem nowym a prawym. Punkt prawy zostaje jako prawy w iteracji nr 2, a za punkt lewy obierany jest x_1^N i połowienie przedziału powtarza się.

⁷ tak na prawdę wystarczy sprawdzić podzielność przez k takie, że $2 \leq k \leq \sqrt{n}$, prawda?

1.22 Równania nieliniowe - metoda Newtona

Inną metodą znajdowania miejsc zerowych funkcji (czyli rozwiązywania równań postaci $f(x) = 0$) jest metoda Newtona. Wymagana jest w tym przypadku znajomość pochodnej $f'(x)$ oraz punkt startowy x_0 niezbyt odległy od poszukiwanego miejsca zerowego. Miejsce zerowe znajdujemy stosując iterację:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Rozwiąż metodą Newtona następujące równania:

- (a) $x^2/4 + \sin(x) = 0$ przy $x_0 = 1.6$
- (b) $\ln x - e^{-3x} = 0$ przy $x_0 = 2$

2 Zadania trudniejsze

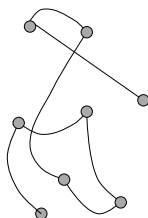
2.1 Epitrochoida - wariacje na temat

Zmodyfikuj program 1.9 tak, aby parametry A oraz a były funkcjami parametru ϕ , np:

$$\begin{aligned} A(\phi) &= A(1.0 + 0.25 \sin(\phi)) \\ a(\phi) &= a(1.0 + 0.5 \cdot |\sin(\phi)|) \end{aligned} \quad (10)$$

W programie zadeklaruj funkcje dla powyższych równań a wyniki zapisz w tabelicy.

2.2 Błądzenie losowe



Pchła startuje z punktu $(0,0,0)$ i wykonuje skoki o długości 1.0 w losowym kierunku. Zbadaj, na jaką odległość od punktu startu oddali się pchła po N skokach.

Wersja nieco trudniejsza ale ciekawsza: długość skoku pchły jest również losowa, np $\in (0...1]$

2.3 Szyfr Cezara (podstawieniowy)

Jednym z najprostszych sposobów szyfrowania wiadomości jest szyfr Cezara⁸. Polega on na zastępowaniu każdej z liter alfabetu inną, zawsze taką samą literą.

Jeżeli na przykład szyfr zdefiniujemy następująco: $\frac{ABCDEFGHIJKLMN\text{OPQRSTUVWXYZ}}{DEFGHIJKLMN\text{OPQRSTUVWXYZABC}}$

to każda litera "A", która pojawi się w tekście jawnym, w szyfrogramie zakodowana zostanie jako D itd. Ogólnie rzecz biorąc, i -ta litera alfabetu kodowana jest jako litera $(i + 3)$ -cia

Napisz program kodujący zadany tekst wg. podanego powyżej sposobu. Następnie zmodyfikuj swój kod tak, aby przesunięcie szyfru było parametrem programu.

2.4 Histogram dla zmiennej losowej z rozkładu normalnego

Napisz funkcję losującą liczby losowe z rozkładu normalnego. Następnie policz kilka tysięcy wartości losowych z tego rozkładu i zrób ich histogram. Przyjmij szerokość klasy histogramu (binu) za równą 0.1

⁸Znany już w czasach starożytnych. Juliusz Cezar szyfrując wiadomości używał przesunięcia = 3, Oktawian August zaś przesunięcia = 1

2.5 Rozkład Maxwella

Rozwiąż zadanie 1.12. Następnie zmodyfikuj program tak, aby wyznaczał histogram prędkości cząsteczek o zadanej masie dla danej temperatury. Znajdź najbardziej prawdopodobną prędkość cząsteczki azotu w temperaturze 300K.

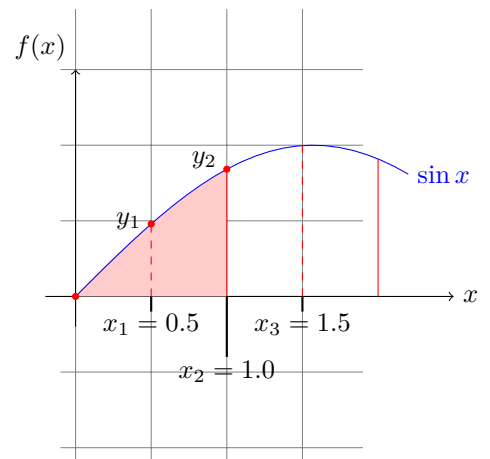
2.6 Całkowanie metodą Simpsona

Zadanie polega na numerycznym całkowaniu dowolnej funkcji (np. $\sin(x)$) wg. wzoru Simpsona:

$$\int_a^b f(x)dx = \frac{h}{3}(y_0 + 4 \cdot y_1 + 2 \cdot y_2 + 4 \cdot y_3 + 2 \cdot y_4 + \dots + 2 \cdot y_{n-2} + 4 \cdot y_{n-1} + y_n) \quad (11)$$

gdzie $y_i = f(x_i)$ a kolejne punkty x generowane są jako $x_i = x_0 + i \cdot h$ a h to odstęp między punktami. Wzór powyższy przybliża całkowaną funkcję odcinkami parabol przechodzącymi przez zadane punkty. Na rysunku obok czerwonym kolorem zaznaczono punkty (x_0, y_0) , (x_1, y_1) , (x_2, y_2) , odcinek paraboli przechodzący przez nie oraz pole pod parabolą. Suma pól wszystkich parabol jest przybliżeniem szukanej całki. Liczba punktów n musi być parzysta.

W tym celu policz wartości funkcji $\sin(x)$ dla $x \in [0.0, \pi]$ zmieniającego się co 0.01.

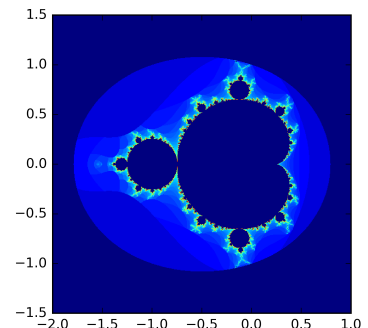


2.7 Zbiór Mandelbrota

Zbiór Mandelbrota tworzą punkty zdefiniowane na płaszczyźnie zespolonej takie, że poniższe równanie rekurencyjne⁹ nie dąży do nieskończoności:

$$\begin{cases} z_0 = 0 \\ z_{n+1} = z_n^2 + c \end{cases} \quad (12)$$

Zadeklaruj w programie dwuwymiarową tablicę T o rozmiarze 600×600 . Każdy element tablicy $T[k][l]$ odpowiadał będzie liczbie zespolonej $c_{kl} = (-2 + k * 0.005, (-1.5 + l * 0.005)i)$. Następnie dla każdego elementu $T[k][l]$ oblicz pierwsze $n_{max} = 100$ wyrazów szeregu i zapisz wartość n , dla którego $|z_n| > 2$.



Przypomnienie: Kwadrat liczby zespolonej $z = a + bi$ obliczamy następująco:

⁹definiujące szereg liczbowy

$$z^2 = (a^2 - b^2) + 2abi$$

2.8 Wyznaczanie kwantyli na podstawie próby losowej

Kwantylem x_p rzędu p dla danego rozkładu zmiennej losowej P_X nazywamy taką liczbę x , dla której wylosowanie liczby nie większej x z tego rozkładu wynosi p . Np mediana, czyli kwantyl $x_{0.5}$ to taka wartość x , że prawdopodobieństwo wylosowania dowolnej liczby z przedziału $(-\infty, x]$ wynosi $1/2$. Kwantyle stopni odpowiadającym dziesiątym częściom prawdopodobieństwa, tzn. $x_{0.1}, x_{0.2}, \dots, x_{0.9}$ nazywamy decylami, kwantyle odpowiadające setnym częściom prawdopodobieństwa - centylami.

Kwantyle bardzo łatwo można wyznaczyć mając (dużą) próbę losową z danego rozkładu. Wystarczy wartości losowe posortować, a następnie odczytać z tablicy odpowiedni element. I tak na przykład, aby znaleźć kwantyl $x_{0.2}$ z próby $N = 1\,000$ wartości losowych, należy tablicę tych wartości posortować a następnie odczytać $N * 0.2 = 200$ element.

Przetestuj generator liczb losowych opisany w zadaniu 1.11. W tym celu wylosuj nim 10 000 wartości i zapisz w tablicy. Wyznacz percentyle metodą opisaną powyżej. Następnie zrób wykres punktów, których odcięte będą "prawdziwymi" percentylami rozkładu normalnego a rzędne - percentylami wyznaczonymi w tym zadaniu. "Prawdziwe" percentyle rozkładu normalnego łatwo znaleźć w Internecie. Można je też policzyć korzystając z dystrybuanty opisanej w zadaniu 1.3.

2.9 Spirala Ulama

Spirala Ulama (zwana również spiralą liczb pierwszych), przedstawiona po raz pierwszy przez Stanisława Ulama w 1963 roku, skonstruowana jest następująco. Na kwadratowej tablicy $N \times N$ wypisywane są liczby naturalne, zaczynając od 1. Następnie wymazuje się te liczby, które nie są pierwszymi. Przykład podano poniżej:

37	36	35	34	33	32	31	37							31	
38	17	16	15	14	13	30	17							13	
39	18	5	4	3	12	29	5						3	29	
40	19	6	1	2	11	28	19						1	2	11
41	20	7	8	9	10	27	41						7		
42	21	22	23	24	25	26	23								
43	44	45	46	47	48	49	43								47

Napisz program, rysujący na ekranie spiralę Ulama dla zadanego N . Zadbaj o odpowiednie formatowanie wydruku tak, aby każda liczba umieszczona była we właściwej sobie kolumnie.

2.10 Sortowanie bąbelkowe

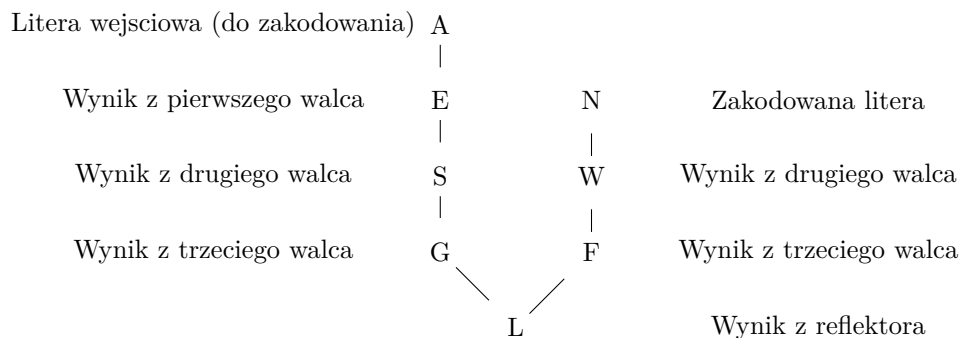
Wylosuj 1000 liczb i zapisz je w tablicy. Następnie posortuj tą tablicę rosnąco wykorzystując algorytm bąbelkowy

3 Zadania trudne

3.1 Enigma

Maszyna szyfrująca Enigma składa się z trzech walców szyfrujących i reflektora (walca odwracającego¹⁰). Działanie każdego z trzech walców szyfrujących jest dokładnie takie jak opisano w zadaniu ?? . Poniżej przedstawiono układ liter na walcach:

Kodowana litera	ABCDEFGHIJKLMNOPQRSTUVWXYZ
walec I	EKMFLGDQVZNTOWYHXUSPAIBRCJ
walec II	AJDKSIRUXBLHWTMCQGZNPYFVOE
walec III	BDFHJLCPRTXVZNYEIWGAKMUSQO
reflektor	YRUHQSLDPXNGOKMIEBFZCWJAT



Kodowana litera przechodzi kolejno przez I, II i III walec. Następnie zamieniana jest przez reflektor i znów przechodzi przez walce III, II i I - w tej właśnie kolejności. Przy kodowaniu litery powracającej z reflektora sygnał wejściowy walca zamieniony jest z wyjściowym. Łatwo sprawdzić, że kodując literę N otrzymuje się w wyniku literę A. Tak więc powtórne szyfrowanie zakodowanej wiadomości prowadzi do odzyskania tekstu jawnego.

Opisany powyżej układ bębnowy tworzy szyfr nieznacznie tylko mocniejszy od szyfru Cezara. Bębny w Enigmie mogą jednakże się obracać, co prowadzi do cyklicznej permutacji klucza szyfrującego. Pierwszy bęben obraca się o jedną pozycję (o jedną literę) przed każdą kodowaną literą. Każdy kolejny bęben obraca się o jedną literę po pełnym obrocie bębna poprzedniego.

3.2 Obliczanie RRSO

Napisz program, który dla zadanej kwoty kredytu i wysokości poszczególnych rat obliczy wartość RRSO (**R**zeczywistej **R**ocznej **S**topy **O**procentowania). Wartość

¹⁰niem. *Umkehrwalze*

ta, oznaczona symbolem r zdefiniowana jest jako:

$$W = \sum_{i=1}^N \frac{A_i}{(1+r)^{t_i}} \quad (13)$$

Poszczególne symbole we wzorze 11 oznaczają:

W - kwota kredytu A_i - kwota spłacona w i -tej racie
 N - liczba rat t_i - ułamek roku na który przypada i -ta rata

Program powinien wczytać liczbę rat N , wysokość kredytu W oraz N kwot poszczególnych rat A_i i wydrukować wartość r na ekran.

Przykład: Bank udzielił pierwszego stycznia kredytu na kwotę 10 000 PLN, spłacanego w 10 comiesięcznych ratach począwszy od marca (na koniec miesiąca). Każda rata jest w wysokości 1 200 PLN. W takim przypadku:

W : 10 000 A_i : 10 wartości równych 1 200
 N : 10 t_i : 3/12, 4/12, 5/12 ... 12/12

RRSO wynosi około 0.3441 czyli 34.41%, bo:

$$\frac{1200}{(1.3441)^{\frac{3}{12}}} + \frac{1200}{(1.3441)^{\frac{4}{12}}} + \frac{1200}{(1.3441)^{\frac{5}{12}}} + \frac{1200}{(1.3441)^{\frac{6}{12}}} + \dots + \frac{1200}{(1.3441)^{\frac{12}{12}}} \approx 10000 \quad (14)$$

Podpowiedź: Wykorzystaj metodę połowienia przedziału, opisaną w zadaniu 1.18.

3.3 Z arabskich na rzymskie

Napisz program, który wczytuje liczbę całkowitą z przedziału $[1, 3000]$ i drukuje ją w zapisie rzymskim. Dla przypomnienia kilka przykładów:

I	=	1	V	=	5	X	=	10	L	=	50
C	=	100	D	=	500	M	=	1000			
IV	=	4	VII	=	7	M	=	1000	XXI	=	21
MCMXCV	=	1995	MXXV	=	1025	CM	=	900	MCMLVI	=	1956

Część II

Projekty obiektowe

4.1 Operacje symetrii

Stwórz bibliotekę klas do pracy z operacjami symetrii i zaimplementuj wszystkie, konieczne do opisu grup przestrzennych. Implementację możesz oprzeć na reprezentacji we *współrzędnych jednorodnych* (jako macierz 4×4).

4.2 Działania na wielomianach

Stwórz klasę `Wielomian` reprezentującą wielomian w przestrzeni liczb rzeczywistych. Zaimplementuj podstawowe operatory: $+$, $*$, $-$. Następnie zdefiniuj wyspecjalizowane klasy pochodne dla kilku często stosowanych wielomianów, np. Czebyszewa.

4.3 Molekuła jako graf

Program stworzony w tym projekcie będzie przechowywał i przetwarzał molekuły jako grafy. Wierzchołkami grafu będą atomy, krawędziami zaś - wiązania. Na początek napisz następujące klasy:

`Atom` - która zawierała będzie co najmniej: indeks atomu, jego położenie (współrzędne x , y oraz z a także typ chemiczny atomu (np. jako liczbę atomową)

`Bond` - zawierającą wskaźniki do dwóch związanych atomów

`MoleculeGraph` - implementującą odpowiedni graf. Możesz np. przechowywać w nim jako prywatne dane:

- listę¹¹ wierzchołków, czyli atomów
- listę krawędzi, czyli wiązań (obiektów klasy `Bond`)
- mapę¹², w której kluczami będą atomy, skojarzone z listą wiązań; w ten sposób dowiesz się, jakie wiązania i jakie inne atomy są bezpośrednio związane z danym atomem

Napisz procedurę, wczytującą dane w formacie PDB i tworzącą odpowiedni graf. Następnie zaimplementuj podstawowe algorytmy związane z przechodzeniem grafu

Teraz możesz wykorzystać program do "chemicznych" zastosowań, jak np:

pierścienie - (np. benzenowy) to po prostu cykl w grafie

¹¹standardowa lista `list` języka Python, `std::vector<Atom>` dla implementacji w C++

¹²standardowy słownik `dict` języka Python, `std::map<Atom, std::vector<Bond>>` dla implementacji w C++

oddzielne molekuly - to grafy rozłączne: dwie grupy atomów A i A' takie, że żaden atom z A nie jest połączony wiązaniem z żadnym atomem z A'

kąty płaskie - wydrukuj wszystkie kąty płaskie w danej molekułe z podaniem wartości kąta i nazw odpowiednich atomów

konformery - napisz procedurę generującą wszystkie konformacje, jakie można stworzyć, obracając wokół wiązań pojedynczych¹³ o 120° i 270°

¹³oczywiście trzeba pominąć wiązania pojedyncze z pierścieni

Część III

Modelowanie komputerowe

5 Metody Monte Carlo - proste losowanie

5.1 Model Isinga

Model Ising to duże uproszczenie ferromagnetyka. Układ składa się z wielu elementów (spinów, magnesów, etc.), z których każdy oddziałuje tylko z najbliższymi sąsiadami. Przykładową realizację modelu Isinga przedstawiono na Rysunku 1. W tym konkretnym przypadku¹⁴ jest to model dwuwymiarowy (wszystkie spiny leżą w jednej płaszczyźnie) oraz dwustanowy, ponieważ każdy spin ma tylko dwie możliwości: "biały" i "szary". Przyjmijmy, że możliwe stany oznaczmy jako -1 i 1. Energię całkowitą układu zdefiniować można następująco¹⁵:

$$E = - \sum_{i,j} \delta(s_i, s_j) \quad (15)$$

Dla przykładu, energia układu z Rysunku 1 wynosi -19 . Do energii wliczają się jedynie oddziaływania z najbliższymi sąsiadami tego samego typu, w pionie lub w poziomie.

Napisz program, losujący konfiguracje układów w dwuwymiarowym dwustanowym modelu Isinga o zadanym rozmiarze (długości boku) N , np. 5, 10 i 15. Wylosuj 10000 takich stanów i dla każdego stanu policz jego energię. Wyniki przedstaw za pomocą histogramu.



Rysunek 1: Przykładowy stan w modelu Isinga o rozmiarze 5×5 . Energia tego stanu wynosi -19 . Każda zielona strzałka oznacza jedno oddziaływanie, wnoszące -1 do energii całkowitej.

¹⁴Można oczywiście badać układy w innej liczbie wymiarów.

¹⁵Jest to energia zdefiniowana przez Potts'a. Możliwe są inne definicje funkcji energii

6 Metody Monte Carlo - schemat Metropolis

Symulacja wg schematu Metropolis polega na wprowadzaniu losowych drobnych zmian układu. Każda zmiana jest akceptowana (bądź odrzucana) wg kryterium opartego o zmianę energii układu. Symulacja prowadzona jest w temperaturze T a jej wynikiem jest rozkład Boltzmana.

6.1 Model Isinga

Symulacja Monte Carlo wg schematu Metropolis przebiega następująco:

- 1) wylosuj fragment badanego układu; tu: wybierz losowo jeden ze spinów
- 2) oblicz energię E_i wylosowanego fragmentu (spinu)
- 3) zmień wylosowany spin na przeciwny
- 4) oblicz energię spinu po zmianie E_{i+1} oraz jej zmianę $\Delta E = E_{i+1} - E_i$
- 5) zaakceptuj zmianę z prawdopodobieństwem $\min(1, \exp(-\Delta E/k_B T))$ gdzie T to temperatura symulacji

Wykonaj symulację wg powyższego algorytmu dla układów o rozmiarze 5×5 , 10×10 i 20×20 .

6.2 Jednoatomowy gaz (np ^{40}Ar)

Wykonaj symulację N atomów gazu w pudle o boku L . Zastosuj periodyczne warunki brzegowe. Energia układu zadana jest wzorem:

$$E = \sum_{i \neq j} 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right]$$

W tym przypadku reguły symulacji Monte Carlo wg schematu Metropolis są następujące:

- w jednym kroku symulacji zmianie podlega położenie \vec{r}_i tylko jednego, i -tego atomu, który jest przesuwany o losowy wektor
- obliczanie zmiany energii polega na policzeniu energii E_{ij} oddziaływania atomu i ze wszystkimi innymi atomami
- atom, który opuści pudło (dowolna jego współrzędna przekroczy L) "wlatuje" ponownie po przeciwnej stronie pudła

7 Dynamika molekularna

7.1 Symulacja gazu jednoatomowego metodą dynamiki dyskretnej

Stosowane obecnie metody dynamiki molekularnej, oparte o numeryczne całkowanie równań ruchu powstały w latach '70 minionego stulecia. Znacznie

wcześniej, bo od lat '50 prowadzono obliczenia dynamiki w ujęciu dyskretnym. Atomy gazu (traktowane jako sztywne kule) poruszały się ruchem jednostajnym, nie działały na nie bowiem żadne siły. Gdy dwa atomy zderzyły się, ich wektory prędkości modyfikowano zgodnie z regułą zderzeń idealnie sprężystych.

Napisz program symulujący dynamikę jednoatomowego gazu (np argonu). Program powinien zawierać następujące elementy:

- 1 Znajdowanie najbliższego w czasie zderzenia kul. Aby dowiedzieć się, kiedy zderzą się dwie kule i oraz j , których prędkości i położenia oznaczone są odpowiednio \vec{v}_i, \vec{v}_j oraz \vec{r}_i, \vec{r}_j , należy rozwiązać równanie kwadratowe:

$$(\vec{v}_i - \vec{v}_j)^2 t^2 + 2(\vec{r}_i - \vec{r}_j)(\vec{v}_i - \vec{v}_j)t + (\vec{r}_i - \vec{r}_j)^2 = \sigma_{ij}^2$$

gdzie σ_{ij} to suma promieni zderzających się kul

- 2 Zmiana prędkości kul - następuje zgodnie z regułą zderzeń idealnie sprężystych:

$$\vec{v}_i' = \vec{v}_i - \frac{(\vec{v}_i - \vec{v}_j)(\vec{r}_i - \vec{r}_j)}{(\vec{r}_i - \vec{r}_j)^2}(\vec{r}_i - \vec{r}_j)$$

$$\vec{v}_j' = \vec{v}_j + \frac{(\vec{v}_i - \vec{v}_j)(\vec{r}_i - \vec{r}_j)}{(\vec{r}_i - \vec{r}_j)^2}(\vec{r}_i - \vec{r}_j)$$

- 3 Aktualizacja położenia wszystkich kulek. Każda kulka i leci zgodnie ze swoim wektorem prędkości \vec{v}_i przez czas t , wyznaczony jak w punkcie [3].

W przygotowaniu...

8 Literatura

Literatura

- [1] Press, William H., Teukolsky, Saul A., Vetterling, William T. and Flannery, Brian P. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007.
- [2] Rapaport, D. C. *The Art of Molecular Dynamics Simulation. Second edition*. Cambridge University Press, 2004.