

Zadania programistyczne z wykorzystaniem **scipy**

October 15, 2024

1 Proste zadania

1. Używając modułu `scipy.constants`, wypisz wartość stałej Plancka oraz prędkości światła w próżni.
2. Skorzystaj z `scipy.constants`, aby obliczyć energię fotonu o długości fali $\lambda = 500 \text{ nm}$.
3. Używając `scipy.linalg`, oblicz wyznacznik macierzy A , gdzie $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$.
4. Wygeneruj wektor 100 liczb losowych z rozkładu normalnego za pomocą `scipy.stats.norm`.
5. Używając `scipy.optimize`, znajdź minimum funkcji kwadratowej $f(x) = (x - 3)^2$.
6. Za pomocą `scipy.integrate.quad`, oblicz całkę funkcji $f(x) = x^2$ w przedziale od 0 do 1.
7. Oblicz iloczyn skalarny dwóch wektorów $[1, 2, 3]$ i $[4, 5, 6]$, używając `scipy.linalg`.
8. Skorzystaj z `scipy.spatial.distance`, aby obliczyć odległość euklidesowa między dwoma punktami $A(1, 2)$ i $B(4, 6)$.
9. Znajdź pierwiastki równania kwadratowego $x^2 - 5x + 6 = 0$ za pomocą `scipy.optimize`.
10. Za pomocą `scipy.special` oblicz funkcję gamma dla liczby 5.

2 Zadania średnio trudne

11. Używając `scipy.optimize.curve_fit`, dopasuj funkcję kwadratową do danych $x = [1, 2, 3, 4, 5]$ i $y = [1, 4, 9, 16, 25]$.

12. Wygeneruj sygnał sinusoidalny o częstotliwości 1 Hz i rozkładzie normalnym, a następnie wygładź go za pomocą filtru Gaussa z `scipy.ndimage`.
13. Używając `scipy.interpolate`, zaimplementuj interpolację liniową dla danych $x = [0, 1, 2, 3, 4]$ oraz $y = [1, 3, 7, 13, 21]$.
14. Wykorzystaj `scipy.fftpack`, aby znaleźć dyskretną transformację Fouriera sygnału $[1, 2, 3, 4]$.
15. Używając `scipy.stats.ttest_ind`, porównaj dwie próbki danych $[2.5, 3.1, 2.9, 4.2]$ oraz $[3.0, 2.9, 3.2, 4.0]$ pod kątem istotnych różnic między nimi.
16. Skorzystaj z `scipy.linalg`, aby rozwiązać układ równań liniowych $2x + 3y = 5$ oraz $x - y = 1$.
17. Za pomocą `scipy.integrate.odeint`, rozwiąż równanie różniczkowe $\frac{dy}{dx} = -2y$ z warunkiem początkowym $y(0) = 1$.
18. Użyj `scipy.stats.linregress`, aby przeprowadzić analizę regresji liniowej dla danych $x = [1, 2, 3, 4, 5]$ oraz $y = [2.1, 4.2, 6.0, 8.1, 10.1]$.
19. Znajdź minimum funkcji $f(x, y) = (x-2)^2 + (y-3)^2$ przy użyciu `scipy.optimize.minimize`.
20. Używając `scipy.spatial`, zaimplementuj algorytm znajdowania najbliższych sąsiadów (ang. nearest neighbors) dla zbioru punktów na płaszczyźnie.

3 Zadania zaawansowane

21. Za pomocą `scipy.signal`, zaprojektuj filtr dolnoprzepustowy Butterwortha o częstotliwości odciecia 0.4 Hz i zbadaj jego charakterystykę.
22. Użyj `scipy.integrate.solve_ivp`, aby zasymulować ruch wahadła z równaniem $\frac{d^2\theta}{dt^2} + \frac{g}{L} \sin \theta = 0$ z warunkiem początkowym $\theta(0) = \frac{\pi}{4}$.
23. Używając `scipy.optimize.linear_sum_assignment`, rozwiąż problem przypisania (assignment problem) dla macierzy kosztów $C = \begin{bmatrix} 4 & 1 & 3 \\ 2 & 0 & 5 \\ 3 & 2 & 2 \end{bmatrix}$.
24. Wykorzystaj `scipy.ndimage` do detekcji krawędzi w obrazie za pomocą filtru Sobela.
25. Używając `scipy.sparse`, zaimplementuj operacje na macierzach rzadkich i wyznacz iloczyn dwóch dużych macierzy rzadkich.
26. Używając `scipy.spatial.Delaunay`, znajdź triangulację Delaunaya dla zestawu punktów w przestrzeni 2D.
27. Użyj `scipy.optimize.differential_evolution`, aby znaleźć globalne minimum funkcji Rastrigina w przestrzeni 2D.

28. Skorzystaj z `scipy.cluster.hierarchy`, aby przeprowadzić hierarchiczne grupowanie danych z próby [1.2, 2.3, 3.1, 4.0, 5.9, 7.8].
29. Użyj `scipy.linalg.svd`, aby znaleźć wartości osobliwe dla macierzy $A = \begin{bmatrix} 3 & 1 & 1 \\ -1 & 3 & 1 \end{bmatrix}$.
30. Za pomocą `scipy.signal.find_peaks`, znajdź lokalne maksima w sygnale [1, 3, 7, 1, 2, 8, 3, 2, 6, 4].