Programowanie I R

Zadania – seria 11.

Trochę filtrowania, całkowanie równań różniczkowych cząstkowych

Zadanie 1. `fourier_filter` – **Filtr Fourierowski.**

W analizie danych eksperymentalnych, które mają cechy periodyczne, jedną z metod pozbywania się szumów jest filtr Fourierowski, który w najprostszej wersji polega na obliczeniu transformaty Fouriera a następnie selekcji częstotliwości poniżej częstotliwości odcięcia, a powyżej wyzerowania. Z wybranych częstotliwości sygnał jest składany na nowo i zwracany.

Napisz program `fourier_filter`, który w argumentach wywołania przyjmuje funkcję sygnału w postaci stringa kodu z wykorzystaniem biblioteki `Numpy` (na przykład `"np.sin(2*t) + np.cos(3*t)"`), częstotliwość próbkowania, czas trwania sygnału, częstotliwość odcięcia filtru dolnoprzepustowego, oraz szerokość rozkładu normalnego z którego losowany jest szum. Program powinien wygenerować sygnał w funkcji `signal_generator`, dodać szum gaussowski w funkcji `noise_generator`, a następnie zastosować filtr Fourierowski w funkcji `fourier_filter`. Wygeneruj wykres, na którym porównasz sygnał oryginalny, sygnał z szumem oraz sygnał po filtracji.

Przetestuj działanie programu dla różnych funkcji sygnału, zmieniaj częstość odcięcia i szum.

- $f(t) = \sin(t) + 0.5\cos(2t) + 0.25\sin(3t)$
- $f(t) = \sin(2t) + 0.5\cos(4t) + 0.25\sin(6t)$
- $f(t) = \sin(4t)^4 + \cos(3t)^3 + \sin(2t)^2$
- $f(t) = e^{-(t-5)^2/0.5}$

By ulepszyć działanie filtra (w szczególności dla impulsu gaussowskiego), zaimplementuj odcinanie przez iloczyn z funkcją supergaussowską o szerokości $2\omega_{\text{cutoff}}$.

Wskazówka – posłuż się metodą `eval` by obsłużyć string z kodem.

Dla chętnych – posłuż się biblioteką `Sympy`, by z wyrażenia tekstowego utworzyć funkcję z pomocą `sp.lambdify`, o którym możesz poczytać tutaj.

Opracowanie: Bartosz Kasza.

Zadanie 2. `heat` – **Model przewodnictwa cieplnego.**

Równanie przewodnictwa cieplnego opisuje przepływ ciepła w ośrodku materialnym. Przyjmuje ono postać

$$\frac{\partial T}{\partial t} - \alpha \Delta T = 0, \quad \Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2},$$

gdzie $T = T(t, \mathbf{r})$ jest temperaturą w punkcie o wektorze wodzącym $\mathbf{r}$ w chwili t, zaś α – charakteryzującą materiał stałą nazywaną *współczynnikiem wyrównania temperatury* lub *dyfuzyjnością cieplną*, określającą tempo, z jakim przebiegają zmiany temperatury w danym materiale. Po uzupełnieniu tego równania o warunki początkowe, czyli zadaniu rozkładu temperatury w

ośrodku w chwili $t = 0$, oraz warunki brzegowe, czyli zadaniu ustalonego, niezmiennego rozkładu temperatury na brzegu obszaru, jesteśmy w stanie wyznaczyć rozkład temperatury w ośrodku w dowolnej chwili $t > 0$.

Rozważmy cienką metalową płytkę w kształcie kwadratu. Przyjmujemy, że w chwili początkowej $t = 0$ płytka miała temperaturę T_0. Do górnej krawędzi płytki przyłożono w chwili początkowej ciało o temperaturze T_1, do krawędzi prawej – ciało o temperaturze T_2, do krawędzi dolnej – ciało o temperaturze T_3, zaś do krawędzi lewej – ciało o temperaturze T_4; zakładamy przy tym, że ciała te utrzymują stałą temperaturę oraz że temperatury T_k, $k = 0, 1, 2, 3, 4$, znajdują się w przedziale od 0°C do 100°C.

Napisz program `heat`, który posługując się metodą różnic skończonych rozwiąże równanie przewodnictwa cieplnego dla omówionej płytki i wyznaczy rozkład temperatury wewnątrz płytki w dowolnej chwili $t > 0$. Program powinien przyjmować jako argumenty wywołania pięć liczb zmiennoprzecinkowych odpowiadających wartościom, odpowiednio, T_0, T_1, T_2, T_3 i T_4. Wynikiem działania programu powinna być animacja przedstawiająca zmiany temperatury w płytce. **Przyjmij** $\alpha = 2$, krok przestrzenny (na obu osiach przestrzennych) $\delta x = 1$, krok czasowy $\delta t = \delta x^2/4\alpha$ oraz rozmiar płytki $l = 50\delta x$.

Metodę różnic skończonych 2D Laplasjanu można wyprowadzić z następujących przybliżeń na siatce $x_i = i\delta x$, $y_j = j\delta y$ w czasie $t_n = n\delta t$:

$$\frac{\partial T}{\partial t} \approx \frac{T_{ij}^{n+1} - T_{ij}^{n}}{\delta t}, \quad \frac{\partial T}{\partial x} \approx \frac{T_{i+1j}^{n} - T_{ij}^{n}}{\delta x}, \quad \frac{\partial T}{\partial y} \approx \frac{T_{ij+1}^{n} - T_{ij}^{n}}{\delta y}, \quad \delta x = \delta y$$
$$\frac{\partial^2 T}{\partial x^2} \approx \frac{T_{i+1j}^{n} - 2T_{ij}^{n} + T_{i-1j}^{n}}{\delta x^2}$$

i analogicznie dla części y Laplasjanu. Finalnie otrzymujemy pełne wyrażenie postaci:

$$T(x_i, y_j, t_n) \approx T_{ij}^{n+1} = \alpha \frac{\delta t}{\delta x^2}(T_{i+1j}^{n} + T_{ij+1}^{n} + T_{i-1j}^{n} + T_{ij-1}^{n} - 4T_{ij}^{n}) + T_{ij}^{n}.$$

Opracowanie: Bartłomiej Zglinicki. Edycja: Bartosz Kasza.