Pakiety matematyczne MAT1349 - 2021

Lista 3: Modelowanie układów 1D i 2D

Przykładową animację zmian macierzy można otrzymać następująco:

```
using Plots
an = @animate for k in 1:10
 heatmap(rand((0, 1),10,10),
   aspect_ratio = 1,
   seriescolor = palette([:blue,:yellow]),
   colorbar = :none,
   framestyle = :none
 )
 end
gif(an,"mojaAnimacja.gif", fps = 2)
```

Tu pokazujemy 10 macierzy losowych co 0.5 s.

1. **Łańcuchy Markowa.** Wiele prostych układów losowych można modelować za pomocą wektora długości n, w którym π_j to prawdopodobieństwo bycia w stanie j, $\sum_j \pi_j = 1$. Jego zmiany opisujemy tzw. *macierzą przejścia*, w której P_{ij} oznacza prawdopodobieństwo przejścia ze stanu i do stanu j w 1 jednostce czasu, $\sum_j P_{ij} = 1$. Tak więc po 1 jednostce czasu otrzymujemy $\pi'_i = \sum_j \pi_j P_{ji}$, równoważnie $\pi' = P\pi$. Wynika z tego, że macierz przejścia w 2 jednostkach czasu to P^2, podobnie macierz przejścia w k krokach to P^k. Rozważmy:

   (a) Prosty model pogody. Stany to słonecznie/deszczowo, macierz przejścia zależy od klimatu oraz pory roku i ma ogólną postać

   $$P = \begin{bmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{bmatrix}.$$

   (b) Model JC69 mutacji DNA. Stany to aminokwasy A, C, G, T. Macierz przejścia to

   $$P_{ij} = \frac{1}{4} + \frac{3}{4}\mathrm{e}^{-4\nu/3}, \quad i = j;$$
   $$P_{ij} = \frac{1}{4} - \frac{1}{4}\mathrm{e}^{-4\nu/3}, \quad i \neq j,$$

   gdzie stała ν określa tempo mutacji.

   Zbadaj, jak będzie zmieniać się macierz przejścia dla krótkich oraz długich czasów. Czy do czegoś zbiega? Jak szybko? Co będzie działo się z różnymi stanami początkowymi π?

   Opcjonalnie 1: Sprawdź, czy symulacja będzie szybsza z użyciem macierzy statycznych z pakietu `StaticArrays`.

Opcjonalnie 2: Sprawdź, wynik pokrywa się z symulacją prawdziwie losową, tzn. taką, w której zaczynamy od danego stanu i faktycznie w jednostce czasu losujemy z prawdopodobieństwem P_{ij} w który inny stan się zmieni.

2. **Gra w życie Conwaya.** Stwórz dynamiczną symulację *gry w życie Conwaya* (*Conway's game of life*). Jej reguły są następujące:

   (a) Gra toczy się na **torusie** $n \times m$, tzn. macierzy, której każda wewnętrzna komórka sąsiaduje z 8 komórkami o współrzędnych ± 1 w pionie i poziomie, a oprócz tego komórki pierwszej kolumny sąsiadują z komórkami ostatniej kolumny, podobnie komórki pierwszego wiersza sąsiadują z komórkami ostatniego wiersza. *Sugestia:* zamiast rozpatrywać wszystkie przypadki ręcznie można skorzystać z wbudowanej funkcji `mod1`.

   (b) Komórki mają 2 stany: „żywy" i „martwy".

   (c) Mając aktualny stan gry A następną iterację A' obliczamy według następujących reguł:

      i. Jeżeli A_{ij} jest żywa i ma ilość żywych sąsiadów różną od 2 lub 3, to umiera, tzn. A'_{ij} jest martwa.

      ii. Jeżeli A_{ij} jest martwa i ma 3 żywych sąsiadów, to A'_{ij} jest żywa.

      iii. W pozostałych przypadkach nie zmienia się nic, $A'_{ij} = A_{ij}$.

   Stwórz animację przykładowych wzorów stałych, okresowych oraz podróżujących (są pokazane m.in. na Wikipedii).

3. **Dywan Sierpińskiego.** Napisz algorytm generujący macierz logiczną $3^n \times 3^n$ będącą przybliżeniem *dywanu Sierpińskiego* (*Sierpinski carpet*). Wartości `true` niech oznaczają białe pola, a `false` czarne pola. Możesz użyć rekurencji lub napisać algorytm bezpośredni.
   Opcjonalne: Napisz funkcję szacującą pole tej figury jako funkcję n (części czarnej lub białej, obojętnie) **bez** alokowania macierzy.

4. **Rozchodzenie się ciepła.** Opiszmy temperaturę pewnego 1D obiektu (możemy myśleć o pręcie lub przekroju przez ścianę) jako n punktów rozmieszczonych co Δx, u_k^t to temperatura punktu k w chwili t przy jednostce czasu Δt. Rozchodzenie się ciepła opisane jest tzw. *dyskretnym równaniem ciepła*

$$D_t u_k^t = c \nabla_k^2 u_k^t,$$

   gdzie D_t to dyskretna pochodna, ∇_k^2 to dyskretny laplasjan, a c to stała przewodnictwa ciepła,

$$D_t u_k^t = \frac{\Delta_t u_k^t}{\Delta t} = \frac{u_k^{t+1} - u_k^t}{\Delta t}, \quad \nabla_k^2 u_k^t = \frac{u_{k+1}^t - 2u_k^t + u_{k-1}^t}{\Delta x^2}.$$

   Wzoru tego należy użyć, aby obliczyć u_k^{t+1} mając dane u_k^t. Wartości brzegowe u_1^t oraz u_n^t traktujemy specjalnie:

   (a) $u_1^t = u_1^0, u_n^t = u_n^0$ opisuje sytuację, gdy z prawej i lewej strony utrzymujemy temperatury odpowiednio u_1^0 oraz u_n^0

(b) $u_1^t = u_2^{t-1}, u_n^t = u_{n-1}^{t-1}$ opisuje sytuację, gdy brzegi są izolowane termicznie.

Przeanalizuj rozchodzenie się ciepła dla różnych warunków początkowych u_k^0 (np. pojedyncze punkty ciepła, sinus, skok temperatury) oraz różnych warunków brzegowych.

Uwaga: Aby symulacja była stabilna musi zachodzić $c\Delta t/\Delta x^2 \leq 1/2$.