

# Dyfuzja ciepła i *domain decomposition*

Programowanie II R - zadanie semestralne

## 1 Wstęp

Przy modelowaniu wielu procesów fizycznych użyteczna jest dyskretyzacja przestrzeni za pomocą regularnej siatki. Czas wykonania tego rodzaju symulacji można znacząco skrócić, wykonując obliczenia jednocześnie na wielu wątkach. Często wykorzystywanym podejściem jest *domain decomposition*, tj. podział domeny na subdomeny, wewnątrz których obliczenia wykonywane są przez jeden wątek. Celem tego zadania jest modelowanie dyfuzji ciepła w dwóch wymiarach i zrównoleżenie obliczeń, używając *domain decomposition*.

## 2 Dyfuzja ciepła

Zmiana temperatury  $T$  na skutek przepływu ciepła opisana jest równaniem dyfuzji:

$$\frac{\partial T}{\partial t} = \alpha \Delta T, \quad (1)$$

gdzie  $\alpha$  to dyfuzyjność termiczna materiału, a  $\Delta$  oznacza laplasjan.

W celu rozwiązania tego równania w dwóch wymiarach podzielimy przestrzeń na siatkę  $L \times L$  równoodległych punktów i użyjemy metody różnic skończonych. Drugą pochodną po przestrzeni można aproksymować za pomocą centralnego ilorazu różnicowego:

$$\frac{\partial^2 T_{i,j}}{\partial x^2} \approx \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2} \quad (2)$$

$$\frac{\partial^2 T_{i,j}}{\partial y^2} \approx \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{\Delta y^2} \quad (3)$$

gdzie indeksy  $i, j$  numerują punkty w przestrzeni, a  $\Delta x$  i  $\Delta y$  to odległości między punktami. Pochodną po czasie można aproksymować za pomocą progresywnego ilorazu różnicowego:

$$\frac{\partial T_{i,j}}{\partial t} \approx \frac{T_{i,j}^{n+1} - T_{i,j}^n}{\Delta t}, \quad (4)$$

gdzie  $\Delta t$  to długość kroku czasowego symulacji. Kryterium stabilności ma postać:

$$\Delta t \leq \frac{1}{2\alpha \left( \frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)} \quad (5)$$

## 3 Zadania

### 3.1 Program do modelowania dyfuzji ciepła

Napisz program do modelowania dyfuzji ciepła na wielu wątkach przy użyciu podejścia *domain decomposition*. W programie powinna być oddzielona logika związana z siatką obliczeniową od szczegółów modelowanego

przypadku. Można to uzyskać tworząc klasę *Grid2D*, która dzieli dwuwymiarową regularną siatkę na subdomeny. Do każdej subdomeny przypisany powinien być jeden wątek, który wykonuje w niej obliczenia. W założeniu klasa *Grid2D* powinna być dostosowana do modelowania różnych procesów fizycznych.

Szczegóły układu modelowanego w tym zadaniu (tj. dyfuzji ciepła) powinny znajdować się w klasie *HeatSolver*, która wykorzystuje klasę *Grid2D*. Idealnie by było, gdyby klasa *HeatSolver* "nie wiedziała" o wielowątkowości. Cała logika związana z wielowątkowością powinna być w *Grid2D* (lub jeszcze innej klasie pomocniczej).

### 3.1.1 Parametry symulacji

- $\alpha = 10^{-3} \text{m}^2/\text{s}$  - zbliżone do przewodnictwa płytek węglowych wykorzystywanych jako tarcze cieplne w statkach kosmicznych
- $X = Y = 50\text{cm}$  - rozmiar modelowanej domeny
- $\Delta x = \Delta y = X/L$
- czas symulacji to minimum 10 sekund

Warunki początkowe – dwa punktowe źródła ciepła:

$$T_0(x, y) = 0 + 100 * \exp\left(-\frac{\sqrt{(x - X/4)^2 + (y - Y/4)^2}}{1}\right) + 1 * \exp\left(-\frac{\sqrt{(x - 3X/4)^2 + (y - 3Y/4)^2}}{10}\right) [C]$$

Warunki brzegowe Dirichleta (stała temperatura 0C na ściankach).

## 3.2 Sprawozdanie

Przygotuj krótkie (2-4 strony) sprawozdanie opisujące:

- Strukturę kodu
- Uzyskane wyniki i ich zależność od parametru  $L$ . Sprawdź zachowanie energii (uwzględnij ciepło uciekające przez ścianki).
- Skalowalność czasu obliczeń z liczbą wątków, jak zależy ona od  $L$ ?

## 3.3 Wskazówki

- Do obsługi wątków można wykorzystać `std::thread`.
- Kod powinien być czytelny (może być podzielony na kilka plików) i powinien zawierać komentarze.
- Przygotuj krótką dokumentację dot. sposobu uruchomienia symulacji i formatu danych wyjściowych.
- Wizualizacje mogą być wykonane innym narzędziem niż C++. Załącz skrypty wykorzystane do wizualizacji wyników.

## 3.4 Dodatkowo punktowane

- Zaimplementuj dwuwymiarową tablicę używając `std::mdspan`.
- Przetestuj, czy czas obliczeń różni się, gdy podział na subdomeny robiony jest wzdłuż osi  $x$  i wzdłuż osi  $y$ .
- Użyj 2D *domain decomposition*, czyli podziel domenę na subdomeny zarówno wzdłuż kierunku  $x$ , jak i  $y$ .

Opracowanie: Piotr Dziekan.