Zadanie 1 – dekompozycja domenowa

Etap 1: implementacja sekwencyjna i projekt algorytmu równoległego

WYKONAŁ:

ARKADIUSZ BEER

Spis treści

Wstęp

Tematem pracy jest rozwiązanie równania dyfuzji cieplnej w dwóch wymiarach (rozpływ ciepła na kwadratowej płytce). Równanie dyfuzji jest przedstawione wzorem:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + f(x,y,t) - \frac{dT}{dt} = 0$$

gdzie:

T(x,y,t) – funkcja koncentracji ciepła

f(x,y,t) –funkcja źródła ciepła

Model numeryczny

Aby uzyskać informacje o zmianie temperatury na płytce to potrzebne nam będą 2 dodatkowe równania na 1 i 2 pochodną:

$$f^1 = \frac{f(x + \Delta x) - f(x - \Delta x)}{2 * \Delta x}$$

$$f^2 = \frac{f(x + \Delta x) - 2 * f(x) + f(x - \Delta x)}{\Delta x^2}$$

Są to wzory na pochodne w jednym wymierzę.

Wprowadźmy oznaczenie:

$$T_{i,j} = f(x,y)$$

Wtedy:

$$T_{i+1,j} = f(x + \Delta x, y)$$

$$T_{i,j+1} = f(x, y + \Delta y)$$

$$T_{i-1,j} = f(x - \Delta x, y)$$

Itp.

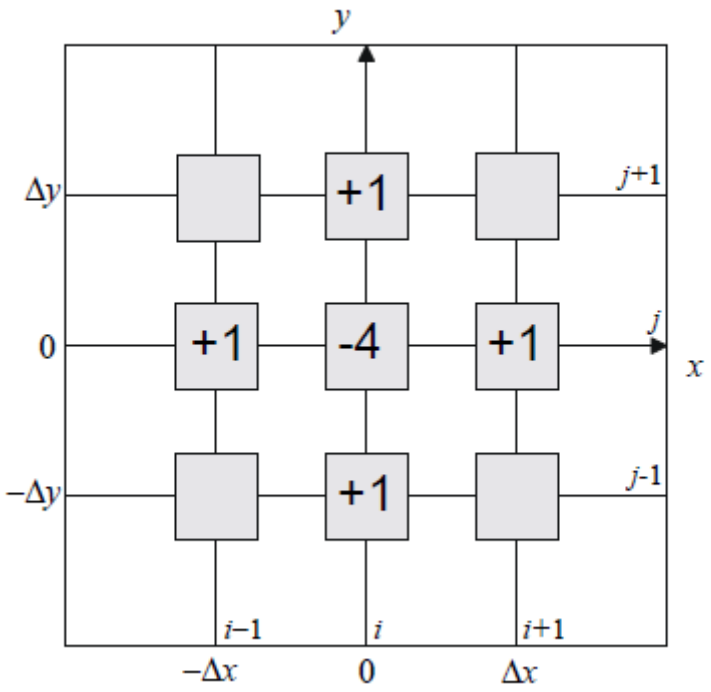
Dla pochodnych cząstkowych drugiego rzędu mamy:

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{i+1,j} - 2 * T_{i,j} + T_{i-1,j}}{\Delta x^2}$$

$$\frac{\partial^2 T}{\partial y^2} = \frac{T_{i,j+1} - 2 * T_{i,j} + T_{i,j-1}}{\Delta y^2}$$

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = \frac{T_{i+1,j} - 2 * T_{i,j} + T_{i-1,j}}{\Delta x^2} + \frac{T_{i,j+1} - 2 * T_{i,j} + T_{i,j-1}}{\Delta y^2}$$

Udział punktów jakie są brane pod uwagę we wzorze są przedstawione w rys. nr: 1



Rys. nr: 1 Wzorzec obliczeń

Dla równych odstępów między punktami badanymi dla współrzędnych X i Y ($\Delta x = \Delta y$) mamy:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = \frac{T_{i+1,j}\, T_{i,j+1} + T_{i,j-1} + T_{i-1,j} - 4 * T_{i,j}}{\Delta x^2}$$

Sumę dwóch pochodnym cząstkowych nazwijmy Laplasjanem z T.

$$\nabla^2 T = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}$$

Do obliczenia kolejnych w czasie wartości temperatur na płytce musimy przedstawić pochodną funkcji koncentracji ciepła z wzoru:

$$\frac{dT}{dt} = \frac{T^{t+1} - T^t}{\Delta t}$$

Podstawiając pod wzór równania dyfuzji ciepła mamy:

$$T^{t+1} = \Delta t * \left(\nabla^2 T^t + f(x, y, t)\right) + T^t$$

Stabilność algorytmu

Dla braku źródła ciepła:

$$f(x, y, t) = 0$$

$$T^{t+1} = \Delta t * \nabla^2 T^t + T^t$$

$$T^{t+1} = \Delta t * \left(\frac{T^{\boldsymbol{t}}_{i+1,j}\, T^{\boldsymbol{t}}_{i,j+1} + T^{\boldsymbol{t}}_{i,j-1} + T^{\boldsymbol{t}}_{i-1,j} - 4 * T^{\boldsymbol{t}}_{i,j}}{\Delta x^2}\right) + T^t$$

Z tego wzoru można wyodrębnić parametr określający stabilność obliczeń:

$$0 < \frac{\Delta t}{\Delta x^2} < 0.5$$

Dla $\Delta t = 1$ powinniśmy dobierać parametr $\Delta x > \sqrt{2}$

Algorytm sekwencyjny

Algorytm sekwencyjny przy znanym już modelu numerycznym jest bardzo prosty. Tworzymy siatkę o wymiarach N na M. Mamy N*M badanych punktów. Dla danych brzegowych tworzymy dodatkowe punkty który spełniałyby warunki brzegowe. Po wszystkich punktach ‘przykładamy’ wzorzec obliczający koncentrację ciepła.

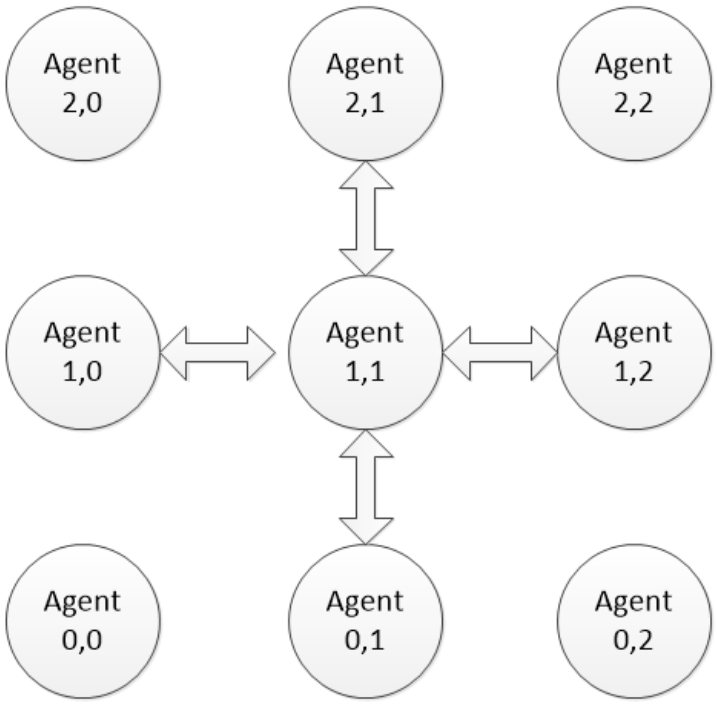
Dla punktów poza siatką przyjmujemy wartość 0 jako wartość zadaną z warunków brzegowych (warunek Dirichleta). Przyjmując źródło ciepła jako wartość 0 to suma każdego badanego punktu będzie dążyć do 0.

Implementacja algorytmu sekwencyjnego jest udostępniona pod adresem:

https://github.com/arekbee/FSharpAgents/tree/master/src/Sequential/FSharpAgents.Sequential

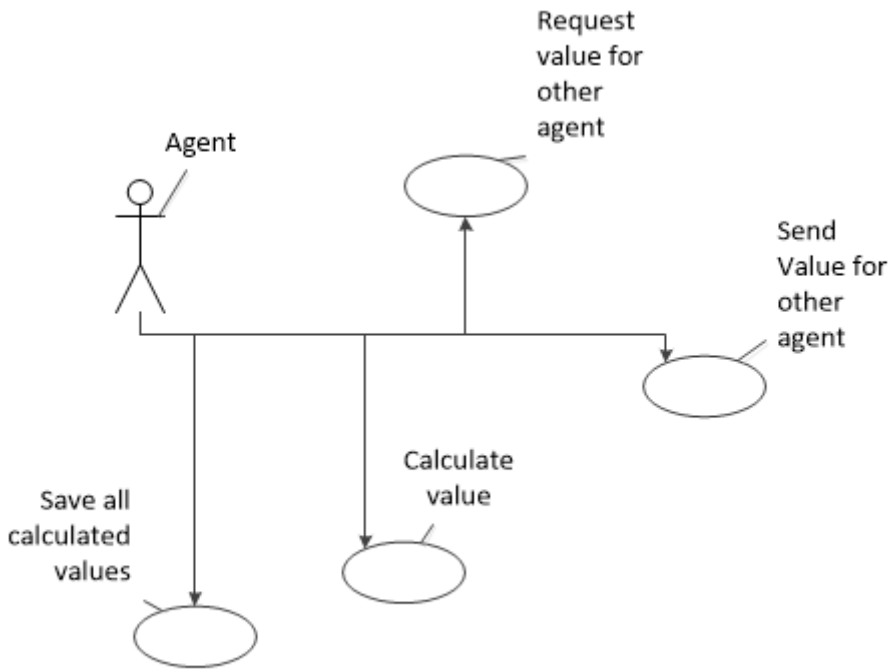
Algorytm równoległy

Do zaprojektowania algorytm równoległego dla siatki z badanymi punktami został zaproponowany model agentowy. Załóżmy, że dla każdego badanego punktu jest mały program, który potrafi komunikować się ze innymi programami. Cała koncepcja jest przedstawiona w rys. nr: 2.



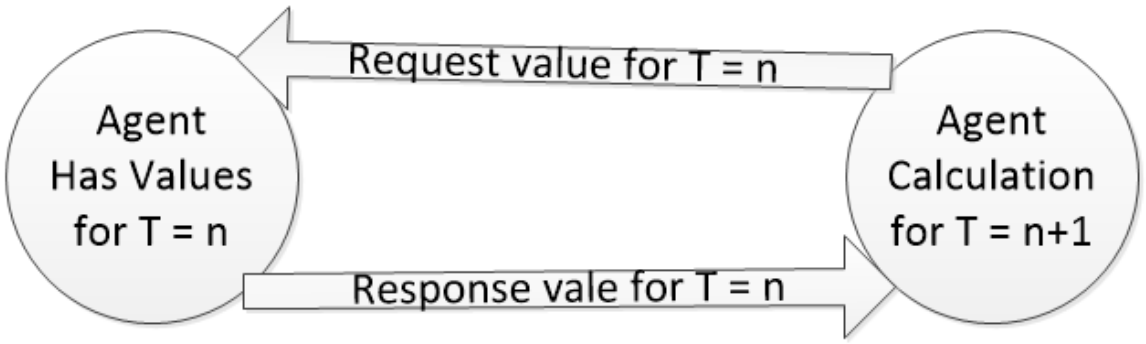
Rys. nr: 2 Koncepcja agentów w siatce

Wśród najważniejszych umiejętności agenta jest możliwość wysyłania prośby o wartość do swoich sąsiadów. Sąsiad agenta może wysłać taką wartość do naszego agenta, a nasz agent dzięki tej informacji może obliczyć kolejną wartość koncentracji ciepła i zapisać ją w pamięci agenta. Przypadki użycia zostały przedstawione na rys. nr: 3.



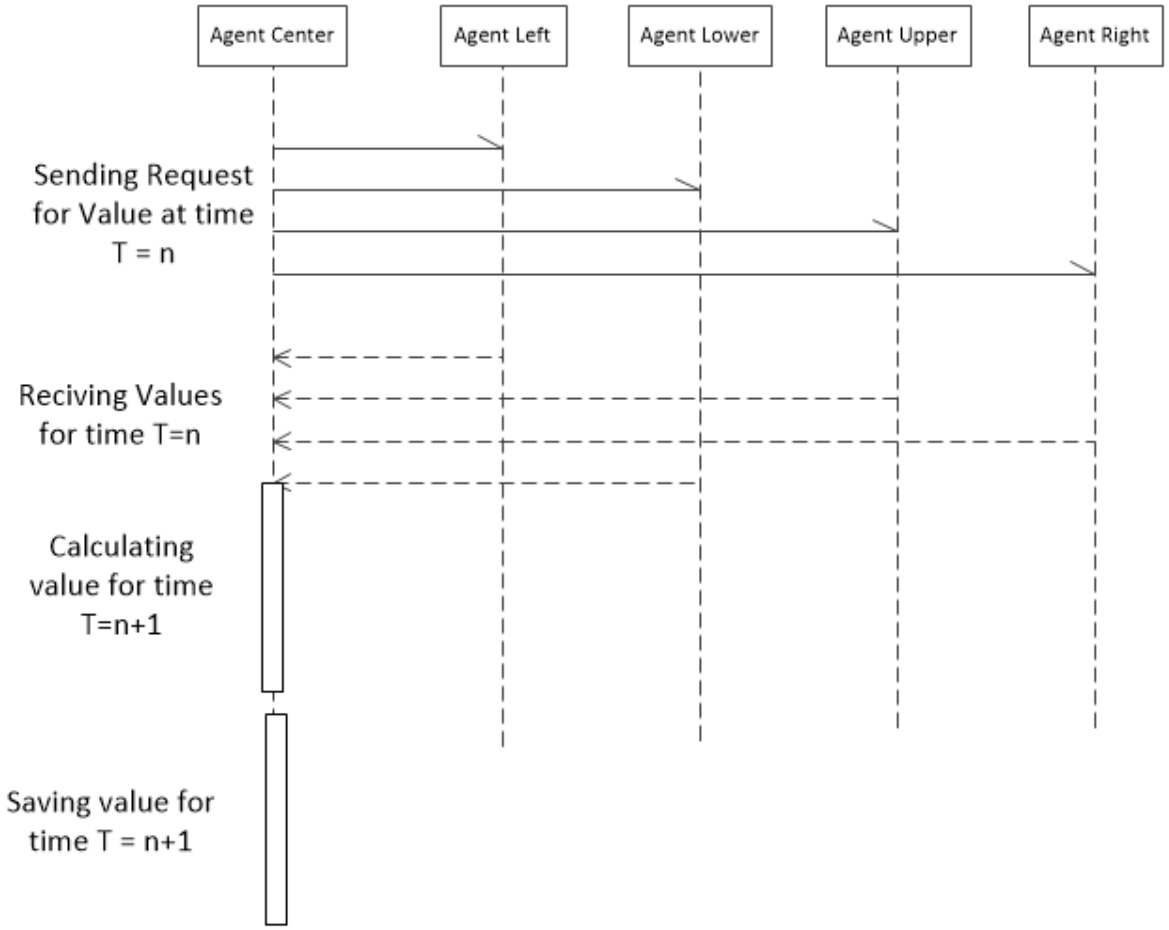
Rys. nr: 3 Przypadki użycia pojedynczego agenta

Kiedy agent chce obliczyć wartość koncentracji ciepła w chwili T = n+1 to wysyła prośbę o wartość w chwili T = n do swojego sąsiada, a on odpowiada mu przesyłając wartość. Komunikacja została przedstawiona na rys. nr: 4.



Rys. nr: 4 Komunikacja między 2 agentami

Kiedy mamy do czynienia z czteroma agentami to wysyłamy asynchronicznie prośbę o wartość i kiedy uzyskami odpowiedz od wszystkich wymaganych agentów to przystępujemy do obliczenia nowej wartości oraz zapisania tej wartość w pamięci. Rys. nr 5 przedstawia sekwencje komunikacji agenta centralnego z agentami sąsiadującymi z nim.



Rys. nr:5 Diagram sekwencji komunikacji agentów dla wzorca

Implementacja algorytmu równoległego za pomocą agentów jest udostępniona w poniższym kodzie źródłowym:

https://github.com/arekbee/FSharpAgents/tree/master/src/MailboxProcessor

Checklista algorytmu równoległego

Wszystkie taski (agenci) wykonują mniej więcej taką samą liczbę komunikacji.

Każdy agent komunikuje się maksymalnie do 4 agentów (jest to bardzo mała liczba komunikacji).

Wszystkie rodzaje komunikacji jakie mogę agent dokonać są wykonywane prawie równolegle.

Każdy agent działa niezależnie od działania innych agentów (blokada agenta lub deaktywacja).

Model alg. równoległego jest modelem obliczeń lokalnych (zredukowano liczbę komunikacji z innymi agentami).

Liczba agentów nie jest zależna od siły ani od ilości procesorów. Liczba agentów może być większa od liczby procesorów jak i od maksymalnej liczy wątków.

Wszystkie agenci są porównywalnych rozmiarów.

Wielkość potrzebnej pamięci starano się zredukować do minimum (przetrzymuje się informacje o kolejnych wartościach koncentracji ciepła), aby koszt jednego agenta był najmniejszy.

Przy zwiększeniu liczby badanych punktów proporcjonalnie zwiększa się liczba agentów.

Liczba agentów jest mocno związana z liczba badanych punktów. Nie ma możliwości zredukowania liczby agentów przy stałej liczbie badanych punktów.

Przedstawiony alg. zrównoleglania problemu dyfuzji ciepła za pomocą modelu agentowego nie może być bardzo prosto zmieniony na algorytm sekwencyjny, gdyż są to różne modele programowania.