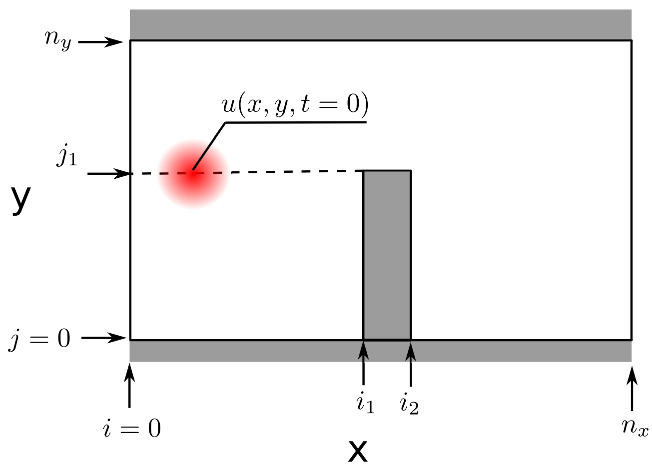
Projekt 8: Równanie adwekcji-dyfuzji – symulacja transportu masy metodą Cranka-Nicolson.

Tomasz Chwiej,Alina Mreńca-Kolasińska, Elżbieta Strzałka

11 grudnia 2018

1 Wstęp



Rysunek 1: Geometria układu, w którym rozkład gęstości $u(x, y, t)$ zmienia się ze względu na adwekcję w polu prędkości i pod wpływem dyfuzji. Na lewym i prawym brzegu nałożone są periodyczne warunki brzegowe – płyn przepływający przez prawy brzeg pojawia się na lewym brzegu.

Na zajęciach wykonamy symulację transportu masy rozwiązując równanie adwekcji-dyfuzji

$$\frac{\partial u}{\partial t} = -\vec{v} \cdot \nabla u + D\nabla^2 u \tag{1}$$

wykorzystując niejawny schemat Cranka-Nicolson.

2 Dyskretyzacja, metoda CN, periodyczne warunki brzegowe, warunki początkowe

2.1 Dyskretyzacja

Wprowadzamy siatkę równoodległych węzłów w przestrzeni (x,y) oraz na osi czasu (t)

$$t = t_n = \Delta t \cdot n, \quad n = 0, 1, \ldots \tag{2}$$

$$\Delta x = \Delta y = \Delta \tag{3}$$

$$x = x_i = \Delta \cdot i, \quad i = 0, 1, \ldots, n_x \tag{4}$$

$$y = y_i = \Delta \cdot j, \quad j = 0, 1, \ldots, n_y \tag{5}$$

$$u(x, y, t) = u(x_i, y_j, t_n) = u^n_{i,j} \tag{6}$$

$$\vec{v} = (v^x(x_i, y_j), v^y(x_i, y_j)) = (v^x_{i,j}, v^y_{i,j}) \tag{7}$$

2.2 Metoda Cranka-Nicolson

Równanie (1) dyskretyzujemy stosując dwupunktowy iloraz różnicowy "do przodu" dla pochodnej czasowej oraz symetryczne ilorazy różnicowe dla pochodnych przestrzennych

$$\begin{aligned}
\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} &= -\frac{1}{2} v_{i,j}^{x} \left[\left(\frac{u_{i+1,j}^{n} - u_{i-1,j}^{n}}{2\Delta} \right) + \left(\frac{u_{i+1,j}^{n+1} - u_{i-1,j}^{n+1}}{2\Delta} \right) \right] \\
&- \frac{1}{2} v_{i,j}^{y} \left[\left(\frac{u_{i,j+1}^{n} - u_{i,j-1}^{n}}{2\Delta} \right) + \left(\frac{u_{i,j+1}^{n+1} - u_{i,j-1}^{n+1}}{2\Delta} \right) \right] \\
&+ \frac{1}{2} D \left[\frac{u_{i+1,j}^{n} + u_{i-1,j}^{n} + u_{i,j+1}^{n} + u_{i,j-1}^{n} - 4u_{i,j}^{n}}{\Delta^2} \right. \\
&\qquad \left. + \frac{u_{i+1,j}^{n+1} + u_{i-1,j}^{n+1} + u_{i,j+1}^{n+1} + u_{i,j-1}^{n+1} - 4u_{i,j}^{n+1}}{\Delta^2} \right] \qquad (8)
\end{aligned}$$

Równanie jest niejawne; aby je rozwiązać przenosimy $u_{i,j}^{n+1}$ na lewą stronę i w każdej iteracji (przejście $t_n \rightarrow t_{n+1}$) *relaksujemy* poniższe równanie 20 razy dla każdego węzła (i, j)

$$\begin{aligned}
\left(u_{i,j}^{n+1} \right)^{\mu+1} &= \left(\frac{1}{1 + \frac{2D\Delta t}{\Delta^2}} \right) \left\{ u_{i,j}^{n} - \frac{\Delta t}{2} v_{i,j}^{x} \left[\left(\frac{u_{i+1,j}^{n} - u_{i-1,j}^{n}}{2\Delta} \right) + \left(\frac{u_{i+1,j}^{n+1} - u_{i-1,j}^{n+1}}{2\Delta} \right)^{\mu} \right] \right. \\
&- \frac{\Delta t}{2} v_{i,j}^{y} \left[\left(\frac{u_{i,j+1}^{n} - u_{i,j-1}^{n}}{2\Delta} \right) + \left(\frac{u_{i,j+1}^{n+1} - u_{i,j-1}^{n+1}}{2\Delta} \right)^{\mu} \right] \\
&+ \frac{\Delta t}{2} D \left[\frac{u_{i+1,j}^{n} + u_{i-1,j}^{n} + u_{i,j+1}^{n} + u_{i,j-1}^{n} - 4u_{i,j}^{n}}{\Delta^2} \right. \\
&\qquad \left. \left. + \left(\frac{u_{i+1,j}^{n+1} + u_{i-1,j}^{n+1} + u_{i,j+1}^{n+1} + u_{i,j-1}^{n+1}}{\Delta^2} \right)^{\mu} \right] \right\} \qquad (9)
\end{aligned}$$

gdzie $\mu = 0, 1, 2, \ldots, 20$ jest numerem iteracji Picarda. Na starcie przyjmujemy dla wszystkich węzłów

$$\left(u_{i,j}^{n+1} \right)^{\mu=0} = u_{i,j}^{n} \qquad (10)$$

2.3 Warunki brzegowe

Na lewy i prawy brzeg nakładamy periodyczne warunki brzegowe. Oznacza to, że

- lewym sąsiadem węzła $(0, j)$ jest (n_x, j)
- prawym sąsiadem węzła (n_x, j) jest $(0, j)$

Dolny i górny brzeg oraz obrys zastawki omijamy (rów. adwekcji spełnia je automatycznie, bo $\vec{v}_{brzeg} = 0$, ale rów. dyfuzji już nie i musimy o tym pamiętać).

2.4 Pole prędkości

Pole prędkości wygenerujemy mając do dyspozycji funkcję strumienia (wczytywaną z pliku). Poza brzegami i zastawką prędkości wyliczamy z definicji, $v^x = \frac{\partial \psi}{\partial y}$ oraz $v^y = -\frac{\partial \psi}{\partial x}$, zastępując pochodne symetrycznymi ilorazami różnicowymi

$$v_{i,j}^{x} = \frac{\psi_{i,j+1} - \psi_{i,j-1}}{2\Delta}, \quad i = 1, 2, \ldots, n_x - 1, \quad j = 1, 2, \ldots, n_y - 1 \qquad (11)$$

$$v_{i,j}^{y} = -\frac{\psi_{i+1,j} - \psi_{i-1,j}}{2\Delta}, \quad i = 1, 2, \ldots, n_x - 1, \quad j = 1, 2, \ldots, n_y - 1 \qquad (12)$$

Na zastawce ustalamy

$$V_{i,j}^{x} = V_{i,j}^{y} = 0, \quad i = i_1, \ldots, i_2, \; j = 0, \ldots, j_1 \qquad (13)$$

na dolnym i górnym brzegu

$$V^x_{i,0} = V^y_{i,n_y} = 0, \quad i = 1, \ldots, n_x - 1 \tag{14}$$

a na lewym i prawym brzegu przepisujemy prędkości z sąsiednich węzłów

$$V^x_{0,j} = V^x_{1,j}, \quad j = 0, 1, \ldots, n_y \tag{15}$$

$$V^x_{n_x,j} = V^x_{n_x-1,j}, \quad j = 0, 1, \ldots, n_y \tag{16}$$

2.5 Warunek początkowy

Jako warunek początkowy $[u(x, y, t = 0)]$ przyjmujemy poniższy rozkład gęstości

$$u(x, y, t = 0) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x - x_A)^2 + (y - y_A)^2}{2\sigma^2}\right) \tag{17}$$

gdzie: (x_A, y_A) to położenie środka gęstości, a σ to rozmycie.

2.6 Algorytm dla równania AD

Pseudokod

```
inicjalizacja parametrów: n_x, n_y, i_1, i_2, j_1, D, Δ
utwórz tablice: u^0, u^1,ψ,v^x,v^y
wczytaj funkcję strumienia: ψ
wyznacz pole prędkości: v^x, v^y
wyznacz: v_max, Δt
inicjalizacja gęstości: u^0 = wzór (17)

FOR IT=1 TO ITMAX STEP 1 DO


  //start iteracji Picarda
  inicjalizacja kolejnego kroku:  u^1 = u^0
  FOR K=1 TO 20 STEP 1 DO
    FOR i=0 TO n_x STEP 1 DO
      FOR j=1 TO n_y-1 STEP 1 DO
        IF((i,j) ∈ (zastawka)) THEN
          CONTINUE
        ELSE IF(i==0 || i==n_x)THEN
          u^1_{i,j} = wzór(9) + periodyczne WB
        ELSE
          u^1_{i,j} = wzór(9)
        END IF
      END DO
    END DO
  END DO
  //zachowujemy rozwiązanie do następnego wywołania
  u^0 = u^1

  wyznacz i zapisz do pliku: c, x_sr


END DO
```

3 Zadania do wykonania

1. W obliczeniach użyć parametrów: $n_x = 400$, $n_y = 90$, $i_1 = 200$, $i_2 = 210$, $j_1 = 50$, $\Delta = 0.01$, $\sigma = 10 \cdot \Delta$, $x_A = 0.45$, $y_A = 0.45$.

2. Wczytać funkcję strumienia z pliku "psi.dat", format:

   `int i, int j, double` $\psi_{i,j}$

3. Wyznaczyć pole prędkości. Stworzyć mapy prędkości $v^x(x, y)$ oraz $v^y(x, y)$. (20 pkt)
   Uwaga: V^x powinno być wszędzie dodatnie (przepływ w prawą stronę), a V^y dodatnie przed zastawką i ujemne za nią.

   Wyznaczyć krok czasowy Δt według wzoru:

   $$\Delta t = \frac{\Delta}{4v_{max}}, \tag{18}$$

   gdzie v_{max} to maksymalny moduł wektora prędkości (należy najpierw znaleźć punkt i, j, dla którego moduł prędkości $|\vec{v}_{i,j}| = \sqrt{\left(v^x_{i,j}\right)^2 + \left(v^y_{i,j}\right)^2}$ jest największy).

4. Zaimplementować algorytm dla równania adwekcji-dyfuzji.

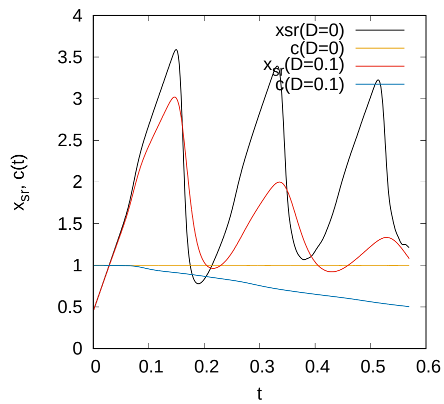
5. Wykonać symulację dla $D = 0$ (brak dyfuzji). Sporządzić wykresy:

   - całki gęstości $c(t)$
     $$c(t_n) = \sum_{i,j} u^n_{i,j} \cdot \Delta^2, \tag{19}$$
     Uwaga: wartość $c(t)$ może odchylać się od 1 o ± 0.005.
   - średniego położenia pakietu w kierunku 'x'
     $$x_{sr}(t_n) = \sum_{i,j} x_i \cdot u^n_{i,j} \cdot \Delta^2, \tag{20}$$
   - 5 map rozkładu $u(x, y, T_k)$ dla $T_k = k \cdot t_{max}/5$, $k = 1, \ldots, 5$, $t_{max} = IT\_MAX \cdot \Delta$. Wartość t_{max} (lub inaczej: IT_MAX) dobrać (empirycznie) tak aby na wykresie $x_{sr}(t)$ widoczne były 3 maksima.

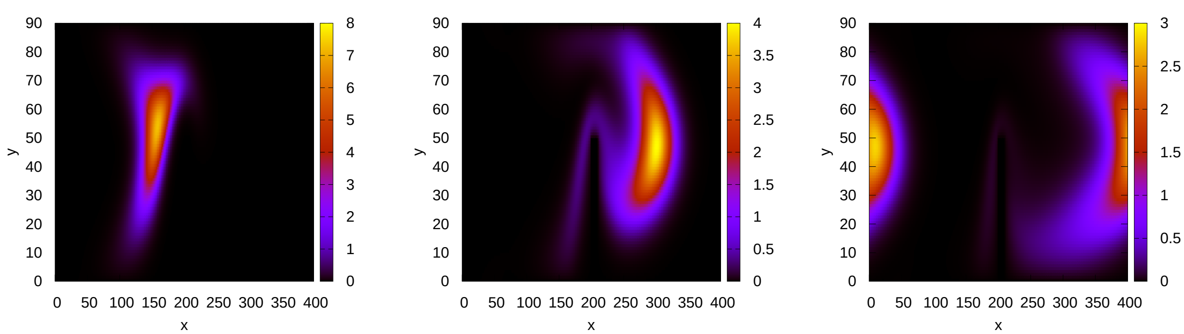
   Za komplet wyników (50 pkt)

6. Powtórzyć obliczenia z poprzedniego punktu dla $D = 0.1$, stworzyć wykresy $c(t)$, $x_{sr}(t)$ oraz mapy $u(x, y, T_K)$, $k = 1, \ldots, 5$. (30 pkt)
   Uwaga: wartość $c(t)$ powinna maleć w czasie (pakiet znika) ze względu na dyfuzję.

4 Przykładowe wyniki



Rysunek 2: $c(t)$ oraz $x_{sr}(t)$ z dyfuzją oraz bez. Przejście od maksimum do minimum oznacza że pakiet przechodzi przez prawy brzeg ($x = x_{max}$) i pojawia się na lewym brzegu ($x = 0$).



Rysunek 3: Rozkład gęstości w trzech wybranych chwilach czasowych dla $D = 0.1$. Na trzecim rysunku widać, jak pakiet przechodzi przez prawy brzeg i pojawia się na lewym brzegu.