

Programowanie i metody numeryczne

Projekt: Pierścień wirowy w ujęciu lagrangowskim

Paweł Jędrejko
pawel.jedrejko@fuw.edu.pl

1 Wstęp

Pierścienie wirowe są strukturami typowymi dla atmosferycznej konwekcji. Formują się na anomaliach wyporności (regionach podwyższonej temperatury) i, zdolne do przebicia inwersji, unoszą ciepło, wilgoć i zanieczyszczenia do wyższych obszarów. W końcowej fazie prowadzą do powstawania chmur typu cumulus albo cumulonimbus [3], wywierając duży wpływ na ich parametry. Z tego powodu, dynamika termalnych pierścieni wirowych ma duże znaczenie dla numerycznej prognozy pogody i predykcji klimatycznych [1].

Lagrangowskie (bezsiatkowe) ujęcie problemu skupia się na śledzeniu wybranych, materialnych wycinków płynu.

2 Opis zagadnienia

Rozważamy dwuwymiarowy przepływ, wymuszony siłą wyporu $\mathbf{b}$ (przybliżenie Boussinesqa), pomijając przy tym wszystkie zjawiska dyfuzyjne. Rozkład $\mathbf{b}$ jest jednorodny i nieciągły, ograniczony początkowo do okrągłej anomalii o promieniu $R = 1$:

$$\mathbf{b}(\mathbf{r}, t) \Big|_{t=0} = \begin{cases} b_0 \hat{y}, & \text{dla } |\mathbf{r}| \leq R \\ 0, & \text{dla } |\mathbf{r}| > R \end{cases}, \quad (1)$$

która jest w spoczynku:

$$\mathbf{u}(\mathbf{r}, t) \Big|_{t=0} = \mathbf{0}. \quad (2)$$

Przyjmijmy $b_0 = 1$. W późniejszym czasie wyporność jest unoszona przez nieściśliwe pole prędkości $\mathbf{u}$ i przy braku dyfuzji utrzymuje nieciągły rozkład.

Można pokazać, że stan układu jest jednoznacznie określony przez powierzchnię wirową ($\nabla \times \mathbf{u} \neq \mathbf{0}$), która powstaje na granicy faz. Jej kształt wygodnie jest opisać parametrycznie:

$$\mathbf{r}(\xi, t) = \begin{bmatrix} x(\xi, t) \\ y(\xi, t) \end{bmatrix}, \quad \xi \in [0, 2\pi) \quad (3)$$

z warunkiem początkowym:

$$\mathbf{r}(\xi, t) \Big|_{t=0} = R \begin{bmatrix} \cos(\xi) \\ \sin(\xi) \end{bmatrix}, \quad (4)$$

a cały układ rozwiązywać w ujęciu lagrangowskim. Lokalne natężenie powierzchni wirowej będziemy wyrażać przez $\gamma(\xi, t)$, które można rozumieć jako gęstość cyrkulacji na jednostkę parametru ξ (podobnie jak rotacja jest, z definicji, gęstością cyrkulacji na jednostkę pola powierzchni). Wartość γ jest proporcjonalna do nieciągłego skoku prędkości na powierzchni rozdziału faz i zależy od wyporności za pośrednictwem:

$$\frac{d\gamma}{dt} = -b_0 \frac{\partial y}{\partial \xi}. \quad (5)$$

Powyższe równanie jest formą równania wirowości i koncepcyjnie wiąże się z zachowaniem momentu pędu. Znając kształt i natężenie powierzchni wirowej, możemy wyznaczyć „indukowaną” prędkość z pomocą wzoru Biota-Savarta:

$$\mathbf{u}(\mathbf{r}_0, t) = \int_0^{2\pi} \frac{\gamma \hat{z} \times (\mathbf{r}_0 - \mathbf{r})}{2\pi(|\mathbf{r}_0 - \mathbf{r}|^2 + \delta^2)} d\xi, \quad (6)$$

gdzie parametr regularyzacji δ wprowadza tłumienie wysokich liczb falowych [2], ograniczając od dołu zakres skal obecnych w przepływie. Sugerowana wartość to $\delta = 0.1$. Mniejsze wartości będą skutkować bardziej skomplikowanym polem prędkości i wyższym kosztem obliczeniowym. Koncepcyjnie regularyzacja może być utożsamiona ze stałą, skończoną grubością powierzchni wirowej.

Pole prędkości, wyznaczone z równania (6) unosi powierzchnię wirową zgodnie z:

$$\frac{d\mathbf{r}}{dt} = \mathbf{u}. \quad (7)$$

Problem przyjmuje zatem postać ciągłej analogii problemu n ciał, gdzie powierzchnia wirowa unoszona jest przez przepływ, który sama indukuje. Ponadto, lokalne natężenie powierzchni nie jest stałe, lecz ewoluuje zależnie od kształtu, zgodnie z (5).

3 Zarys numerycznego rozwiązania

Kształt powierzchni powinien być zdyskretyzowany przy pomocy $N(t)$ węzłów. Aby utrzymać dokładność rozwiązania, należy wprowadzać nowe węzły w regionach, które są rozciągane. Wygodnie jest w związku z tym, reprezentować powierzchnię jako graf - zbiór węzłów połączonych krawędziami. Ponieważ δ jest miarą najmniejszych skal obecnych w przepływie, właściwie jest uzależnić od niej maksymalną długość krawędzi (Δl). Dla liniowej interpolacji między węzłami $\Delta l < 10\delta$ powinno być dostateczne.

W każdym kroku czasowym, na podstawie aktualnego kształtu powierzchni, należy wyznaczyć prędkość każdego z węzłów (6) oraz zmianę jego γ (5). Następnie należy przejść do kolejnego kroku (7) i, w razie potrzeby, poprawić dyskretyzację. Należy samodzielnie dobrać schematy numeryczne, wykorzystywane przy rozwiązaniu wspomnianych równań. Można przy tym korzystać z funkcji bibliotecznych, jednak ze świadomością, jaki schemat realizują.

Symulacja powinna obejmować czas do $T = 7$. Podczas jej trwania, należy monitorować wielkości kilku całek, których wartość można wyznaczyć analitycznie:

- całkowita cyrkulacja wokół pierścienia wirowego:

$$\Gamma = \int_0^{2\pi} \gamma d\xi = 0 \quad (8)$$

- całkowita siła wyporu:

$$B = \int_A b dA = \pi \quad (9)$$

(może się przydać twierdzenie Gaussa-Ostrogradskiego)

- całkowity impuls:

$$L = -\frac{1}{2} \int_0^{2\pi} x \gamma d\xi = B t. \quad (10)$$

4 Zarys implementacji

Kod powinien być napisany w sposób czytelny z deskryptywnymi nazwami zmiennych i funkcji. Komentarze są również mile widziane. W ramach możliwości powinien mieć modułową strukturę, podzieloną na funkcje. Poprawia to czytelność, ułatwia testowanie i pozwala unikać duplikacji. Poniżej funkcje, które muszą znaleźć się w kodzie:

- `biotSavart` – wyznacza prędkość w punkcie o zadanych współrzędnych; przyjmuje m.in. δ jako argument
- `computeSheetVel` – wyznacza prędkość we wszystkich węzłach powierzchni wirowej; odwołuje się do `biotSavart`
- `computeGammaSource` – wyznacza $d\gamma/dt$ we wszystkich węzłach
- `dSheetDt` – wyznacza zmianę w czasie wszystkich parametrów powierzchni x, y, γ , wywołując `computeSheetVel` i `computeGammaSource`

- **timeStepper** – wykonuje jeden krok w czasie, aktualizując parametry powierzchni wirowej. Funkcja, zgodnie z którą ewoluuje powierzchnia powinna być przyjmowana jako jeden z argumentów o domyślnej „wartości” równej **dSheetDt**. Długość kroku Δt powinien również być przyjmowany jako argument.
- **refineSheet** – wprowadza dodatkowe węzły wszędzie tam, gdzie jest to potrzebne; odwołuje się do **tooFarAway**
- **tooFarAway** – stwierdza czy należy wprowadzić nowy węzeł między dane dwa, sąsiednie węzły; kryterium odwołuje się do (fizycznej) długości powierzchni między węzłami, wynikającej ze sposobu interpolacji x i y . Długość nie może przekroczyć ustalonej Δl_{max} , uzależnionej od δ .
- **computeL**, **computeB**, **computeGamma** – wyznaczają wartości całek (10), (9), (8)
- **toFile** – zapisuje do pliku wszystkie dane potrzebne do narysowania powierzchni
- **plotSheet** – rysuje kształt powierzchni, kolorem zaznacza wartość γ

4.1 Testy

W celu zweryfikowania implementacji przeprowadzone powinny być testy opisane poniżej. W szczególnym przypadku, gdy któryś z testów nie nadaje się do wybranej metody numerycznej, można go zmodyfikować albo zastąpić innym. Ważne jednak, żeby test pozwalał stwierdzić czy metoda została zaimplementowana poprawnie.

1. **testBiotSavart** – dla powierzchni:

$$\mathbf{r}(\xi) = \begin{bmatrix} \cos(\xi) \\ \sin(\xi) \end{bmatrix}, \quad \gamma = -\cos(\xi), \quad (11)$$

przy $\delta = 0$ indukowaną prędkość można wyznaczyć analitycznie. Wybierając konkretny punkt, otrzymujemy referencyjne rozwiązanie:

$$\mathbf{r}_0 = \begin{bmatrix} 1/2 \\ 1 \end{bmatrix}, \quad \mathbf{u}_0 = \begin{bmatrix} 8/25 \\ 6/25 \end{bmatrix}.$$

Należy numerycznie wyznaczyć prędkość w tym punkcie, dyskretyzując powierzchnię (11) przy pomocy różnych liczb węzłów N . Następnie sprawdzić, czy błąd E całkowania (6) skaluje się zgodnie z teorią dla wykorzystanego schematu. Wynik powinien być przedstawiony na wykresie $\log_{10}(|E|)$ od $\log_{10}(N)$. Należy również wyznaczyć współczynnik nachylenia spodziewanej prostej (np. regresję liniową) i zestawić go z predykcją teoretyczną.

2. `testComputeGammaSource` – dla powierzchni (11), analitycznie otrzymujemy:

$$\frac{d\gamma}{dt} = -\cos(\xi)$$

W sposób podobny jak w 1. sprawdzić, czy średnia wartość bezwzględna błędu skaluje się zgodnie z teorią dla wybranego schematu.

3. `testTimeStepper` – dla jednowęzłowej ($N = 1$) „powierzchni”:

$$\mathbf{r}(\xi) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \gamma = 0, \quad (12)$$

wykonać N_t kroków w czasie o długości Δt , takich żeby końcowy czas był równy $T = 15$. Funkcja `timeStepper` powinna otrzymać jako argument funkcję, która następująco określa ewolucję powierzchni:

$$\mathbf{u} = \begin{bmatrix} -y \\ x \end{bmatrix}, \quad \frac{d\gamma}{dt} = x y.$$

Analityczne rozwiązanie dla czasu T to:

$$\mathbf{r} = \begin{bmatrix} \sin(T) \\ \cos(T) \end{bmatrix}, \quad \gamma = \frac{\sin^2(T)}{2}$$

Sprawdzić, czy przy różnych wartościach N_t (ale stałym T), błąd rozwiązania skaluje się zgodnie z teorią, osobno dla x , y i γ .

4. `testRefineSheet` – dana jest okrągła powierzchnia, jak (11), ale o promieniu ρ , zdyskretyzowana przy pomocy $N = 32$ równo rozstawionych węzłów. Przyjmując $\delta = 0.1$ wywołaj funkcję `refineSheet` dla dwóch powierzchni o promieniach odpowiednio równych:

$$\rho_1 = \frac{3\Delta N}{8\pi} l_{max}, \quad \rho_2 = \frac{3\Delta N}{4\pi} l_{max}.$$

Wykorzystując `computeGamma` wyznacz całkowitą cyrkulację przed i po re-dyskretyzacji obu powierzchni. Cyrkulacja nie może ulec zmianie. Przedstaw łącznie 4 rysunki – dla obu powierzchni przed i po. Zamieść na nich informację na temat dokładnej wartości Γ i liczby węzłów. Węzły powierzchni powinny być zaznaczone markerami.

5 Wyniki

W ramach projektu należy przedstawić uzyskaną ewolucję pierścienia wirowego oraz wyniki testów. Najlepiej, jeśli będą zebrane w jednym pliku (np. pdf) załączonym do kodu.

5.1 Ewolucja pierścienia wirowego

- a) Rysunki kształtu powierzchni wirowej dla reprezentatywnych kroków czasowych. Wartość γ powinna być zaznaczona kolorem wzdłuż powierzchni (odzworowanie nie musi być „ciągłe”; kilka kolorów z przypisanymi przedziałami γ jest w porządku). Każdy rysunek powinien mieć zaznaczony czas. Dobrze, jeśli wszystkie rysunki będą miały taką samą podziałkę na osiach, tak, żeby łatwo było je porównywać.
- b) Wykresy $\Gamma(t)$, $B(t)$, $L(t)$. Na liniach powinny być drobne markery, które zaznaczają konkretny pomiar.

5.2 Testy

- 1. `testBiotSavart`
- 2. `testComputeGammaSource`
- 3. `testTimeStepper`
- 4. `testRefineSheet`

Bibliografia

- [1] D. Grosvenor i in. “A study of the effect of overshooting deep convection on the water content of the TTL and lower stratosphere from Cloud Resolving Model simulations”. W: *Atmos. Chem. Phys.* (2007).
- [2] R. Krasny. “Desingularization of Periodic Vortex Sheet Roll-up”. W: *J. Comput. Phys.* (1985).
- [3] R. Plant i J-I. Yano. *Parametrization of Atmospheric Convection*. T. 1. Imperial College Press, 2016. Rozd. 1.