Numeryczne obliczanie pochodnych funkcji zadanej na sieci

Gabriel Wlazłowski

2 marca 2016

1 Problem 1D

Cel i założenia

Wyznaczyć numerycznie n-tą pochodną funkcji zespolonej $f(x)$.

Numerycznie funkcja jest zadana przez wektor liczb zespolonych o długości n_x, gdzie k-ty element odpowiada wartości funkcji f w punkcie $x_0 + k\Delta x$, tzn:

$$f_k \equiv f(x_0 + k\Delta x), \quad k = 0, 1, ..., n_x - 1 \tag{1}$$

gdzie Δx jest stałą sieci. Dziedziną funkcji jest przedział $x \in (x_0, x_0 + (n_x - 1)\Delta x)$. Zazwyczaj $x_0 = 0$. Mówimy, że funkcja zadana jest na *sieci* o rozmiarze n_x i stałej sieci Δx. Parametry sieci (n_x i Δx) są zazwyczaj parametrami algorytmu, które definiują dokładność obliczeń. Coraz większą dokładność obliczeń osiągamy poprzez zwiększanie rozmiaru sieci n_x i zmniejszanie stałej sieci Δx. Granicę: $n_x \to \infty$ i $\Delta x \to 0$ nazywamy *granicą kontinuum.*

Wykorzystanie transformaty Fouriera do obliczania pochodnych

Transformata Fouriera może być wykorzystana do obliczania pochodnych funkcji dowolnego rzędu. Dzięki algorytmowi FFT (ang. *Fast Fourier Transform*) w praktycznych zastosowaniach otrzymujemy bardzo elastyczną metodę (w zasadzie ten sam algorytm do obliczania pochodnej dowolnego rzędu), która charakteryzuje się jednocześnie dużą szybkością i duża dokładnością. Idea metody polega na zauważeniu, że operację liczenia pochodnej można wykonać analitycznie jeśli funkcję $f(x)$ przedstawimy jako:

$$f(x) = \int e^{ipx} \tilde{f}(p) \, \frac{dp}{2\pi}. \tag{2}$$

Powyższe wyrażenia przedstawia odwrotną transformatę Fouriera. Wtedy operacja n-tej pochodnej przybiera formę:

$$\frac{d^n}{dx^n} f(x) = \int (ip)^n e^{ipx} \tilde{f}(p) \, \frac{dp}{2\pi} \equiv \int e^{ipx} \tilde{F}^{(n)}(p) \, \frac{dp}{2\pi}, \tag{3}$$

gdzie w ostatnim wyrażeniu wprowadziłem funkcję $\tilde{F}^{(n)}(p) \equiv (ip)^n \tilde{f}(p)$. A więc n-ta pochodna dana jest przez odwrotną transformatę funkcji $\tilde{F}^{(n)}(p)$. Funkcję tą można łatwo skonstruować, gdyż $\tilde{f}(p)$ jest transformatą Fouriera funkcji $f(x)$, tzn.:

$$\tilde{f}(p) = \int e^{-ipx} f(x)\, dx. \tag{4}$$

Zwyczajowo mówimy, że transformata Fouriera zmienia reprezentację położeniową funkcji $f(x)$ na reprezentację pędową $\tilde{f}(p)$, natomiast odwrotna transformata Fouriera zmienia reprezentację pędową na reprezentację położeniową.

Ostatecznie algorytm można zapisać jako:

1. Obliczyć transformatę Fouriera funkcji $f(x)$. W wyniku tej operacji otrzymujemy funkcję $\tilde{f}(p)$.
2. Skonstruować funkcję $\tilde{F}^{(n)}(p) \equiv (ip)^n \tilde{f}(p)$.
3. Obliczyć odwrotną transformatę Fouriera funkcji $\tilde{F}^{(n)}(p)$. Wynikiem tej operacji jest funkcja $\dfrac{d^n}{dx^n} f(x)$.

Wskazówki praktyczne

W praktycznych zastosowaniach do obliczania transformaty Fouriera wykorzystujemy gotową bibliotekę FFT. Najpopularniejszą biblioteką (uważaną z najlepszą implementację algorytmu FFT) jest darmowa biblioteka FFTW (www.fftw.org). Dowolna biblioteka FFT oblicza dyskretną transformatę Fouriera (ang. *Dicrete Fourier Transform*, DFT), tzn:

$$Y_m = \sum_{n=0}^{n_x-1} e^{\pm i 2\pi nm/n_x} X_n, \tag{5}$$

gdzie X_n (Y_m) jest wektorem wejściowym (wyjściowym) o długości n_x.

Zauważmy, że równanie (4) po dyskretyzacji można zapisać jako [zakładam, że $x_0 = 0$, $x_n = n\Delta x$ oraz $f_n = f(x_n)$]:

$$\tilde{f}(p) = \int e^{-ipx} f(x)\, dx \approx \sum_{n=0}^{n_x-1} e^{-ipx_n} f_n\, \Delta x. \tag{6}$$

Równanie to stanie się bardzo podobne do równania (5) jeśli: $x_n = n\Delta x$ oraz $p_m = \frac{2\pi m}{n_x \Delta x}$. Zatem jeśli wykonamy DFT na tablicy która reprezentuję funkcję f_n to jako wynik otrzymamy tablicę w której zapisanie są wartości $\tilde{f}_m \equiv \tilde{f}(\frac{2\pi m}{n_x \Delta x})$.

Należy zwrócić uwagę na jeden szczegół, który jest istotny z praktycznego punktu widzenia. Rozważmy funkcję $e^{-ip_m x_n}$, gdzie $m > \frac{n_x}{2}$. Niech $m = \frac{n_x}{2} + k$, gdzie $k = 1, 2, \ldots \frac{n_x}{2} - 1$. Wtedy łatwo można pokazać, że

$$e^{-ip_m x_n} = e^{-i \frac{2\pi m}{n_x \Delta x} (\frac{n_x}{2} + k)\, n\Delta x} = e^{-i \frac{2\pi m}{n_x \Delta x} (k - \frac{n_x}{2})\, n\Delta x}. \tag{7}$$

Zatem pęd $\frac{2\pi m}{n_x \Delta x}(\frac{n_x}{2} + k)$ jest równoważny pędowi $\frac{2\pi m}{n_x \Delta x}(k - \frac{n_x}{2})$. Mówimy, że pędy są ograniczone do *pierwszej strefy Brillouina* - taki sam efekt pojawia się w fizyce ciała stałego, gdy atomy tworzą sieć. Ostatecznie, gdy wykonamy DFT na tablicy f_n to wyniku otrzymujemy tablicę $\tilde{f}_m = \tilde{f}(p_m)$, gdzie:

$$p_m = \begin{cases} \dfrac{2\pi m}{n_x \Delta x}, & \text{dla } m = 1, 2, \ldots \dfrac{n_x}{2}, \\ \dfrac{2\pi(m - n_x)}{n_x \Delta x}, & \text{dla } m = \dfrac{n_x}{2} + 1, \dfrac{n_x}{2} + 2, \ldots n_x - 1. \end{cases} \tag{8}$$

Największa możliwa wartość pędu to $\frac{\pi}{\Delta x}$ - nazywamy go pędem obcięcia.

Zatem z technicznego punktu widzenia algorytm obliczania n-tej pochodnej jest następujący:

1. Za pomocą funkcji bibliotecznej wykonujemy dyskretną transformatę Fouriera tablicy f_n. W wyniku otrzymujemy nową tablicę $\tilde{f}_m$ o takim samym rozmiarze co tablica wejściowa. Operację tą można wykonywać w miejscu, tzn. tablica wynikowa nadpisuje tablicę wejściową.

2. Przemnażamy elementy tablicy $\tilde{f}_m$ przez $(ip_m)^n$ gdzie pędy p_m dane są przez (8).

3. Za pomocą funkcji bibliotecznej wykonujemy odwrotną transformatę Fouriera tablicy $\tilde{f}_m$ (gdzie elementy są przemnożone przez $(ip_m)^n$). Operację tę również można wykonywać w miejscu.

4. Zazwyczaj wynikową tablicę, należy przemnożyć przez brakujący element normujący $\frac{1}{n_x}$. Jest to związane z tym, że jeśli wykonamy DFT jakiejś tablicy a następnie odwrotną DFT to otrzymamy wejściową tablicę przemnożoną przez ilość elementów tablicy (ten szczegół należy sprawdzić w dokumentacji wykorzystywanej biblioteki).

Wykorzystanie DFT do obliczania pochodnych automatycznie zakłada, że nasza funkcja jest periodyczna na przedziale $(0, n_x\Delta x)$ tzn. $f(x) = f(x+n_x\Delta x)$. Mówimy, że mamy *periodyczne* warunki brzegowe. Aby zminimalizować wpływ periodycznych warunków brzegowych należy zadbać aby sieć była na tyle duże, że wartości funkcji $f(x)$ na brzegach są zaniedbywalne.

Przedstawioną powyżej metodę można bardzo łatwo uogólnić na przypadek 2D i 3D.

2 Zadanie

Napisać skrypt obliczający numerycznie pierwszą i drugą pochodną funkcji $f(x) = e^{-x^2}$. Wynik porównać z wynikiem analitycznym. Sprawdzić dokładność obliczania pochodnych w funkcji parametrów sieci.