

Symulacja procesów ciągłych i algorytmy adaptacyjne

Projekt Tsunami

Patryk Czuchnowski, Michał Pędrak, Andrzej Wacławik

1. Wstęp

Celem projektu jest modyfikacja dostarczonego kodu symulującego falę tsunami tak, aby symulacja brała również pod uwagę ukształtowanie dna morskiego pobranego z pliku oraz przeprowadzenie eksperymentów i wizualizacja otrzymanych wyników.

2. Analiza wzoru i kodu przed modyfikacją

Kod przed modyfikacją reprezentuje następujące równanie:

$$\underbrace{u_t}_{\text{Next state}} = \underbrace{u_{t-1}}_{\text{Previous state}} + \underbrace{u_{t-1} - u_{t-2}}_{\text{States difference}} + \underbrace{\Delta t^2}_{\text{Time step squared}} \underbrace{g \nabla (u_{t-1} \nabla u_{t-1})}_{\text{Physics}}$$

Oznaczmy *Physics* jako X :

$$\mathbf{X} = \nabla(u \nabla u) = \partial_x(u \partial_x u) + \partial_y(u \partial_y u)$$

$$\partial_x(u \partial_x u) = u_x^2 + u u_{xx} \quad \partial_y(u \partial_y u) = u_y^2 + u u_{yy}$$

$$\mathbf{X} = u_x^2 + u u_{xx} + u_y^2 + u u_{yy}$$

Dla przypomnienia wzór na iloraz różnicowy oraz na drugą pochodną:

$$u'(x) \approx \frac{u(x+h) - u(x-h)}{2h} \quad u''(x) \approx \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}$$

Używamy ich do przybliżeń:

$$u_x^2 \approx X_1 = \left(\frac{u_{i+1,j} - u_{i-1,j}}{2h} \right)^2$$

$$u u_{xx} \approx X_2 = u_{i,j} \cdot \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2}$$

$$u_y^2 \approx X_3 = \left(\frac{u_{i,j+1} - u_{i,j-1}}{2h} \right)^2$$

$$u u_{yy} \approx X_4 = u_{i,j} \cdot \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2}$$

Powyższym wzorom odpowiada następujący, dostarczony nam kod:

```
h = 10/u.shape[0]
X_1 = ((u[i+1, j] - u[i-1, j]) / 2*h)**2
X_2 = u[i, j] * (
    (u[i+1, j]-2*u[i, j]+ u[i-1, j]) / h**2
)
X_3 = ((u[i, j+1] - u[i, j-1]) / 2*h)**2
X_4 = u[i, j] * (
    (u[i, j+1]-2*u[i, j]+ u[i, j-1]) / h**2
)
X = X_1 + X_2 + X_3 + X_4
```

3. Modyfikacja wzoru i kodu o zmienną z odpowiadającą głębokości

Chcemy wprowadzić zmienną z reprezentującą głębokość w danym miejscu tak jak w następującym wzorze:

$$\underbrace{u_t}_{\text{Next state}} = \underbrace{u_{t-1}}_{\text{Previous state}} + \underbrace{u_{t-1} - u_{t-2}}_{\text{States difference}} + \underbrace{\Delta t^2}_{\text{Time step squared}} \underbrace{g \nabla \left((u_{t-1} - z) \nabla u_{t-1} \right)}_{\text{Physics}}$$

Oznaczmy *Physics* jako *X*:

$$\mathbf{X} = \nabla((u - z) \nabla u)$$

Wprowadzamy zmienną pomocniczą *A*:

$$A = u - z$$

$$\mathbf{X} = \nabla(A \nabla u) = \partial_x(A u_x) + \partial_y(A u_y) = A_x u_x + A u_{xx} + A_y u_y + A u_{yy}$$

Ponownie przybliżamy ilorazami różnicowymi:

$$\begin{aligned} A_x u_x &\approx X_1 = \left(\frac{(u_{i+1,j} - z_{i+1,j}) - (u_{i-1,j} - z_{i-1,j})}{2h} \right) \left(\frac{u_{i+1,j} - u_{i-1,j}}{2h} \right) \\ A u_{xx} &\approx X_2 = (u_{i,j} - z_{i,j}) \cdot \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \\ A_y u_y &\approx X_3 = \left(\frac{(u_{i,j+1} - z_{i,j+1}) - (u_{i,j-1} - z_{i,j-1})}{2h} \right) \left(\frac{u_{i,j+1} - u_{i,j-1}}{2h} \right) \\ A u_{yy} &\approx X_4 = (u_{i,j} - z_{i,j}) \cdot \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} \end{aligned}$$

W ten sposób otrzymujemy dokładne wzory które możemy zmienić w kodzie:

```
h = 10/u.shape[0]
X_1 = (((u[i+1,j] - z[i+1,j]) - (u[i-1,j] - z[i-1,j])) / (2*h)) * ((u[i+1,j] - u[i-1,j]) / (2*h))
X_2 = (u[i,j] - z[i,j]) * (u[i+1,j] - 2 * u[i,j] + u[i-1,j]) / (h**2)
X_3 = (((u[i,j+1] - z[i,j+1]) - (u[i,j-1] - z[i,j-1])) / (2*h)) * ((u[i,j+1] - u[i,j-1]) / (2*h))
X_4 = (u[i,j] - z[i,j]) * ((u[i,j+1] - 2 * u[i,j] + u[i,j-1]) / (h**2))

X = X_1 + X_2 + X_3 + X_4
```

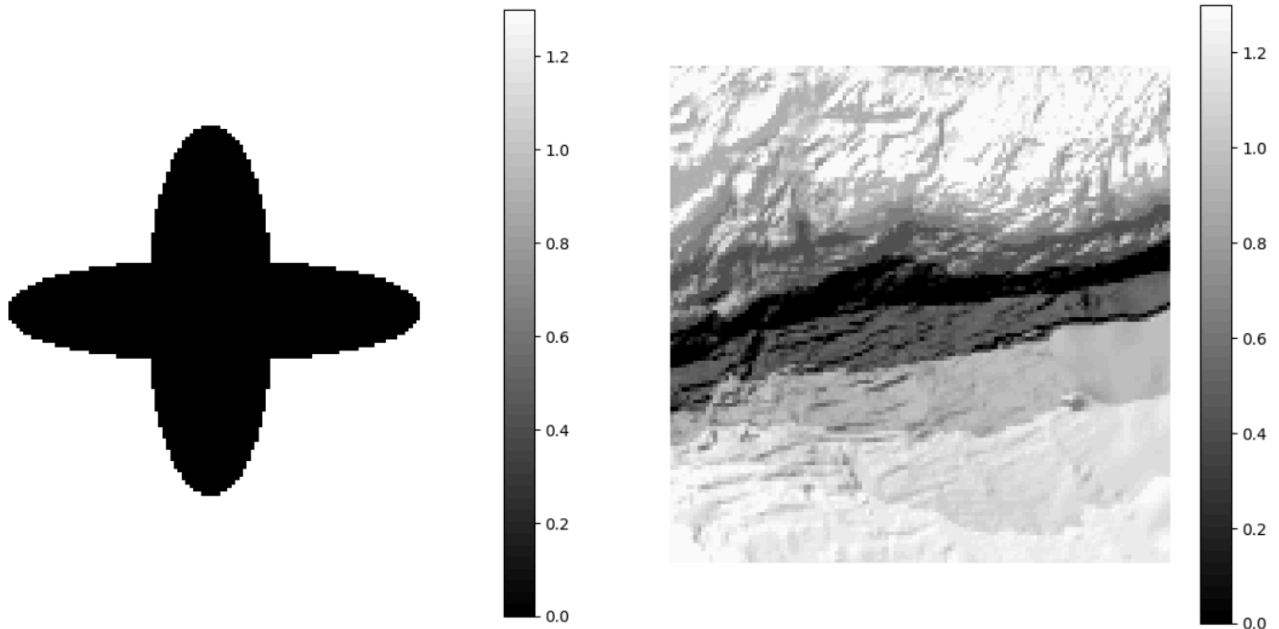
4. Wczytywanie terenu

Stworzyliśmy również funkcję która pozwala nam na wczytywanie ukształtowania dna morskiego ze zdjęcia:

```
def load_terrain(filename, normalize_val):
    img = Image.open(filename).convert("L") # convert to 8 bit grayscale
    z = np.array(img, dtype=np.float32)
    max_val = np.max(z)
    z = (z / max_val) * normalize_val
    return z
```

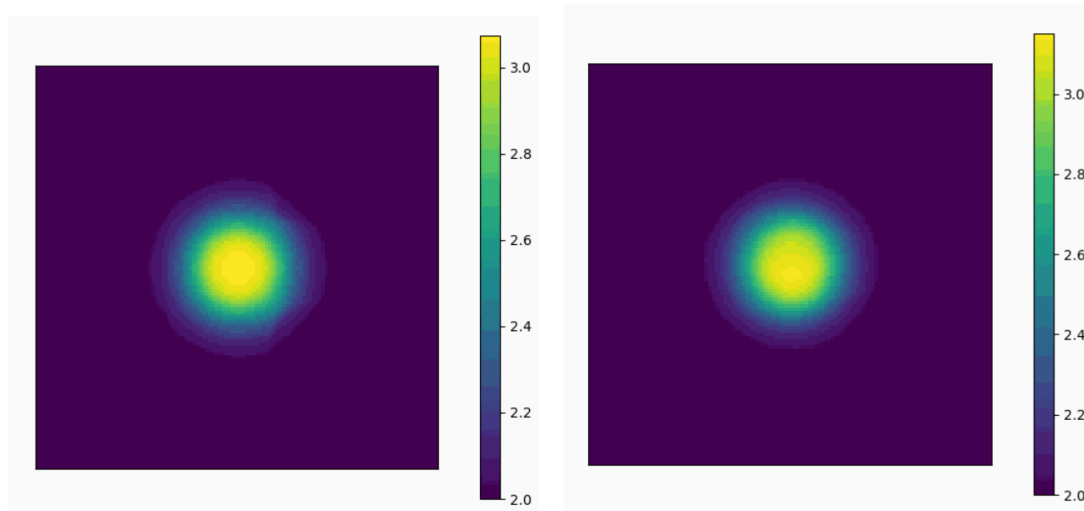
5. Bitmapy powierzchni dna morskiego

Do symulacji utworzyliśmy dwie bitmapy dna morskiego o rozmiarach 128 na 128 pikseli - jednej sztucznej oraz drugiej naturalnej, obie są zamieszone poniżej wraz ze skalą wysokości w [j. u.].

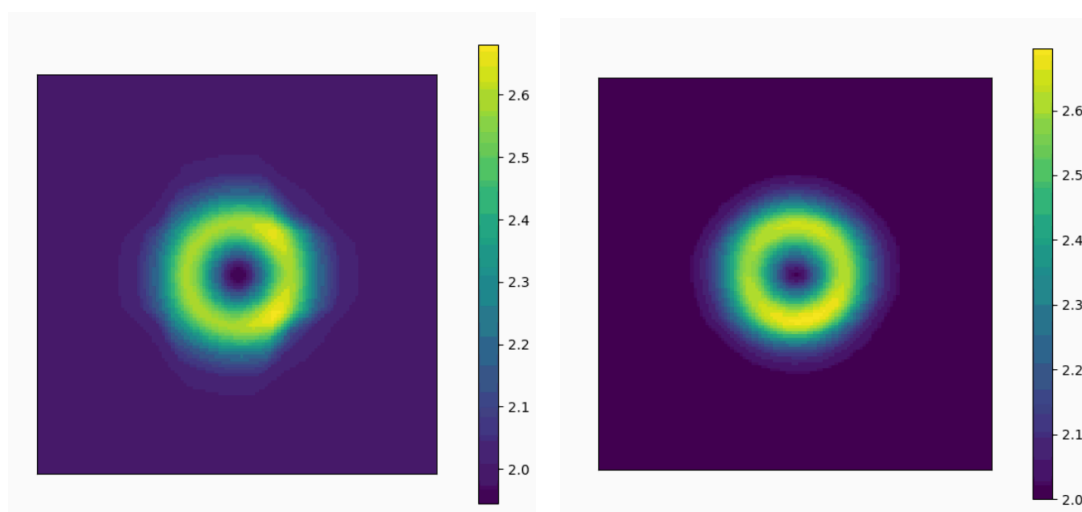


6. Przeprowadzone symulacje

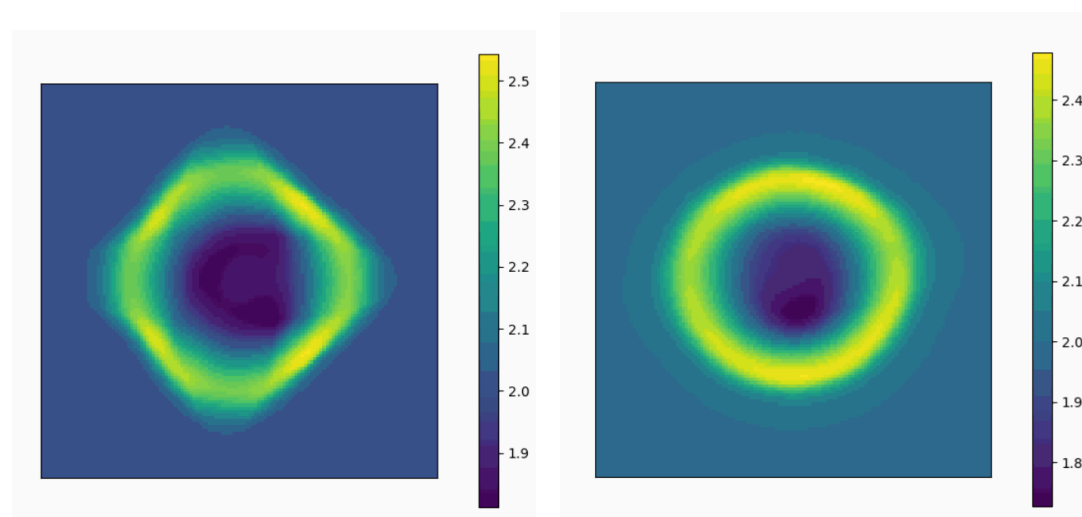
Symulacje były przeprowadzane z rozmiarem 128 na 128 i krokiem czasowym równym 0,001. Poniżej znajdują się wybrane z otrzymanych klatek z utworzonych gifów. Dla każdego wybranego kroku są 2 klatki - jedna dla dna sztucznego (po lewej) oraz druga dla dna naturalnego (po prawej), wraz ze skalą wysokości w [j. u.].



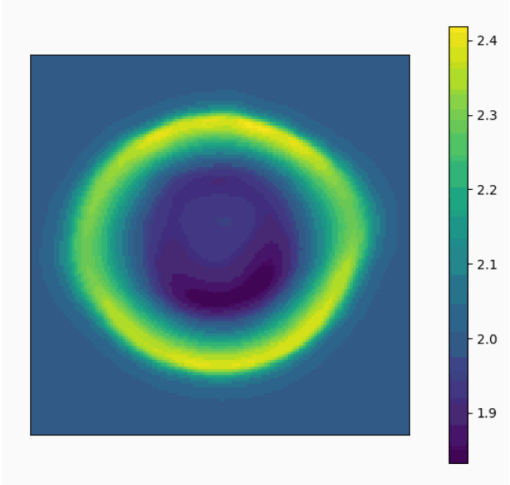
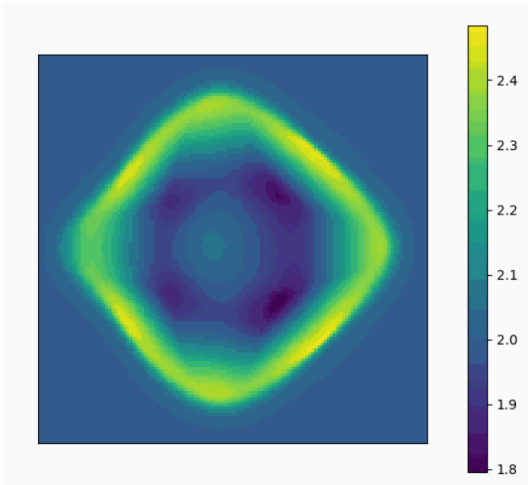
krok 100



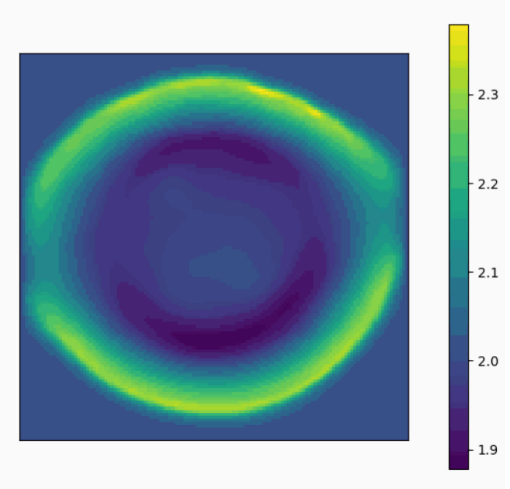
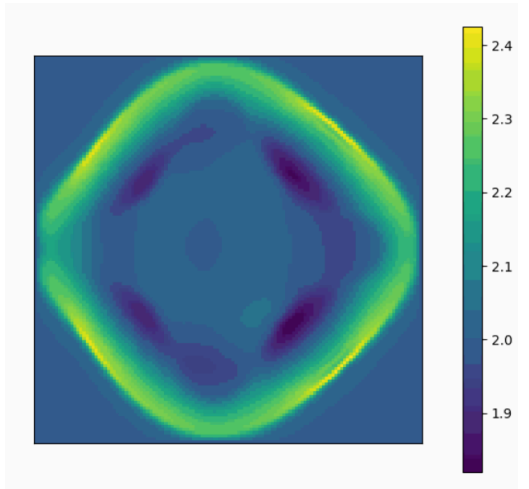
krok 200



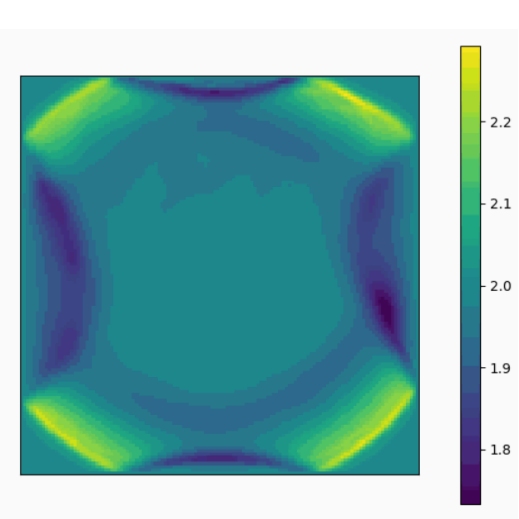
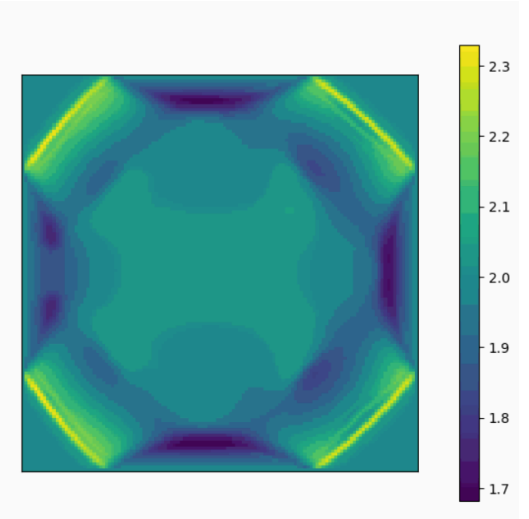
krok 500



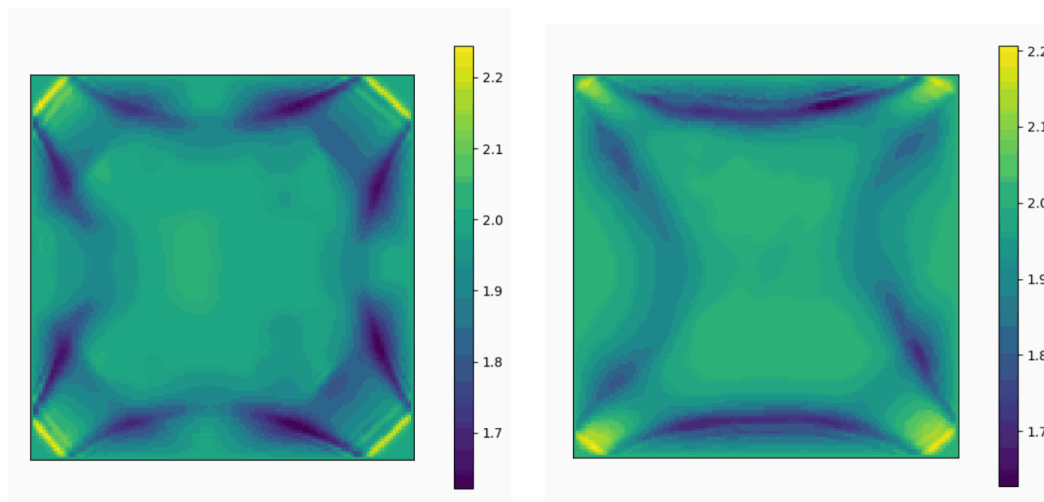
krok 700



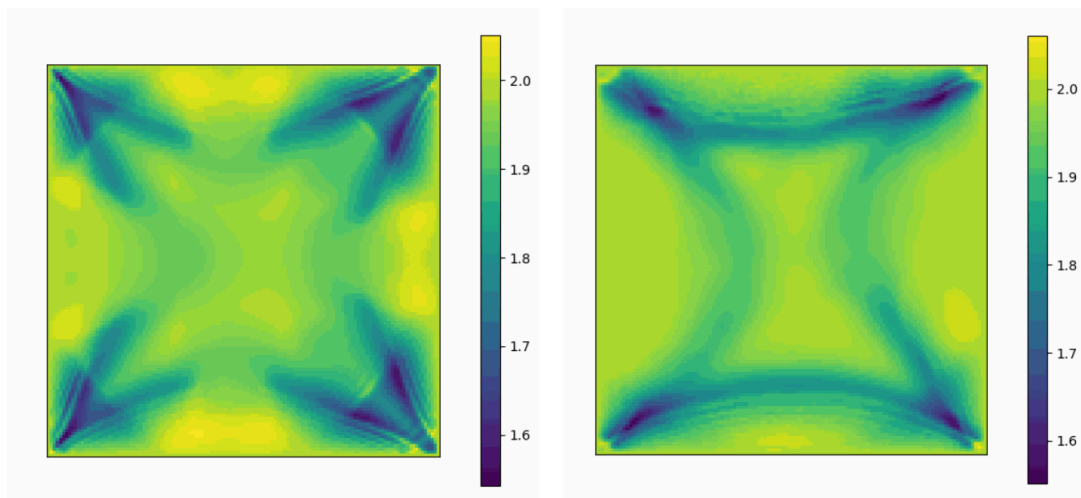
krok 1000



krok 1400

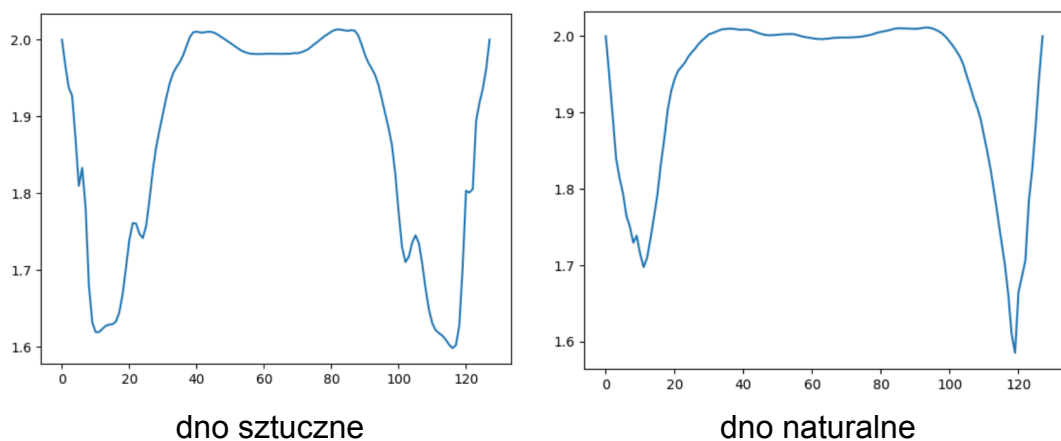


krok 1700



krok 2000

Poniżej znajdują się funkcje przekroju powierzchni fali po ostatnim kroku symulacji dla wiersza nr 10 siatki (blisko brzegu planszy):



7. Wnioski

- Dodanie zmiennej reprezentującej głębokość dna pozwoliło uzyskać bardziej realistyczną propagację fali

- Ukształtowanie dna morskiego wpływa na prędkość i wysokość fali tsunami
- Ostre zmiany głębokości prowadzą do odbić i zakłóceń w kształcie fali
- Naturalne, łagodniejsze dna powodują bardziej równomierne rozchodzenie się fali