

3 Równanie falowe

Rozwiąż dwuwymiarowe równanie falowe:

$$\partial_t^2 u(t, x, y) = c_x^2 \partial_x^2 u(t, x, y) + c_y^2 \partial_y^2 u(t, x, y) \quad (10)$$

na obszarze:

$$\left(\left|\frac{x}{a}\right|\right)^c + \left(\left|\frac{y}{b}\right|\right)^c \leq r \quad (11)$$

(poza tym obszarem $u(t, x, y) = 0$), z warunkami początkowymi:

$$u(0, x, y) = f(x, y) \quad (12)$$

$$u'(0, x, y) = g(x, y) \quad (13)$$

Skorzystaj z metody różnic skończonych, przybliżając drugie pochodne przez:

$$\partial_t^2 u(t, x, y) = \frac{u(t+k, x, y) - 2u(t, x, y) + u(t-k, x, y)}{k^2} = \frac{u_{i,j}^{n+1} - 2u_{i,j}^n + u_{i,j}^{n-1}}{k^2} \quad (14)$$

$$\partial_x^2 u(t, x, y) = \frac{u(t, x+h, y) - 2u(t, x, y) + u(t, x-h, y)}{h^2} = \frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{h^2} \quad (15)$$

$\partial_y^2 u$ - obliczamy analogicznie do $\partial_x^2 u$. Żeby algorytm był stabilny, warunek Couranta–Friedrichsa–Lewy’ego powinien być spełniony:

$$(c_x + c_y) \frac{k}{h} \leq 1. \quad (16)$$

Zauważmy, że do obliczenia u_{ij}^{n+1} musimy znać u_{ij}^n i u_{ij}^{n-1} . Jak zatem obliczyć u_{ij}^1 ? Z warunków początkowych mamy $u_{ij}^0 = f(x)$. Do obliczenia u_{ij}^{-1} wykorzystamy drugi warunek początkowy:

$$\partial_t u(0, x, y) = g(x, y) = \frac{u(t+k, x, y) - u(t-k, x, y)}{2k} = \frac{u_{ij}^1 - u_{ij}^{-1}}{2k} \rightarrow u_{ij}^{-1} = u_{ij}^1 - 2kg(x, y) \quad (17)$$

Napisz program w jednej z dwóch wersji:

3.1 Wersja interaktywana

- wyświetla w czasie rzeczywistym heatmapę rozwiązania
- warunki początkowe i obszar rozwiązania mają być wczytywane przez wskaźniki (tj. konstruktor klasy ma brać za argumenty wskaźniki do odpowiednich funkcji)
- sprawdź działanie programu dla $f(x, y) = \frac{1}{x^2+y^2+1}$ i $g(x, y) = 0$
- ma następującą funkcjonalność: po naciśnięciu (CTRL + lpm) w obszarze rozwiązania, wygeneruje fluktuacje w $u(t, x, y)$ tj. ręcznie zmienia wartość $u(t, x_0, y_0)$, gdzie kliknięcie nastąpiło w punkt (x_0, y_0) . Sprawdź działanie dla $f(x, y) = 0 = g(x, y)$. Czy widać wzory interferencyjne? Czy widoczna jest zmiana fazy na granicy obszaru rozwiązania?
- ma guzik resetujący układ do stanu początkowego
- w czasie rzeczywistym modyfikuje region rozwiązania przez zmianę parametrów a, b i c .
- w czasie rzeczywistym modyfikuje prędkości c_x i c_y .
- w osobnym oknie wyświetla w czasie rzeczywistym przekrój rozwiązania $u(t, x_0 = const, y)$ (lub $u(t, x, y_0 = const)$). Dodaj możliwość zmiany x_0 (lub y_0)

3.2 Wersja pasywna

- wyświetla heatmapę rozwiązania dla różnych czasów albo tworzy animację z poszczególnych klatek
- warunki początkowe i obszar rozwiązania mają być wczytywane przez wskaźniki (tj. konstruktor klasy ma brać za argumenty wskaźniki do odpowiednich funkcji)
- sprawdź działanie programu dla $f(x, y) = \frac{1}{x^2+y^2+1}$ i $g(x, y) = 0$
- parametry a, b, c, c_x, c_y powinny być łatwo modyfikowalne.
- Rozwiąż następujące zagadnienie: Zaczynij od warunków początkowych $f(x, y) = 0 = g(x, y)$. Następnie dla pewnego czasu t_0 zaburzyć rozwiązanie $u(t, x, y)$ w dwóch punktach (x_0, y_0) i (x_1, y_1) (tj. ręcznie ustaw $u(t_0, x_0, y_0) = u(t_0, x_1, y_1) = const$) Czy widać wzory interferencyjne? Czy widoczna jest zmiana fazy na granicy obszaru rozwiązania?
- Dla dowolnych (nietrywialnych) warunków początkowych, sprawdź zachowanie układu, gdy od pewnego t_0 parametr c opisujący region rozwiązania się zmienia. Zmiana może być ciągła lub skokowa.
- Dla dowolnych (nietrywialnych) warunków początkowych, sprawdź zachowanie układu, kiedy prędkości c_x i c_y są różne.
- wyświetla przekrój rozwiązania $u(t, x_0 = const, y)$ (lub $u(t, x, y_0 = const)$).

3.3 Przykład

3.4 ImPlot i ImGui

Polecany narzędziem do napisania projektu jest wykorzystanie ImGui i ImPlot'a. Oprócz klasycznych metod wyświetlania, potrzebne funkcje do projektu to:

- `ImGui::SliderFloat("text", &val, -5.f, 5.f);`

wyświetla suwak, który zmienia wartość zmiennej `val` od -5 do 5 (po naciśnięciu CTRL+lpm na suwaku pojawia się możliwość ręcznego wpisania wartości)

- heatmapa -

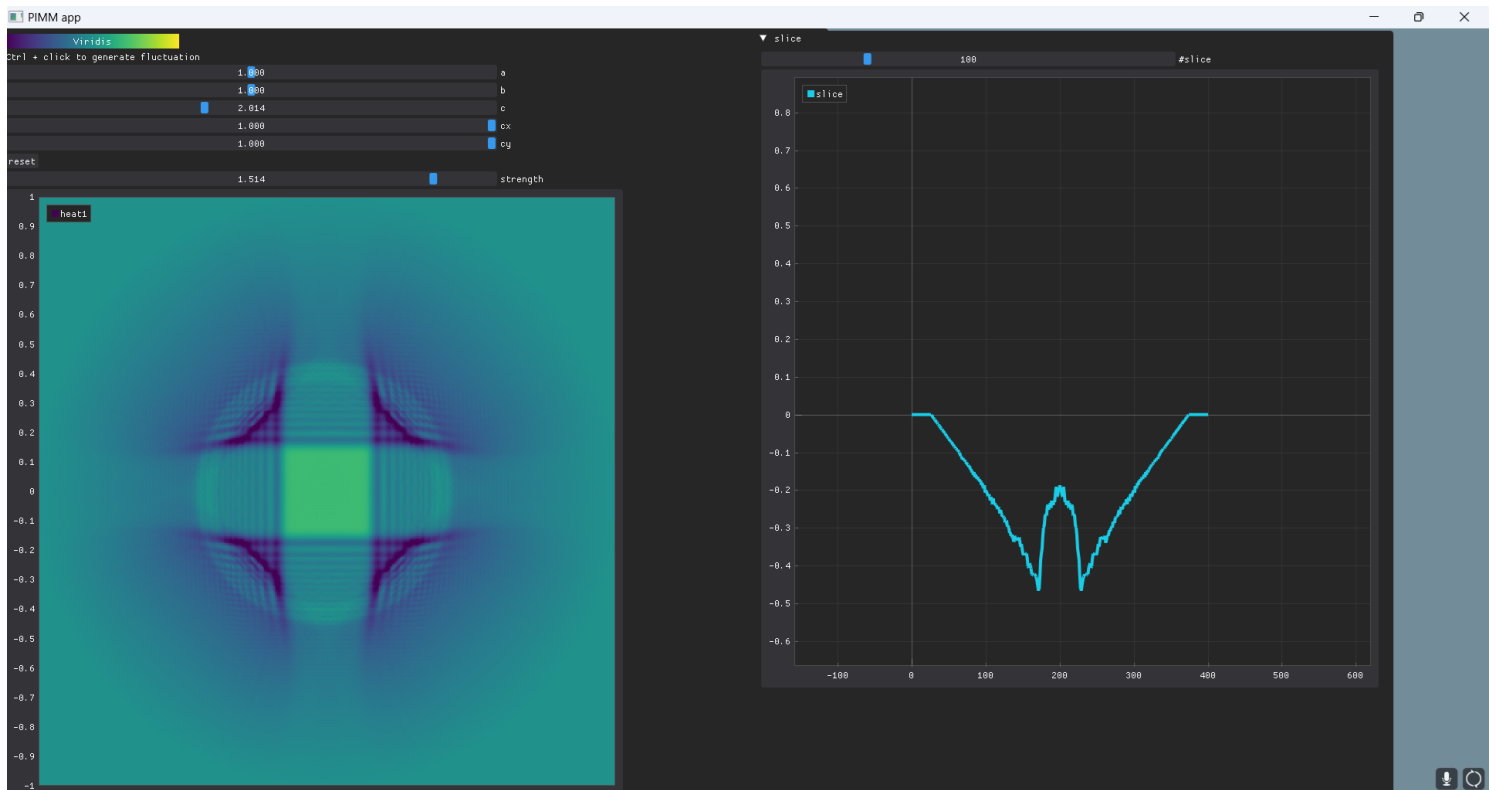


Figure 1: Przykładowe okno rozwiązania

```

static ImPlotColormap map = ImPlotColormap_Viridis; //kolorystyka

    //możliwość zmiany palety kolorów heatmapy
//wybór palety
if (ImPlot::ColormapButton(ImPlot::GetColormapName(map), ImVec2(225, 0), map)) {
    map = (map + 1) % ImPlot::GetColormapCount();
    ImPlot::BustColorCache("##Heatmapa");
}
//ustawienie palety
ImPlot::PushColormap(map);

//flagi - blokada okna, brak linii siatki, brak znaczników
ImPlotAxisFlags axes_flags = ImPlotAxisFlags_Lock | ImPlotAxisFlags_NoGridLines
                                | ImPlotAxisFlags_NoTickMarks;

//otwórz okno heatmapy, o rozmiarze 800x800
if (ImPlot::BeginPlot("##Heatmapa", ImVec2(800, 800))) {

    //ustaw flagi
    ImPlot::SetupAxes(nullptr, nullptr, axes_flags, axes_flags);
    //ustaw granice osi
    ImPlot::SetupAxesLimits(-1, 1, -1, 1);
    //wyrysuj heatmapę;
    ImPlot::PlotHeatmap("heat1", U2.data(), N, N, -1, 1, nullptr,
                        ImPlotPoint(-r, -r), ImPlotPoint(r, r));
}
    ImPlot::EndPlot();
}
//zdejmuje paletę kolorów;
ImPlot::PopColormap();

```

kolejne argumenty funkcji `ImPlot::PlotHeatmap` to: wyświetlana nazwa w legendzie; wskaźnik na pierwszy element jednowymiarowej tablicy/vektora; liczba kolumn; liczba wierszy; wartość minimalna; wartość maksymalna; wskaźnik do

etykiet; lewy dolny róg ekranu; prawy górny róg ekranu;

UWAGA! Jak widać do wyrysowania heatmapy musimy użyć jednowymiarowej tablicy/vectora. Dodatkowo należy sprawdzać, czy wartości w tablicy/vectorze nie są nan (heatmapa nie potrafi tego rysować).

- generowanie fluktuacji:

```
//jeśli: wskaźnik myszy na wykresie BB naciśnięty lpm BB naciśnięty ctrl
  if (ImPlot::IsPlotHovered() && ImGui::IsMouseClicked(0) && ImGui::GetIO().KeyCtrl) {
    //przypisz do pt współrzędne myszy
    ImPlotPoint pt = ImPlot::GetPlotMousePos();
    //korzystanie ze składowej x-owej: pt.x
    //korzystanie ze składowej y-owej: pt.y

  }
```

- przycisk:

```
if (ImGui::Button("reset")) {
    //instrukcje wykonane po naciśnięciu przycisku "reset"
}
```

3.4.1 Dodatkowe informacje

Osoby, które nie korzystały jeszcze ImPlota i ImGui, a chciałyby poznać to narzędzie, zachęcam do kontaktu - chętnie wytłumaczę wszystko od podstaw. Kontakt: mmarciniak26@uw.edu.pl, lub pokój 4.30.

3.5 Proponowana punktacja:

- napisanie poprawnej deklaracji klasy - zadeklarowanie wszystkich potrzebnych zmiennych i metod (10pkt)
- poprawna implementacja funkcji początkowych i dziedziny rozwiązania (5pkt)
- poprawna implementacja pierwszego kroku, tj. obliczenie u_{ij}^1 (5pkt)
- poprawne rozwiązanie równania falowego (10pkt)
- poprawne wyświetlanie heatmapy (10pkt)
- poprawne wyświetlanie przekroju rozwiązania (5pkt)
- (wersja interaktywna) poprawna implementacja zmiany parametrów a , b , c , c_x i c_y w czasie rzeczywistym (5pkt)
(wersja pasywna) poprawna implementacja przypadku ze zmiennym parametrem c oraz różnymi prędkościami c_x i c_y
- generowanie fluktuacji w u (5pkt)
- ogólna poprawność kodu (5pkt)