Program rozwiązujący funkcje zespolone

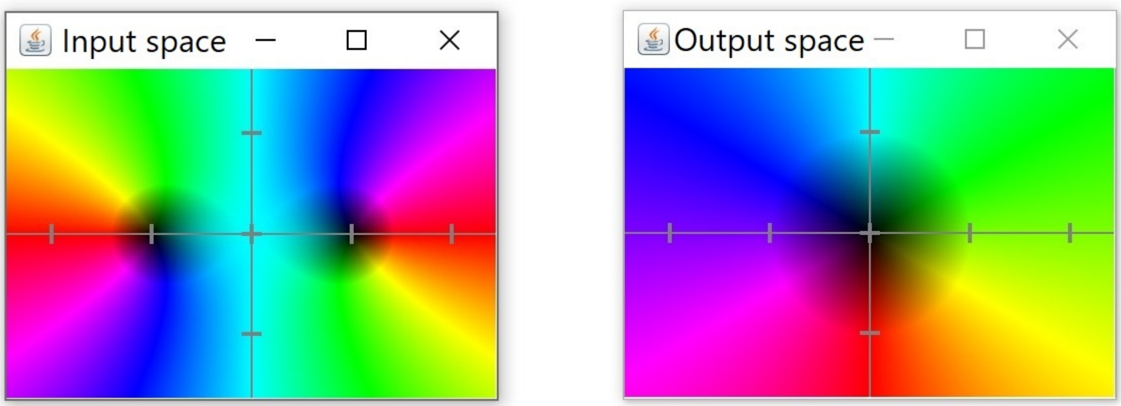
Piotr Machura, Kacper Ledwosiński

Cel główny

Celem projektu jest stworzenie kalkulatora znajdującego miejsca zerowe funkcji zespolonych zmiennej zespolonej. Użytkownik za pomocą interfejsu graficznego podaje funkcję, z której zostaje utworzony obiekt `Function` zawierający listę rozwiązań `solutions`. Następnie algorytm matematyczny wykorzystujący *"winding number" (indeks punktu względem krzywej, patrz sekcja "Podłoże matematyczne")* oraz rekurencyjną bisekcję na płaszczyźnie zespolonej **znajduje przybliżone miejsca zerowe** `Function` i umieszcza je w `solutions`, skąd mogą zostać **wyświetlone na ekran**.

Dodatkowo kalkulator wykorzystuje metodę kolorowania dziedziny aby wyświetlić obszary *input space* i *output space* funkcji *(patrz sekcja "Podłoże matematyczne)* oraz rysuje przeprowadzaną na płaszczyźnie zespolonej bisekcję w czasie rzeczywistym.

Podłoże matematyczne

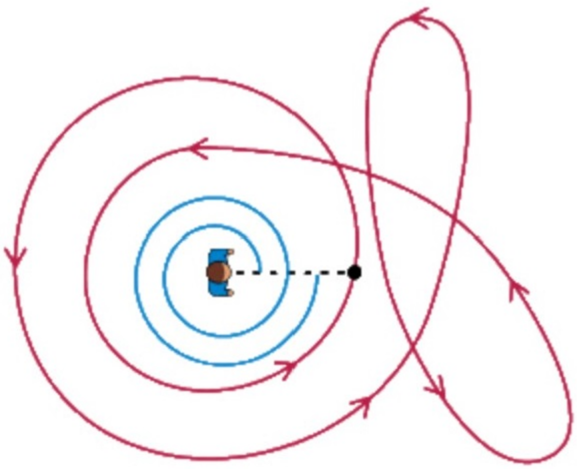


Kolorowanie dziedziny: z^2+1

Metoda kolorowania dziedziny

NArysowanie wykresu funkcji $f : \mathbb{C} \to \mathbb{C}$ ze względu na naturę liczb zespolonych wymagałoby czterowymiarowego układu współrzędnych, co jest niewykonalne w trójwymiarowej rzeczywistości. Z pomocą przychodzi metoda kolorowania dziedziny, polegajaca na nadaniu każdemu punktowi w *input space* koloru odpowiadającego fazie $\phi = arg(f(z))$ o jasności proporcjonalnej do $r = |f(z)|$. Pozwala to na łatwą wizualną ocenę przybliżonych miejsc zerowych.

Indeks punktu względem krzywej



Źródło: https://pl.wikipedia.org/wiki/Indeks_punktu_wzgl%C4%99dem_krzywej

Indeks punktu z_0 względem krzywej C jest to ilość okrążeń wykonywanych przez punkt z wokół z_0 przy jednym okrążeniu krzywej C. Przyjmuje on zatem jedynie wartości całkowite lub 0, jeśli z_0 nie zawiera się wewnatrz krzywej C. Na płaszczyźnie zespolonej określony jest jako:

$$W(C, z_0) = \frac{1}{2\pi i} \oint\limits_C \frac{dz}{z - z_0}$$

Przyjmijmy, że punktem z_0 jest punkt $0 + 0i$ w *output space*. Mamy zatem:

$$W(C) = \frac{1}{2\pi i} \oint\limits_C \frac{dz}{z}$$

Rozważmy prostokąt **R** (w rozumieniu: krzywa będąca krawędzią prostokąta) zawierający się w *input space* oraz oznaczmy jego obraz $C = f(R)$. Zatem jeśli indeks $W(C) \neq 0$, to 0 *output space* znajduje się wewnątrz C, a zatem **wewnątrz R znajduje się miejsce zerowe** f.

Liczbę obrotów *W* nazywać będziemy dalej zamiennie z *winding number* i jest to właściwość danego prostokąta **R**.

Algortym szukający miejsc zerowych

Algorytm zaczyna więc od narysowania prostokąta wystarczająco dużego, aby zawarło się w nim co najmniej jedno miejsce zerowe (*istnieje opcja pozwolenia użytkownikowi na wybór rozmiaru startowego prostokąta*). Następnie dzieli prostokąt na cztery mniejsze i sprawdza, czy *winding number* każdego z nich jest niezerowy. Jeśli jest **zerowy**, to wewnątrz nie ma miejsca zerowego i taki prostokąt zostaje **odrzucony**. Jeśli jest **niezerowy**, to taki prostokąt zostaje znowu podzielony na cztery itd. (*tzw. struktura quad tree*)

Rekurencja zatrzymuje się w momencie, gdy pola prostokątów o niezerowym *W* są **odpowiednio małe** i zwraca ich **środki** jako przybliżone miejsca zerowe funkcji.

Interfejs użytkownika

Cele dodatkowe

Konieczne:

- Użycie systemu kontroli wersji `git`

- Program napisany w `JavaFX`

Ewentualne:

- Program wielojęzyczny (wersja po angielsku i po polsku)
- Program dostępny w licencji *Open Source*