

Jeżeli zmienna/argument funkcji pojawia się w różnych miejscach programu, to o ile nie zostało powiedziane inaczej, jej definicja się nie zmienia.

1. Nietrywialne przykłady do obliczania grup homologii użyte w projekcie:

Torus, Butelka Kleina, 2-Sfera, Rzeczywista płaszczyzna rzutowa

2. Generowanie danych wejściowych

```
1 GenerateRandomSet[n_, k_, LowerLeft_, UpperRight_]
```

Parametry

- n - wymiar przestrzeni
- k - liczba generowanych punktów
- **LowerLeft, UpperRight** - lewy dolny i prawy górny róg hiperprostokąta

Funkcja zwraca listę punktów o współrzędnych całkowitych w przestrzeni \mathbb{R}^n .

```
1 GenerateDistanceMatrix[points_, r_]
```

Parametry

- **points** - zbiór wierzchołków kompleksu
- r - promień

Funkcja tworzy macierz, w której każdy element to odległość euklidesowa między dwoma punktami.

3. Konstrukcja kompleksu symplecjialnego

```
1 VietorisRipsComplex[points_, distanceMatrix_, r_, maxDim_ : 10]
```

Parametry

- **distanceMatrix** - macierz odległości
- **maxDim** - przybliżony szacunek wymiaru, do którego obliczenia będą sensowne. Domyślnie jest ustawiony na 10.

Funkcja generuje kompleks symplecjialny Vietorisa-Ripsa.

```
1 BuildSimplicialComplex[inputSimplices_, maxDim_ : 10]
```

Parametry

- **inputSimplices** - lista sympleksów, z których kompleks ma być zbudowany

Funkcja generuje kompleks symplecjialny na podstawie podanych sympleksów.

4. Funkcje odpowiadające za wizualizację

```
1 DrawNeighborhoods[showNeighborhoods_, vertices_, r_, showIndices_]
```

Parametry

- *showNeighborhoods* - zmienna logiczna, która kontroluje, czy mają być rysowane otoczenia 0-sympleksów
- *vertices* - wierzchołki kompleksu
- *r* - promień, który odpowiada za rozmiar otoczeń
- *showIndices* - lista punktów, które mają być uwzględnione w wizualizacji

Jeśli *showNeighborhoods* jest ustawione na *True*, to dla każdego punktu wskazanego w *showIndices* rysowana jest kula o środku w tym punkcie i promieniu $\frac{r}{2}$. Barwa każdego otoczenia jest ustawiona ze zredukowaną przezroczystością, co pozwala uwidocznąć sąsiadujące otoczenia.

```
1 DrawOneSimplices[vertices_, distanceMatrix_, r_, showIndices_]
```

```
2 DrawTwoSimplices[vertices_, distanceMatrix_, r_, showIndices_]
```

Funkcja rysuje 1-sympleksy/2-sympleksy pomiędzy punktami, które spełniają warunek Vietoris-Ripsa.

```
1 ManipulateComplex[vertices_, distanceMatrix_, R_]
```

Funkcja Manipulate, odpowiedzialna za interaktywną wizualizację kompleksu w przestrzeni 2D, która korzysta z powyższych funkcji.

5. Analiza struktury kompleksu

```
1 CountSimplices[complex_]
```

Parametry

- *complex* - kompleks utworzony za pomocą wcześniej podanej funkcji

Funkcja zwraca ilość sympleksów dla każdego wymiaru w podanym kompleksie symplecjonalnym.

```
1 GenerateGraph[complex_, n_]
```

Funkcja generuje reprezentację graficzną kompleksu symplecjonalnego w postaci grafu. Jest głównie używana by ocenić, czy liczba składowych spójnych (czyli β_0) jest poprawna.

```
1 ShowConnectedComponents[complex_]
```

Funkcja zwraca składowe spójne kompleksu, co jest potrzebne do wyliczenia β_0 :

```
1 ComplexConnectedComponents = ShowConnectedComponents[SimplicialComplex];
```

```
2 Betti0 = Length[ComplexConnectedComponents];
```

```
1 CountPotentiallyNontrivialHomologies[complex_]
```

Funkcja zliczająca, dla których wymiarów istnieją sympleksy w podanym kompleksie, w celu eliminacji redundantnych obliczeń dla trywialnych grup homologii.

6. Obliczenia macierzowe

1 `ColumnEchelonForm[matrix_]`

Funkcja sprowadza macierz do postaci kolumnowo-eszelonowej.

1 `BorderMatrix[complex_, k_]`

Funkcja konstruuje symplecjialny operator brzegu dla podanego wymiaru **k**.

1 `ranks = {};`

Tablica do przechowywania rzędów macierzy brzegu.

1 `ShowBorderMatrices[SimplicialComplex_, PotentiallyNontrivialHomologies_]`

Funkcja zwraca nietrywialne macierze homomorfizmu brzegu dla danego kompleksu symplecjialnego.

7. Wyznaczanie liczb Bettiego

1 `DisplayBettiNumbers[SimplicialComplex_, PotentiallyNontrivialHomologies_, ranks_, CardSn_]`

Parametry:

- *CardSn* = Normal[CountSimplices[SimplicialComplex]]
- *PotentiallyNontrivialHomologies*: CountPotentiallyNontrivialHomologies[SimplicialComplex]
- *CardSn* = Normal[CountSimplices[SimplicialComplex]]

Funkcja zlicza i wyświetla liczby Bettiego.