Computational topology
Homework 1 (due: November 9th 2025)

Each problem is worth a certain amount of points. Some problems are theoretical, others require you also submit the code (that conforms to the requirements given in the problem description). You may choose which problems to solve, **15 points is equal to 100%**.

You have to submit your solutions before the deadline as **one .zip file** to the appropriate mailbox at `https://ucilnica.fri.uni-lj.si/course/view.php?id=111` (near the top of the page).

This .zip file should contain:

1. **a NameSurname.pdf file written in LaTeX** and containing the solutions to the theoretical problems you have chosen as well as solutions and explanations for the programming problems (also make sure you sign your name on the top of the first page),
2. **.jl files** containing the code (one for each of the programming problems you have chosen).

1 Theoretical problems

1. **(3 points) Exploring different metrics.**

   View elements of $\mathbb{R}^2$ as two-component column vectors, e.g. $\mathbf{0} = [0,0]^\top$, $\mathbf{x} = [x_1, x_2]^\top$ and $\mathbf{y} = [y_1, y_2]^\top$. We define the metrics $\alpha, \beta, \gamma \colon \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$,

$$\alpha(\mathbf{x},\mathbf{y}) = \begin{cases} 0 & \text{if } \mathbf{x} = \mathbf{y}, \\ ||\mathbf{x}|| + ||\mathbf{y}|| & \text{otherwise,} \end{cases}$$

$$\beta(\mathbf{x},\mathbf{y}) = \begin{cases} ||\mathbf{x}-\mathbf{y}|| & \text{if } x_1 y_2 = x_2 y_1, \\ ||\mathbf{x}|| + ||\mathbf{y}|| & \text{otherwise,} \end{cases}$$

$$\gamma(\mathbf{x},\mathbf{y}) = \begin{cases} |y_2 - x_2| & \text{if } x_1 = y_1, \\ |x_2| + |y_1 - x_1| + |y_2| & \text{if } x_1 \neq y_1. \end{cases}$$

   (a) Determine the distances between the vectors $[1,2]^\top$, $[2,4]^\top$ and $[2,-1]^\top$ in all three metrics.

   (b) Draw the open balls $B(\mathbf{0},1)$, $B([0,1]^\top,2)$ and $B([1,2]^\top,1+\sqrt{5})$ in the metric α. Careful, the centre of the ball is always contained in it! Why?

   (c) Draw the open balls $B(\mathbf{0},1)$, $B([0,1]^\top,2)$ and $B([2,2]^\top,\sqrt{2})$ in the metric β.

   (d) Draw the open balls $B(\mathbf{0},1)$, $B([0,2]^\top,3)$ and $B([1,-1]^\top,2)$ in the metric γ.

   This might help:

   α is called the *postman metric* because the distance from the postman $P(x_1,x_2)$ to the customer $C(y_1,y_2)$ is obtained by adding the (standard Euclidean) distance from the postman to the post office $O(0,0)$ and the (standard Euclidean) distance from the post office to the customer.

   With the *radial metric* β, trips along the lines through origin are shorter than trips in other directions (if the two trips have the same length in the standard Euclidean metric).

   The *river metric* (or the *French railroad metric*) simulates a sistem where you can move along the vertical lines directly, but you can only move in the horizontal direction along the x-axis. So, if you wish to move from one vertical line to another, you have to first travel from the starting point in the vertical direction until you reach the x-axis, then move along it and finally travel in the vertical direction to the second point.

2. **(1 point) Discrete metric.**

   The *discrete metric* on a space X is defined as $d\colon X \times X \to \mathbb{R}$,

$$d(x,y) = \begin{cases} 0 & x = y, \\ 1 & \text{otherwise.} \end{cases}$$

   (a) Let $X = \mathbb{N}$. Recall that $B(x_0, r)$ denotes the open ball with centre x_0 and radius r and $\overline{B}(x_0, r)$ denotes the closed ball with centre x_0 and radius r. Describe the sets $B(1, \frac{1}{2})$, $B(2,1)$, $\overline{B}(3, \frac{1}{2})$ and $\overline{B}(4,1)$.

   (b) Given three pairwise distinct integers a, b and c, when is the triangle with vertices at a, b and c equilateral? Why?

3. **(2 points) Homeomorphic spaces.**

   Let $X_n = S^{n-1} \times [0,1] \subset \mathbb{R}^{n+1}$ and $Y_n = \{(x_1, \ldots, x_n) \in \mathbb{R}^n \,;\, 1 \le x_1^2 + \ldots + x_n^2 \le 4\}$. Draw X_n and Y_n for $n = 1$ and $n = 2$. Prove that X_n and Y_n are homeomorphic. (Hint: Prove it for $n = 2$ first, then generalize to $n \ge 1$.)

2 Programming problems

4. **(3 points) Deciding connectivity**

   Let G be a simple graph with n vertices and m edges.

   Write a simple algorithm that returns the connected components of the graph. You can use either breadth-first search or a depth-first search to traverse the graph.

   Your file `graphcomponents.jl` should contain a function `findcomponents(V, E)` that returns a list `[C1,C2,...,Ck]` of all components. Each component `Ci` is a list of vertices `[v1,v2,...,vk]`.

   The input will consist of

   (a) a list `V = [1,2,..., n]` of n vertices and

   (b) a list of m 2-tuples `E = [(v1,v2),...]` that represent the m edges.

   You may want to pre-process the data into a more suitable data structure to speed up the algorithm.

   Sample inputs:

```
V = [1,2,3,4,5,6,7,8]
E = [(1,2),(2,3),(1,3),(4,5),(5,6),(5,7),(6,7),(7,8)]
```

```
V = [1,2,3,4,5]
E = [(1,2),(1,3),(1,4),(1,5)]
```

   Corresponding outputs:

```
[[1, 2, 3], [4, 5, 6, 7, 8]]
```

```
[[1, 2, 3, 4, 5]]
```

Run these test cases to determine if your program works correctly (work out the final correct output on your own). Then put together two more test cases and include them in your report.

5. **(3 points) Line sweep triangulation in arbitrary direction**

   Given a point cloud $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, write two functions

   ```
   line_sweep(S, sweep_direction = [1; 0])
   ```
   and
   ```
   circle_sweep(S, circle_origin = (p, q))
   ```

   that returns a list of abstract simplices as tuples of indices of points of the given point cloud. By default the sweep direction for `line_sweep` should be along the x-axis, and the circle origin for `circle_sweep` should be the origin $(0, 0)$. Implement `sweep_direction` and `circle_origin` as optional parameters (i.e., assuming the default values if they are not passed). You can assume that the points (x_i, y_i) are in general position.

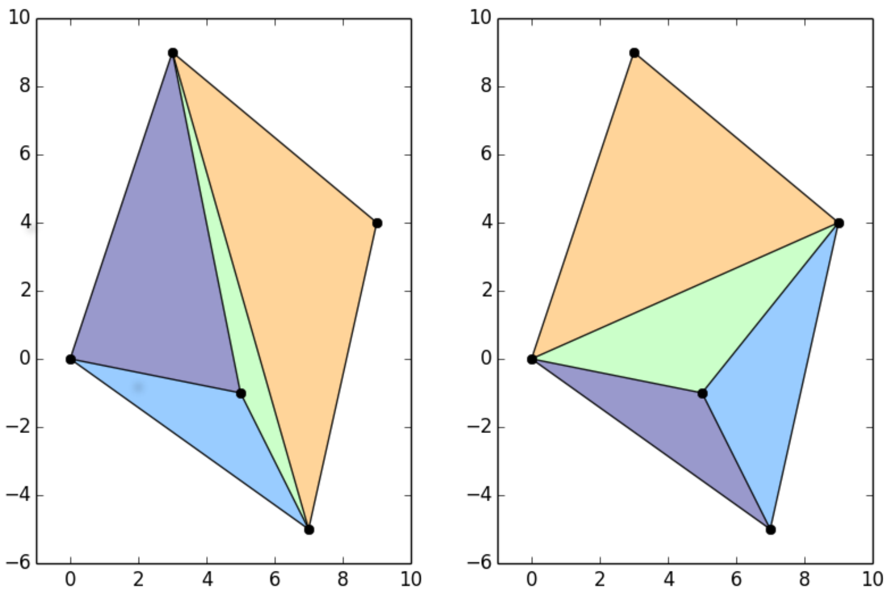
   Submit a file named `sweep_triangulation.jl` containing both functions.

   As a visual test plot the points, edges and triangles in the plane. (Your submitted functions should nevertheless only return specified outputs.)

   Sample inputs for `line_sweep`:

   ```
   S = [(0, 0), (3, 9), (5, -1), (9, 4), (7, -5)], sweep_direction = [1; 0]
   S = [(0, 0), (3, 9), (5, -1), (9, 4), (7, -5)], sweep_direction = [0; 1]
   ```

   Sample plots:

   Make up two more test cases consisting of at least 100 points and include the images of resulting triangulations (with a sweep direction and circle origin of your choice) in your report.

6. **(3 points) Jordan curve theorem**

   A simple closed curve in the plane is a connected curve with no self-intersections. We will consider a special case of a finite chain of straight line segments closing in a loop to form a simple closed polygonal curve.

The Jordan Curve Theorem states that every simple closed curve in $\mathbb{R}^2$ decomposes $\mathbb{R}^2$ into two components, the bounded inside and the unbounded outside.

Your file `jordan.jl` should contain a function `insideQ(P, T)`, that returns a boolean value: `true` if the point T lies inside the polygonal curve P and `false` otherwise. To determine this, count the number of intersections of an infinite ray starting at T with the segments of the curve P. You can assume that the vertices of P are in general position.

The curve P is given as a `Vector` of `Tuples` representing the vertices:

```
P = [(x1, y1), (x2, y2), ..., (xn, yn)].
```

Any two consecutive vertices are joined by a segment and the first vertex is joined to the last one to close the curve. The point T is given as a `Tuple` `T = (x0, y0)`.

Sample input:

```
T = (2.33, 0.66)
```

```
P = [(0.02, 0.10), (0.98, 0.05), (2.10, 1.03), (3.11, -1.23), (4.34, -0.35),
(4.56, 2.21), (2.95, 3.12), (2.90, 0.03), (1.89, 2.22)]
```

Test your code on this example and ensure the output is `true`. Explain how your algorithm works in your report. Also, find some nice examples to test your code, and include corresponding images in your report.