Marie Skłodowska-Curie Actions

CAD reverse engineering based on clustering and approximate implicitization

ESR 9

Andrea Raffo

April 30, 2018

SINTEF Digital, Oslo, Norway

Short CV

Education

- Bachelor's and Master's degree in Mathematics (2012-2016, University of Genova, Italy).
  Master Thesis: "*Compartmental Analysis of 4D MRI Data: Application to Renal Physiology*" (2016) developed during a 7-months Erasmus programme in Bergen (Norway).

Now:

- PhD candidate in Mathematics (SINTEF and University of Oslo, Norway)
  Start date: 1 November, 2016

Background

Past background:

Background

Past background:

- Numerical linear algebra

Background

Past background:

- Numerical linear algebra
- Geometry for applications

Background

Past background:

- Numerical linear algebra
- Geometry for applications
- Inverse problems and applications

Background

Past background:

- Numerical linear algebra
- Geometry for applications
- Inverse problems and applications
- Image processing

Background

Past background:

- Numerical linear algebra
- Geometry for applications
- Inverse problems and applications
- Image processing
- Applications of mathematics to medicine

Background

Past background:

- Numerical linear algebra
- Geometry for applications
- Inverse problems and applications
- Image processing
- Applications of mathematics to medicine
- ...

Background

Past background:

- Numerical linear algebra
- Geometry for applications
- Inverse problems and applications
- Image processing
- Applications of mathematics to medicine
- ...

enriched with courses at UiO:

Background

Past background:

- Numerical linear algebra
- Geometry for applications
- Inverse problems and applications
- Image processing
- Applications of mathematics to medicine
- ...

enriched with courses at UiO:

- Spline methods

Background

Past background:

- Numerical linear algebra
- Geometry for applications
- Inverse problems and applications
- Image processing
- Applications of mathematics to medicine
- ...

enriched with courses at UiO:

- Spline methods
- Algebraic Topology I

Background

Past background:

- Numerical linear algebra
- Geometry for applications
- Inverse problems and applications
- Image processing
- Applications of mathematics to medicine
- ...

enriched with courses at UiO:

- Spline methods
- Algebraic Topology I
- Statistical Learning: Advanced Regression and Classification

Background

Past background:

- Numerical linear algebra
- Geometry for applications
- Inverse problems and applications
- Image processing
- Applications of mathematics to medicine
- ...

enriched with courses at UiO:

- Spline methods
- Algebraic Topology I
- Statistical Learning: Advanced Regression and Classification
- Science, Ethics and Society

Background

Past background:

- Numerical linear algebra
- Geometry for applications
- Inverse problems and applications
- Image processing
- Applications of mathematics to medicine
- ...

enriched with courses at UiO:

- Spline methods
- Algebraic Topology I
- Statistical Learning: Advanced Regression and Classification
- Science, Ethics and Society
- ...

PhD project

In modern CAD systems, it is possible to identify two main representations of curves and surfaces: implicit and parametric.

PhD project

In modern CAD systems, it is possible to identify two main representations of curves and surfaces: implicit and parametric. Having both representations makes it possible to answer to a wide range of questions (e.g. intersection problems).

PhD project

In modern CAD systems, it is possible to identify two main representations of curves and surfaces: implicit and parametric. Having both representations makes it possible to answer to a wide range of questions (e.g. intersection problems). Classical approaches to implicitization are affected by computational and geometrical problems: (e.g. numerical instability, unwanted branches...).

PhD project

In modern CAD systems, it is possible to identify two main representations of curves and surfaces: implicit and parametric. Having both representations makes it possible to answer to a wide range of questions (e.g. intersection problems). Classical approaches to implicitization are affected by computational and geometrical problems: (e.g. numerical instability, unwanted branches...).
Purpose: Study of approximate implicitization and applications, considering possible improvements to existing methods (e.g. use of truly refinable geometric tools like LR B-splines).

PhD project

In modern CAD systems, it is possible to identify two main representations of curves and surfaces: implicit and parametric. Having both representations makes it possible to answer to a wide range of questions (e.g. intersection problems). Classical approaches to implicitization are affected by computational and geometrical problems: (e.g. numerical instability, unwanted branches...).

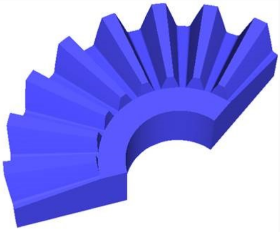
Purpose: Study of approximate implicitization and applications, considering possible improvements to existing methods (e.g. use of truly refinable geometric tools like LR B-splines).

Secondments:

- ATHENA (Athens, Greece, May 2018).
- RISC-Software (Linz, Austria, October 2018).

A problem of reverse engineering

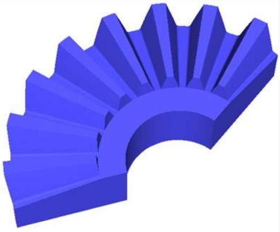
A problem of reverse engineering



Part of a NUGEAR developed by Stam S.r.l., Genova, Italy.

In industrial applications like computer aided design, geometric models are often represented numerically as polynomial splines or NURBS, even when they originate from primitive geometry.

A problem of reverse engineering



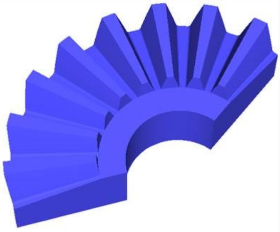
Part of a NUGEAR developed by Stam S.r.l., Genova, Italy.

In industrial applications like computer aided design, geometric models are often represented numerically as polynomial splines or NURBS, even when they originate from primitive geometry.

Problem

Extraction of information about the underlying geometry through reverse engineering.

A problem of reverse engineering



Part of a NUGEAR developed by Stam S.r.l., Genova, Italy.

In industrial applications like computer aided design, geometric models are often represented numerically as polynomial splines or NURBS, even when they originate from primitive geometry.

Problem

Extraction of information about the underlying geometry through reverse engineering.

Proposed approach

Combination of **clustering methods** with **approximate implicitization** to determine these primitive shapes and extract their features.

Clustering and Discrete Approximate Implicitization

What is clustering? An unsupervised statistical learning technique, aiming at partitioning a set $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \mathbb{R}^n$ such that all elements of a cluster share the same or closely related properties.

Clustering and Discrete Approximate Implicitization

What is clustering? An unsupervised statistical learning technique, aiming at partitioning a set $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \mathbb{R}^n$ such that all elements of a cluster share the same or closely related properties.
The grouping is performed by defining a notion of dissimilarity between data points. A *dissimilarity* on a set X is a function $d : X \times X \to \mathbb{R}$ such that for all $x, y \in X$:

1. $d(x, y) = d(y, x)$ (symmetry);
2. $d(x, y) \geq 0$ (positive definiteness);
3. $d(x, y) = 0$ iff $x = y$ (identity of indiscernibles).

Clustering and Discrete Approximate Implicitization

What is clustering? An unsupervised statistical learning technique, aiming at partitioning a set $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \mathbb{R}^n$ such that all elements of a cluster share the same or closely related properties.
The grouping is performed by defining a notion of dissimilarity between data points. A *dissimilarity* on a set X is a function $d : X \times X \to \mathbb{R}$ such that for all $x, y \in X$:

1. $d(x, y) = d(y, x)$ (symmetry);
2. $d(x, y) \geq 0$ (positive definiteness);
3. $d(x, y) = 0$ iff $x = y$ (identity of indiscernibles).

Once the dissimilarity (or distance) d between points of $\mathbb{R}^n$ has been chosen, it is necessary to define a generalization of d to compare clusters.

Clustering and Discrete Approximate Implicitization

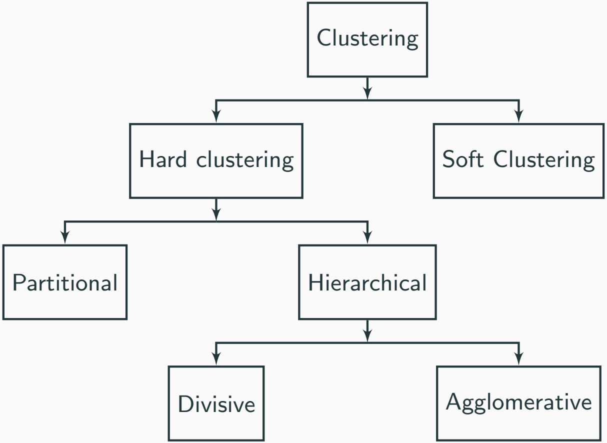
What is clustering? An unsupervised statistical learning technique, aiming at partitioning a set $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \mathbb{R}^n$ such that all elements of a cluster share the same or closely related properties.
The grouping is performed by defining a notion of dissimilarity between data points. A *dissimilarity* on a set X is a function $d : X \times X \to \mathbb{R}$ such that for all $x, y \in X$:

1. $d(x, y) = d(y, x)$ (symmetry);
2. $d(x, y) \geq 0$ (positive definiteness);
3. $d(x, y) = 0$ iff $x = y$ (identity of indiscernibles).

Once the dissimilarity (or distance) d between points of $\mathbb{R}^n$ has been chosen, it is necessary to define a generalization of d to compare clusters. Finally, an algorithm for grouping the data is designed via the defined d so as to reach a faithful clustering after a finite number of steps.

Clustering and Discrete Approximate Implicitization

Wide number of approaches in clustering theory...



Clustering

Hard clustering

Soft Clustering

Partitional

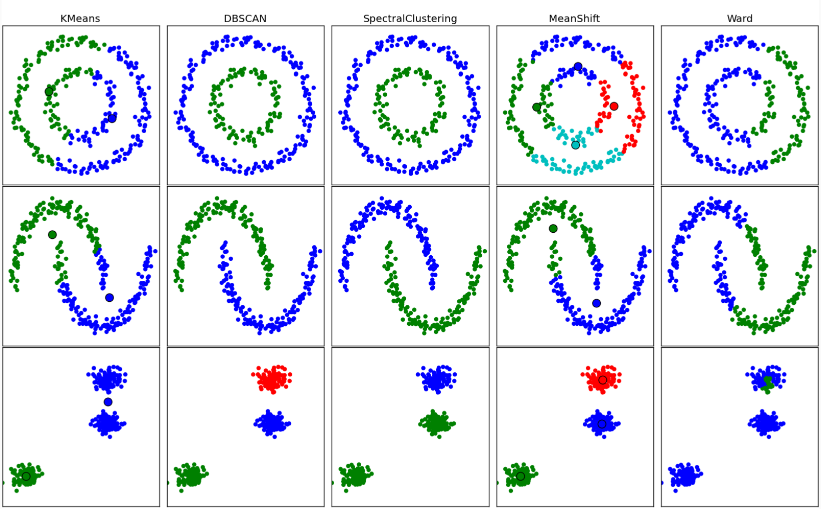
Hierarchical

Divisive

Agglomerative

Clustering and Discrete Approximate Implicitization

…and different solutions.



scikit-learn

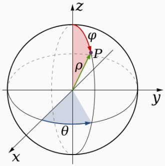
Clustering and Discrete Approximate Implicitization

Definition (Exact implicitization of a parametric hypersurface)

Let $\mathbf{p} : \Omega \subset \mathbb{R}^{n-1} \to \mathbb{R}^n$ be a hypersurface in $\mathbb{R}^n$. An exact implicitization of $\mathbf{p}$ is a n-variate polynomial q such that:

$$q(\mathbf{p}(\mathbf{s})) = 0, \quad \forall \mathbf{s} \in \Omega.$$

$x^2 + y^2 + z^2 = r^2 \longleftarrow$



$$\longrightarrow \begin{cases} x = r \sin\theta \cos\varphi, \\ y = r \sin\theta \sin\varphi, \\ z = r \cos\theta \end{cases}$$

Clustering and Discrete Approximate Implicitization

Assume that:

- $q(\mathbf{x}) = 0$ is an algebraic hypersurface of degree m;
- $\mathbf{p} = \mathbf{p}(\mathbf{s})$ is a polynomial or rational parametrization of (multi)degree $\mathbf{n}$ expressed in a given basis.

Then it follows that $q(\mathbf{p}(\mathbf{s}))$ can be expressed in a basis $\boldsymbol{\alpha}(\mathbf{s})$ for the polynomials of (multi)degree at most $m\mathbf{n}$. Explicitly,

$$q(\mathbf{p}(\mathbf{s})) = \boldsymbol{\alpha}(\mathbf{s})^T \mathbf{D}\mathbf{b},$$

where $\mathbf{b}$ is a column vector containing the unknown coefficients of the approximate implicit form.

Clustering and Discrete Approximate Implicitization

Now, we have

$$\min_{||\boldsymbol{b}||_2=1} \max_{\mathbf{s}\in\Omega} |q(\mathbf{p}(\mathbf{s}))| \leq \max_{\mathbf{s}\in\Omega} ||\boldsymbol{\alpha}(\mathbf{s})||_2 \sigma_{min},$$

where σ_{min} is the smallest singular value of $\mathbf{D}$.

Clustering and Discrete Approximate Implicitization

Now, we have

$$\min_{||\boldsymbol{b}||_2=1} \max_{\mathbf{s}\in\Omega} |q(\mathbf{p}(\mathbf{s}))| \leq \max_{\mathbf{s}\in\Omega} ||\boldsymbol{\alpha}(\mathbf{s})||_2 \sigma_{min},$$

where σ_{min} is the smallest singular value of $\mathbf{D}$.

Then:

- We consider the SVD decomposition $\mathbf{D} = \mathbf{U\Sigma U}^T$.
- The singular vector $\mathbf{v}_{min}$ corresponding to the smallest singular value offers the coefficients of the approximate implicit form in the chosen basis.

Clustering and Discrete Approximate Implicitization

One of the fastest and simplest implementations of approximate implicitization is based on the computation of the matrix $\mathbf{D}$:

$$D_{i,j} = \pi_j(\mathbf{p}(\mathbf{s}_i)), \quad j = 1, \ldots, m \text{ and } i = 1, \ldots, N, \tag{1}$$

where:

- $\{\mathbf{s}_i\}_{i=1}^{N} \subset \Omega$ are N given parameters in general position.
- $\{\pi_j\}_j$ is the basis of the set of n-variate polynomials of total degree degree at most m.

$$\mathbf{D} = \begin{array}{c} \\ \mathbf{p}(s_1) \\ \mathbf{p}(s_2) \\ \vdots \\ \mathbf{p}(s_N) \end{array} \begin{array}{c} \begin{array}{cccc} 1 & x & y & \ldots \end{array} \\ \begin{bmatrix} 1 & x(s_1) & y(s_1) & \ldots \\ 1 & x(s_2) & y(s_2) & \ldots \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x(s_N) & y(s_N) & \ldots \end{bmatrix} \end{array}$$

It follows that

$$\min_{\|\boldsymbol{b}\|_2=1} \max_{\mathbf{s}\in\Omega} |q(\mathbf{p}(\mathbf{s}))| \leq \max_{\mathbf{s}\in\Omega} \Lambda(\boldsymbol{\alpha})\sigma_{min},$$

where $\Lambda(\boldsymbol{\alpha})$ is the Lebesgue constant from interpolation theory defined by $\Lambda(\boldsymbol{\alpha}) := \max_{t\in\Omega}\|\boldsymbol{\alpha}(\mathbf{s})\|_1$.

It follows that

$$\min_{\|\boldsymbol{b}\|_2=1} \max_{\mathbf{s}\in\Omega} |q(\mathbf{p}(\mathbf{s}))| \leq \max_{\mathbf{s}\in\Omega} \Lambda(\boldsymbol{\alpha})\sigma_{min},$$

where $\Lambda(\boldsymbol{\alpha})$ is the Lebesgue constant from interpolation theory defined by $\Lambda(\boldsymbol{\alpha}) := \max_{t\in\Omega}\|\boldsymbol{\alpha}(\mathbf{s})\|_1$.

Then:

- We consider the SVD decomposition $\mathbf{D} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^T$.
- The singular vector $\mathbf{v}_{min}$ corresponding to the smallest singular value offers the coefficients of the discrete approximate implicit form in the chosen basis.

The algorithm

Input and output

Input

A set of segments X in $\mathbb{R}^2$ (patches in $\mathbb{R}^n$, $n > 2$).

Output

A partition of X such as segments in the same cluster comes from the same primitive shape (e.g. straight line, conic in $\mathbb{R}^2$, quadric in $\mathbb{R}^3$, $\ldots$)

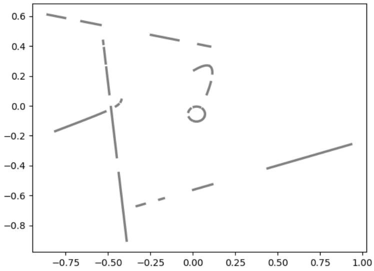


Figure 1: Initial dataset.

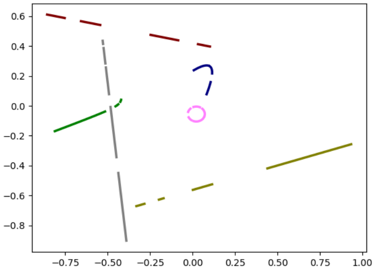


Figure 2: Classified dataset.

Step 1. Preprocessing step and calibration

Remark

Given a polynomial or rational patch τ of implicit degree d:

- *Discrete approximate implicitization of degree $\bar{d} < d$ provides the coefficients of an approximate implicit form, together with the smallest singular value $\sigma_{min}(\tau) > 0$ that measures the accuracy of the approximation.*
- *Discrete approximate implicitization of degree $\bar{d} = d$ provides the coefficients of the exact implicit form, together with a null smallest singular value ($\sigma_{min}(\tau) = 0$).*

Therefore, τ has implicit degree d if and only if $\sigma_{min}(\tau) = 0$.

Step 1. Preprocessing step and calibration

Remark

Given a polynomial or rational patch τ of implicit degree d:

- *Discrete approximate implicitization of degree $\bar{d} < d$ provides the coefficients of an approximate implicit form, together with the smallest singular value $\sigma_{min}(\tau) > 0$ that measures the accuracy of the approximation.*
- *Discrete approximate implicitization of degree $\bar{d} = d$ provides the coefficients of the exact implicit form, together with a null smallest singular value ($\sigma_{min}(\tau) = 0$).*

Therefore, τ has implicit degree d if and only if $\sigma_{min}(\tau) = 0$.

Weaker criterion in floating-point arithmetic:

$$\sigma_{min}(\tau) < \xi := \xi(d),$$

where the parameter ξ is computed on a training set.

Step 1. Preprocessing step and calibration

At the end of step 1, the set X is partitioned into $X = \cup_i X_i$, where X_i denotes the set of segments (or patches) of implicit degree i.

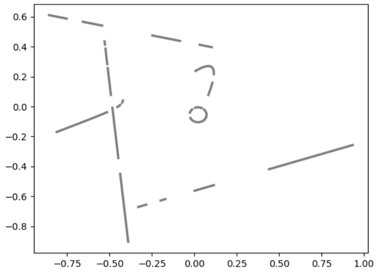


Figure 3: Initial dataset.

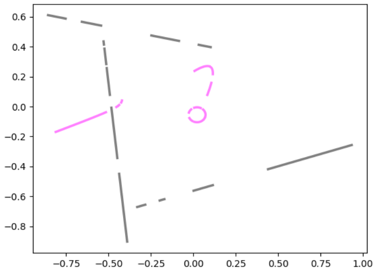


Figure 4: First clustering step.

Step 2. Definition of the candidate dissimilariy

We start defining the candidate dissimilarity d. We will apply a agglomerative approach separately to each X_i.

Step 2. Definition of the candidate dissimilariy

We start defining the candidate dissimilarity d. We will apply a agglomerative approach separately to each X_i.
We define the map $d : X \times X \to [0, +\infty)$ as follows:

Step 2. Definition of the candidate dissimilariy

We start defining the candidate dissimilarity d. We will apply a agglomerative approach separately to each X_i.
We define the map $d : X \times X \to [0, +\infty)$ as follows:

1. For each patch $\tau \in X$, $d(\tau, \tau) := 0$.

Step 2. Definition of the candidate dissimilariy

We start defining the candidate dissimilarity d. We will apply a agglomerative approach separately to each X_i.
We define the map $d : X \times X \to [0, +\infty)$ as follows:

1. For each patch $\tau \in X$, $d(\tau, \tau) := 0$.
2. For each pair of patches τ_1 and τ_2, $d(\tau_1, \tau_2)$ is defined as the smallest singular value computed by applying discrete approximate implicitization to a set of points uniformly sampled from each of the two patches.

Step 2. Definition of the candidate dissimilariy

We start defining the candidate dissimilarity d. We will apply a agglomerative approach separately to each X_i.
We define the map $d : X \times X \to [0, +\infty)$ as follows:

1. For each patch $\tau \in X$, $d(\tau, \tau) := 0$.
2. For each pair of patches τ_1 and τ_2, $d(\tau_1, \tau_2)$ is defined as the smallest singular value computed by applying discrete approximate implicitization to a set of points uniformly sampled from each of the two patches.
3. For each pair of clusters C_A and C_B, $d(C_A, C_B)$ is defined in terms of a complete-linkage approach:

$$D(C_A, C_B) = \max_{\tau_i \in C_A} \max_{\tau_j \in C_B} d(\tau_i, \tau_j).$$

Step 2. Definition of the candidate dissimilarity

It straightforward that:

- d depends on the points that are sampled on each patch.
- d is not a dissimilarity, because if $\tau_1 \neq \tau_2$ lies on the same primitive shape, then $d(\tau_1, \tau_2) = 0$.

Step 2. Definition of the candidate dissimilarity

It straightforward that:

- d depends on the points that are sampled on each patch.
- d is not a dissimilarity, because if $\tau_1 \neq \tau_2$ lies on the same primitive shape, then $d(\tau_1, \tau_2) = 0$.

but:

- it is not restrictive for our purposes to consider a map depending on some points that are a-priori sampled on each segment (patch).
- It is an advantage for our scope to have that $d(\tau_1, \tau_2) = 0$ if τ_1 and τ_2 lie on the same primitive shape.

Step 3. Agglomerative process

Starting from each patch in a separate cluster, at each step the two clusters with smallest value of d are merged.

The merging continues until the correct number of primitive shapes is detected.

Step 3. Agglomerative process

Starting from each patch in a separate cluster, at each step the two clusters with smallest value of d are merged.
The merging continues until the correct number of primitive shapes is detected.

How to estimate the number of clusters?

- At each iteration k we compute, for each cluster, the maximum value of d for pairs of patches. From a numerical viewpoint, it corresponds to an empirical estimation of the maximum error in the approximate implicitization of the patches.
- We consider then the maximum of the maxima and denote it as $e^{(k)}$.

Step 3. Agglomerative process

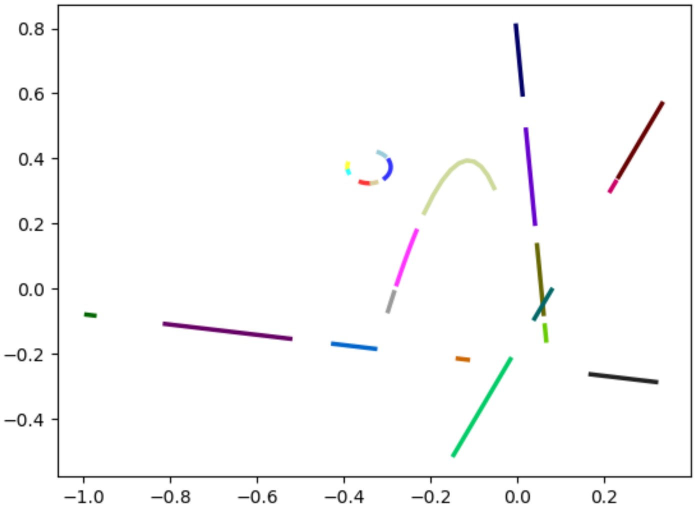
We observe that, if K is the effective number of clusters, then:

- In exact arithmetic, $e^{(k)} = 0$ for $k = 0, \ldots, N-K$ and $e^{(k)} > 0$ for $k = N-K+1, \ldots, N-1$. Therefore the number of iterations can be defined as:

$$\bar{k} := \arg\max_{k} \{k | e^{(k)} = 0\},$$

  that is the maximum index $\bar{k}$ such that $e^{(\bar{k})} = 0$.
- In floating-point arithmetic, we can either define a new stopping tolerance η or define an alternative criterion. We propose to estimate the number of clusters $\bar{k}$ using the following approach:
  - Set $e^{(-1)} := e^{(0)}$.
  - Compute through backward differences the finite sequence $\delta^{(k)} := e^{(k)} - e^{(k-1)}$.
  - Compute the number of clusters $\bar{x} = \arg\max_k \delta^{(k)}$.

Example 1: random segments of degree 1 and 2



Example 2: two-dimensional gear

Let X be the set of segments of the two-dimensional gear in Figure 5. The gear is built as follows:

- Three circles are approximated by means of Bézier curves.
- Line segments are used to connect inner and outer arcs to form the teeth.

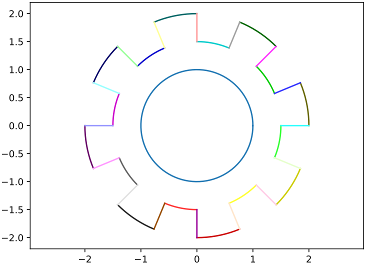


Figure 5: Initial gear.

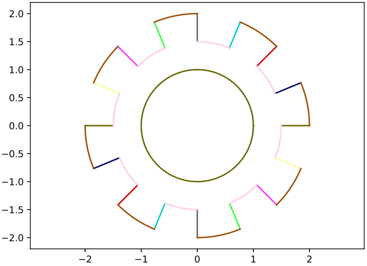


Figure 6: Classified gear.

Takk for meg!