Inner and outer approximation of the workspace of a robotic arm

Maël Godard[1] Luc Jaulin[1] Damien Massé[2]

[1]Lab-STICC, ROBEX Team, ENSTA

[2]Lab-STICC, ROBEX Team, UBO

June 29, 2025

Avec le soutien de

ENSTA INSTITUT POLYTECHNIQUE DE PARIS

Robotic arm

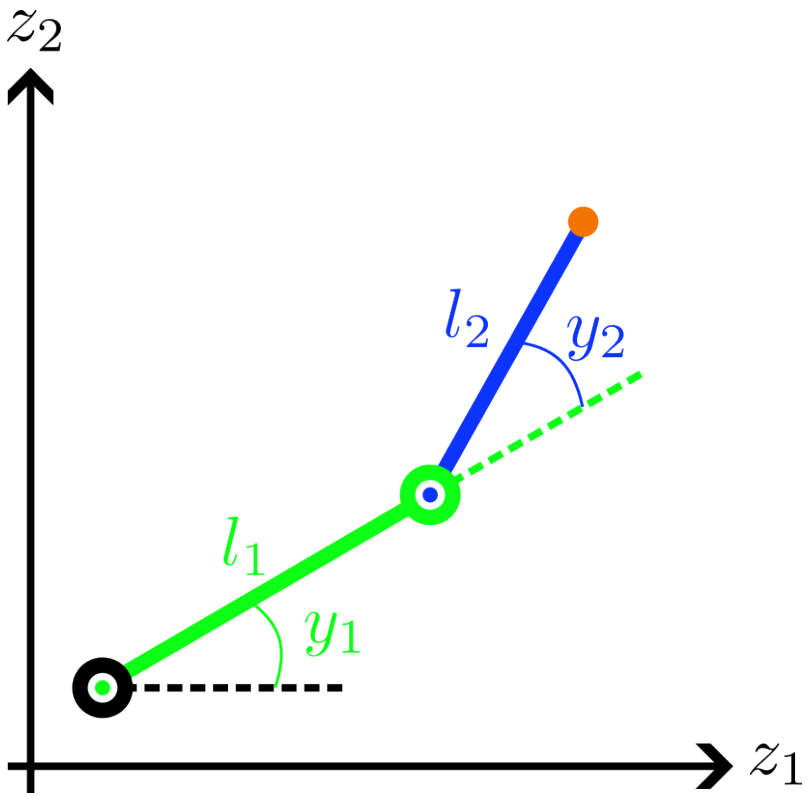


Figure: 2D Robotic arm

Position of the effector

Le us denote by $\mathbf{y} = (y_1, y_2)^T$ the state of the arm. The position of the effector is defined by :

$$\mathbf{f}(\mathbf{y}) = \begin{pmatrix} l_1 \cos(y_1) + l_2 \cos(y_1 + y_2) \\ l_1 \sin(y_1) + l_2 \sin(y_1 + y_2) \end{pmatrix}$$

For this study $\mathbf{y} \in \mathbb{Y}_0 = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]^2$ and $\partial\mathbb{Y}_0$ is the boundary of $\mathbb{Y}_0$.

Wrapping effect

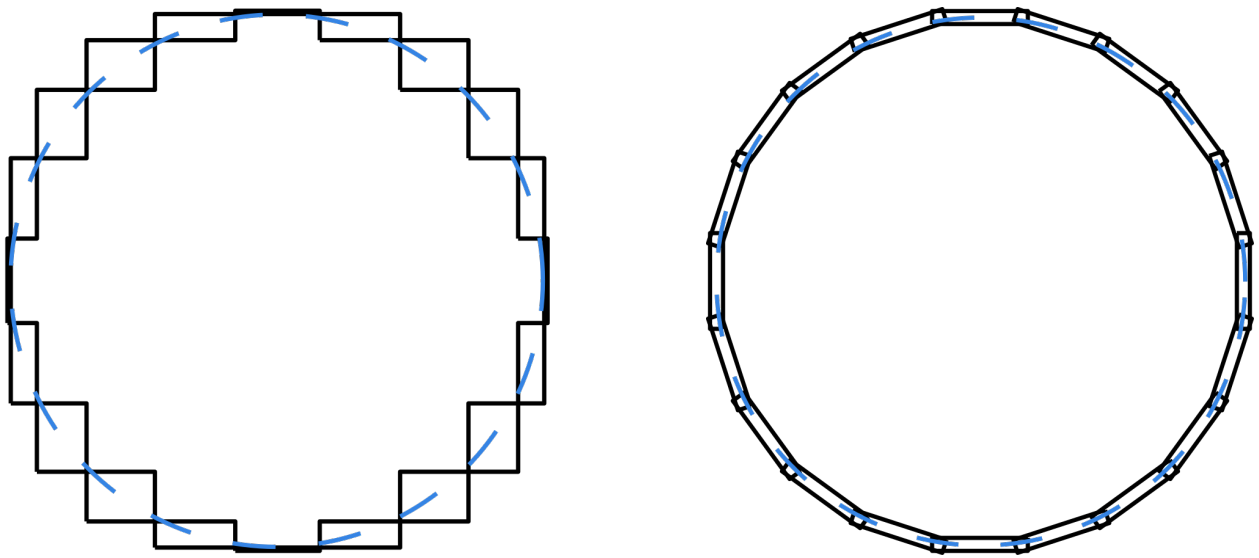


Figure: Enclosing of the unit circle with boxes and parallelepipeds

Principle

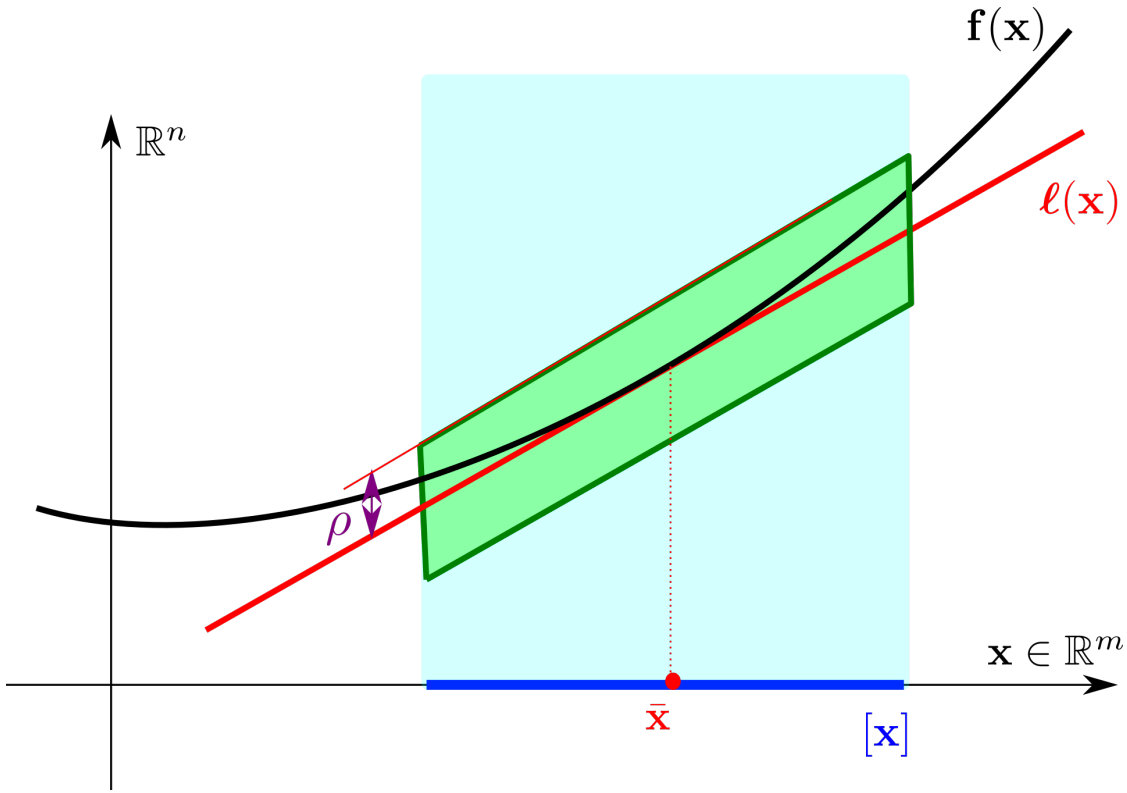


Figure: Parallelepipedic inclusion

Parallelepipedic inclusion function

Given $\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^n$. A parallelepipedic inclusion function is obtained as follows:

$$\langle \mathbf{f} \rangle([\mathbf{x}]) \quad = \quad \bar{\mathbf{y}} + \mathbf{A} \cdot [-1, 1]^n$$

with

$$\begin{array}{rcc} \bar{\mathbf{y}} & = & \mathbf{f}(\bar{\mathbf{x}}) \\ \mathbf{A}_0 & = & \frac{d\mathbf{f}}{d\mathbf{x}}(\bar{\mathbf{x}}) \cdot rad([\mathbf{x}]) \\ \rho & = & \text{ub}\left(\left\| \left(\left[\frac{d\mathbf{f}}{d\mathbf{x}} \right] ([\mathbf{x}]) - \frac{d\mathbf{f}}{d\mathbf{x}}(\bar{\mathbf{x}}) \right) \cdot ([\mathbf{x}] - \bar{\mathbf{x}}) \right\| \right) \\ \mathbf{A} & = & \text{Inflate}(\mathbf{A}_0, \rho) \end{array}$$

Note : This inclusion function converges with an order 1 (order 0 for natural extension and centered inclusion functions)

ENSTA INSTITUT POLYTECHNIQUE DE PARIS

Atlas of the initial set [1]

Let us define $s_1 = e^{\mathbf{k}\frac{\pi}{2}}$ the rotation of $\frac{\pi}{2}$ with respect to $\mathbf{k}$. The parallelepipedic inclusion of the initial box can be obtained by the symmetries :

$$\Sigma = \{1, s_1, s_1^2, s_1^{-1}\}$$

By using the function $\psi_0 : [-1, 1] \rightarrow \mathbb{Y}_0$

$$\psi_0(x) = \begin{pmatrix} -\frac{\pi}{2} \\ x \cdot \frac{\pi}{2} \end{pmatrix}$$

The boundary of the initial set then corresponds to:

$$\partial\mathbb{Y}_0 = \bigcup_{\sigma \in \Sigma} \sigma \circ \psi_0([-1, 1])$$

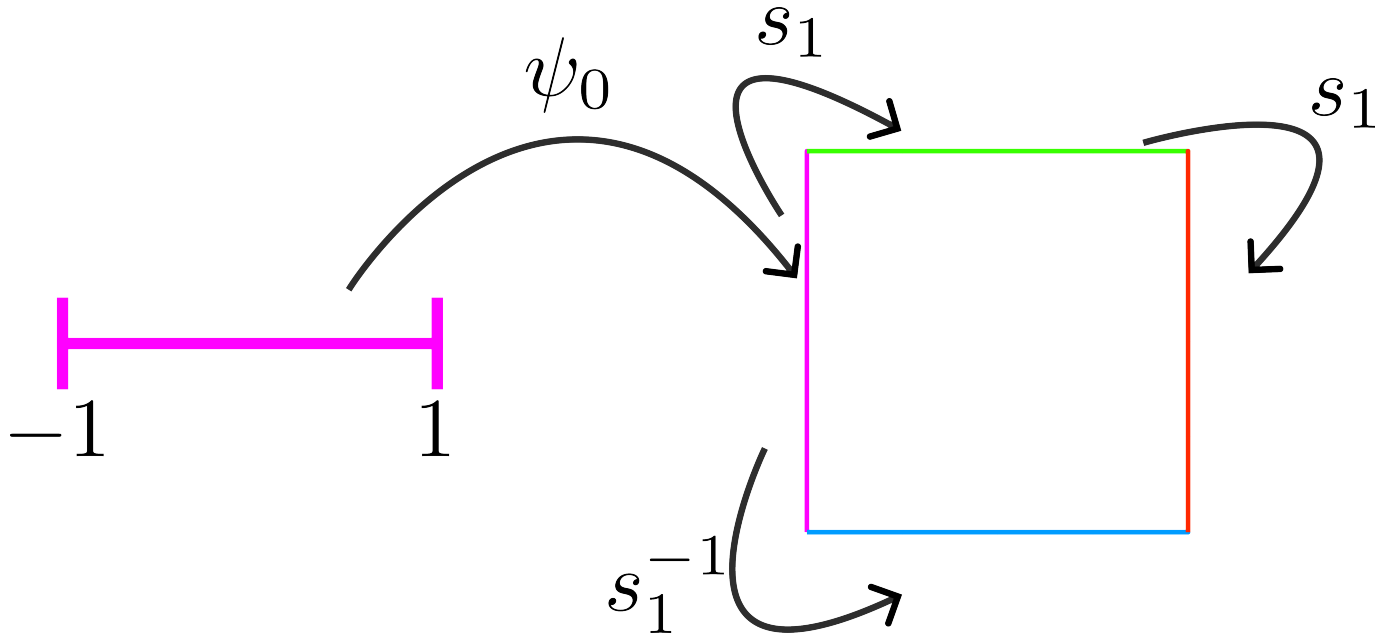


Figure: Construction of the initial box

Image of the initial set

Consider a function $\mathbf{f} : \mathbb{R}^2 \to \mathbb{R}^2$. We can then write the image of the boundary of the initial set $\mathbb{Y}_0$ by $\mathbf{f}$ as :

$$\mathbf{f}(\partial\mathbb{Y}_0) = \bigcup_i \mathbf{g}_i([-1, 1])$$

With :

$$\mathbf{g}_i = \mathbf{f} \circ \sigma_i \circ \psi_0$$

Image of the atlas

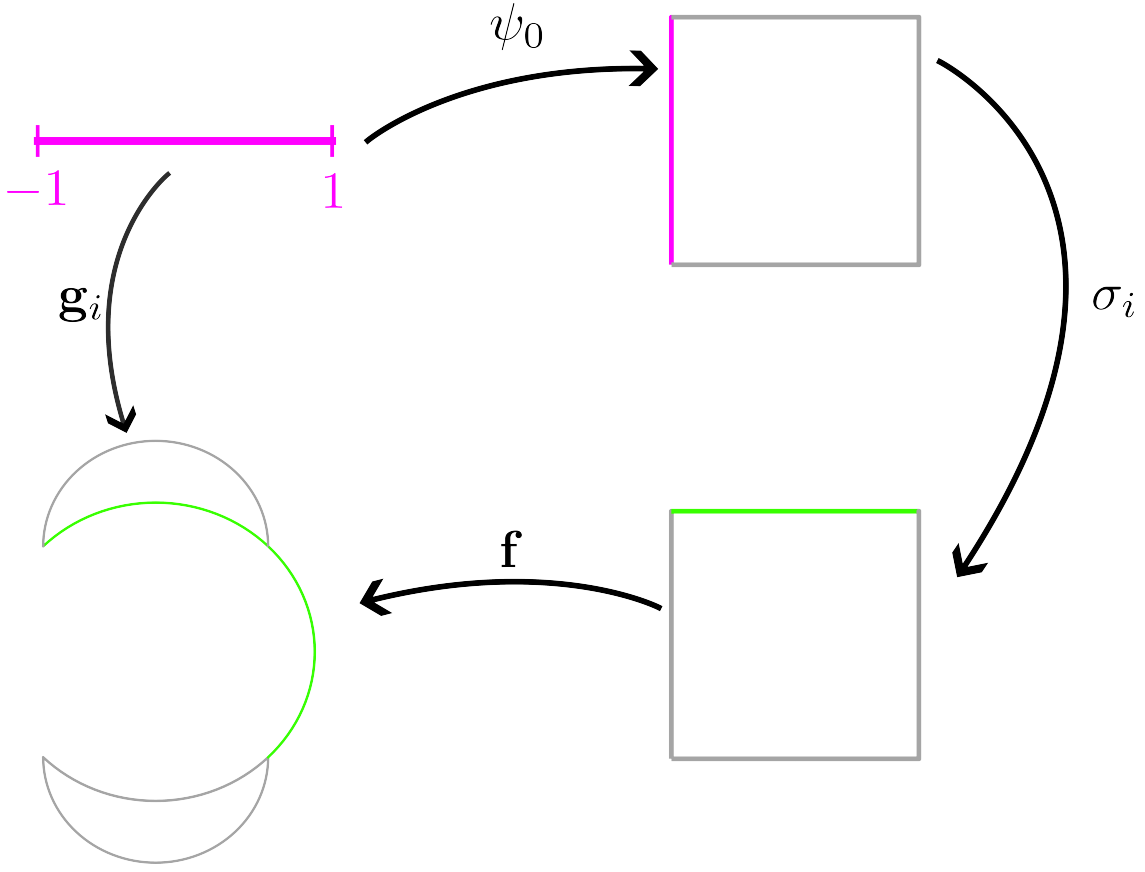


Figure: Image of $\mathbb{Y}_0$

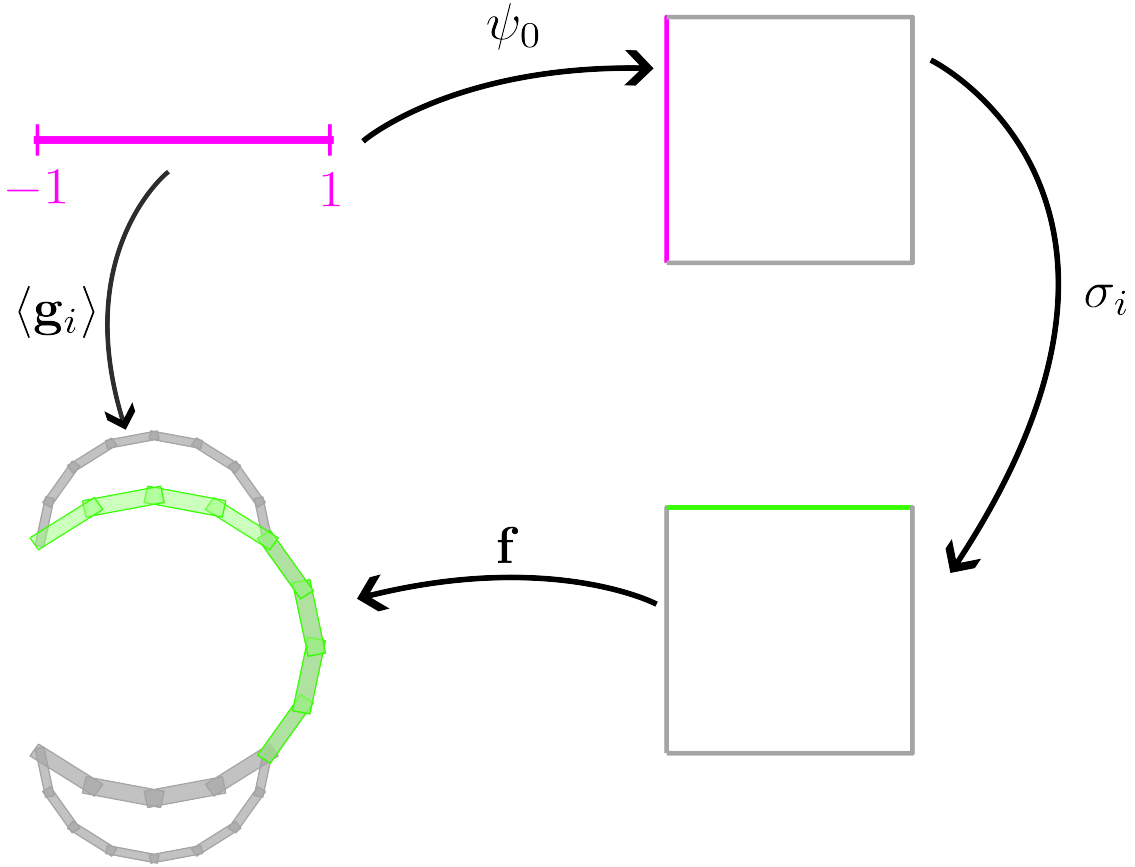


Figure: Image of $\mathbb{Y}_0$ with parallelepipeds [2]

Singularity

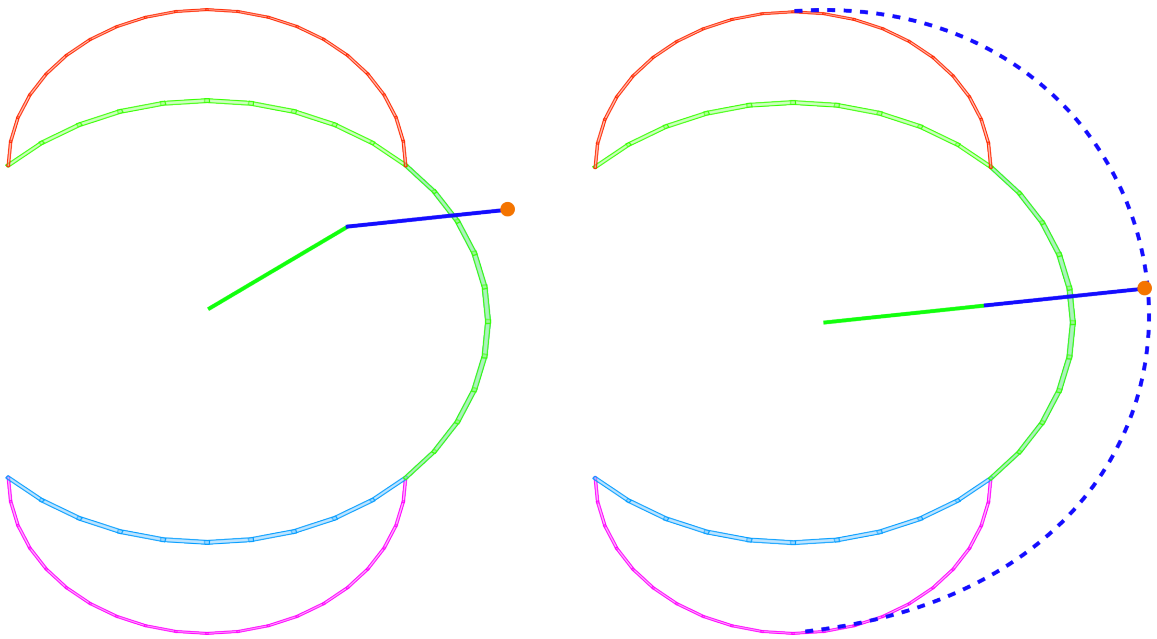


Figure: Singularity in the 2D arm

Singularity detection

Consider a function $\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^n$ and its Jacobian $\mathbf{J_f} : \mathbb{R}^m \to \mathbb{R}^{n \times m}$. The function $\mathbf{f}$ has a singularity at $\mathbf{y} \in \mathbb{R}^m$ if

$$\det\left(\mathbf{J_f}\left(\mathbf{y}\right)\right) = 0$$

For the robotic arm, we find that

$$\forall y_1 \in \left[\frac{-\pi}{2}, \frac{\pi}{2}\right], \det\left(\mathbf{J_f}\begin{pmatrix} y_1 \\ 0 \end{pmatrix}\right) = 0$$

Singularity management

To handle the singularity, we add an transformation function $\psi_s : [-1, 1] \rightarrow \mathbb{Y}_0$

$$\psi_s(x) = \begin{pmatrix} -x \cdot \frac{\pi}{2} \\ 0 \end{pmatrix}$$

We then define

$$\mathbf{g}_s = \mathbf{f} \circ \psi_s$$

Which gives us a total of 5 functions to study : $\mathbf{g}_0, \mathbf{g}_1, \mathbf{g}_2, \mathbf{g}_3, \mathbf{g}_s$

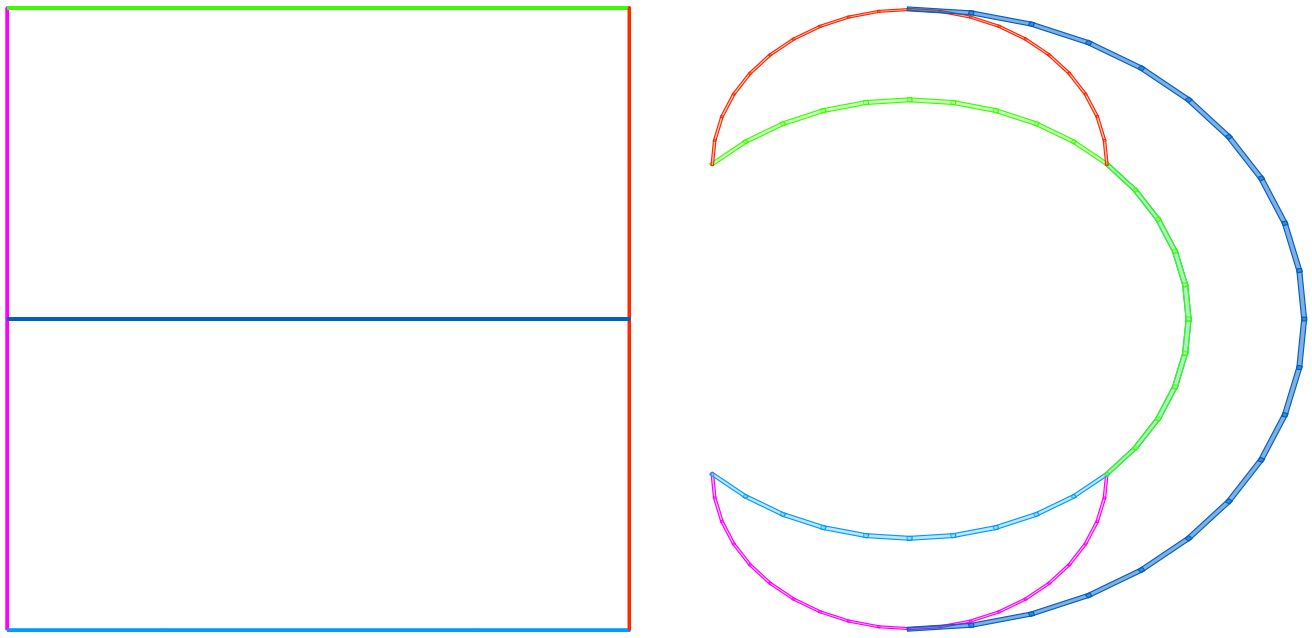


Figure: Atlas (left) and Workspace (right) with singularity

Polygon view [3]

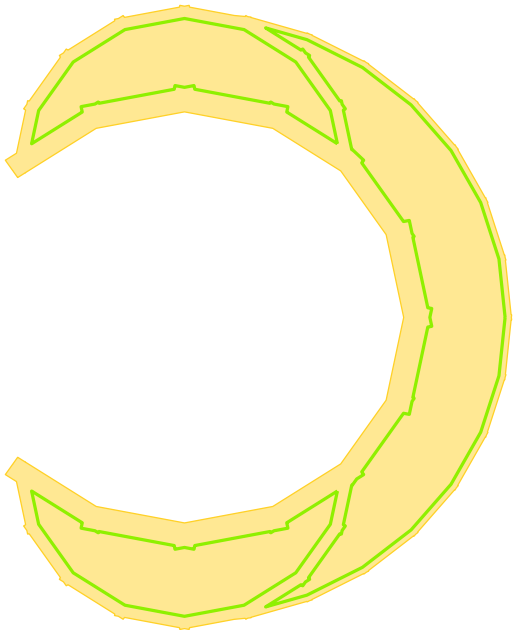


Figure: Polygon (yellow) with holes (green)

Convex partitions

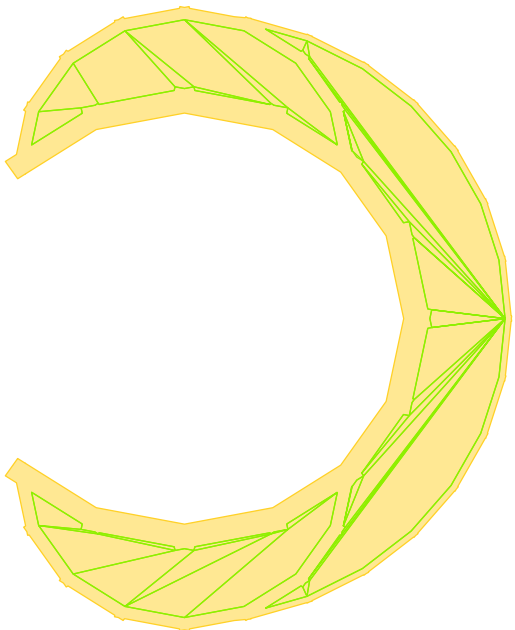


Figure: Convex decomposition of the holes

Delaunay triangulation

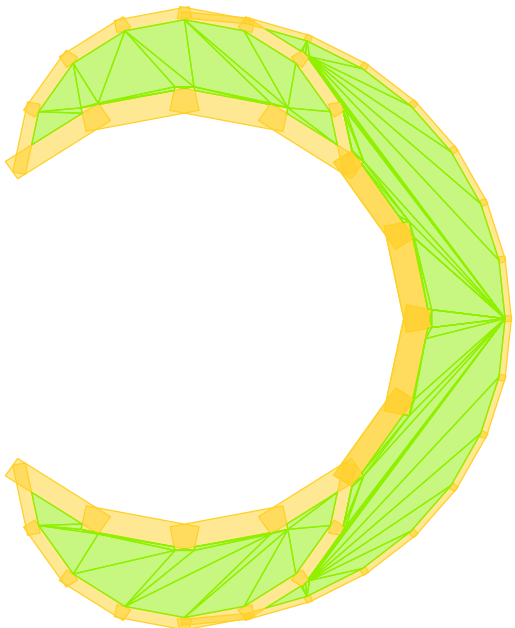


Figure: Delaunay triangulation

Outside triangulation

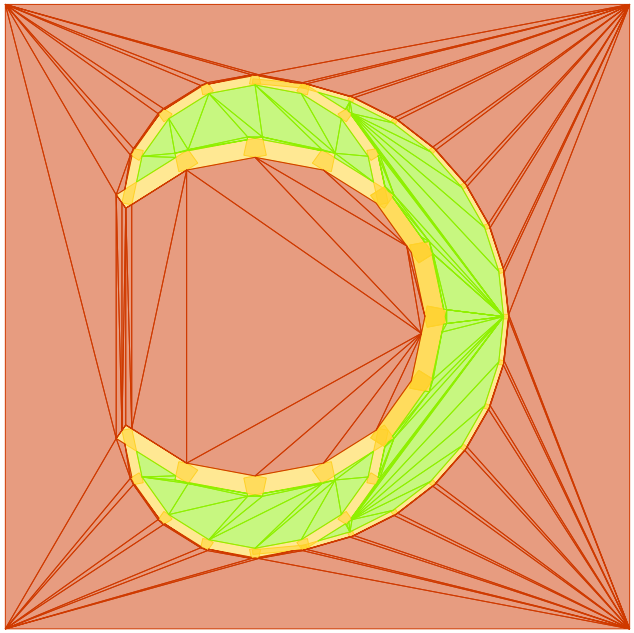


Figure: Outside triangulation

Suppressing fake boundaries

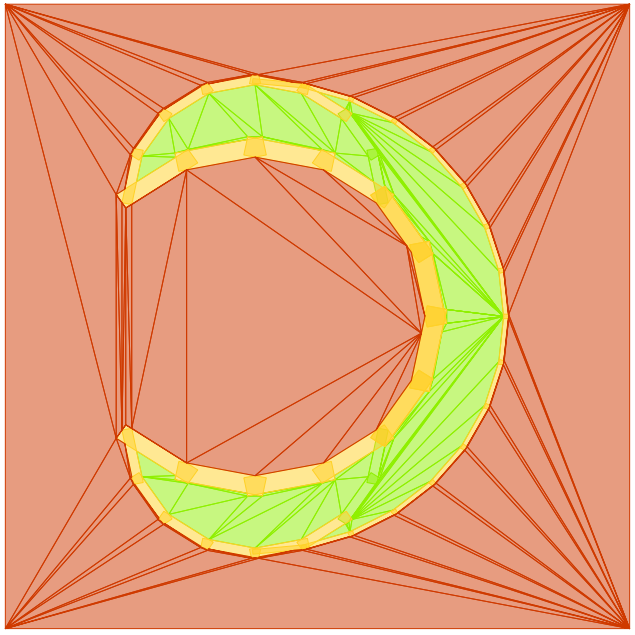


Figure: Inner (green) and outer (green + yellow) approximations

Additional use case

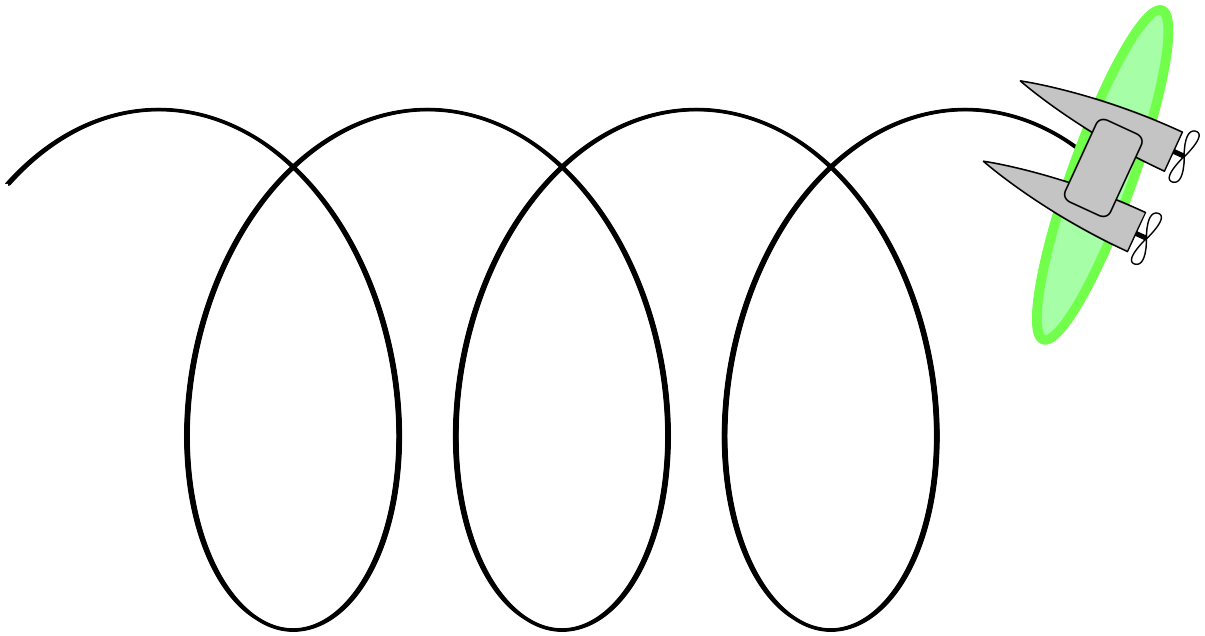


Figure: Robot sweeping an area

Modelization

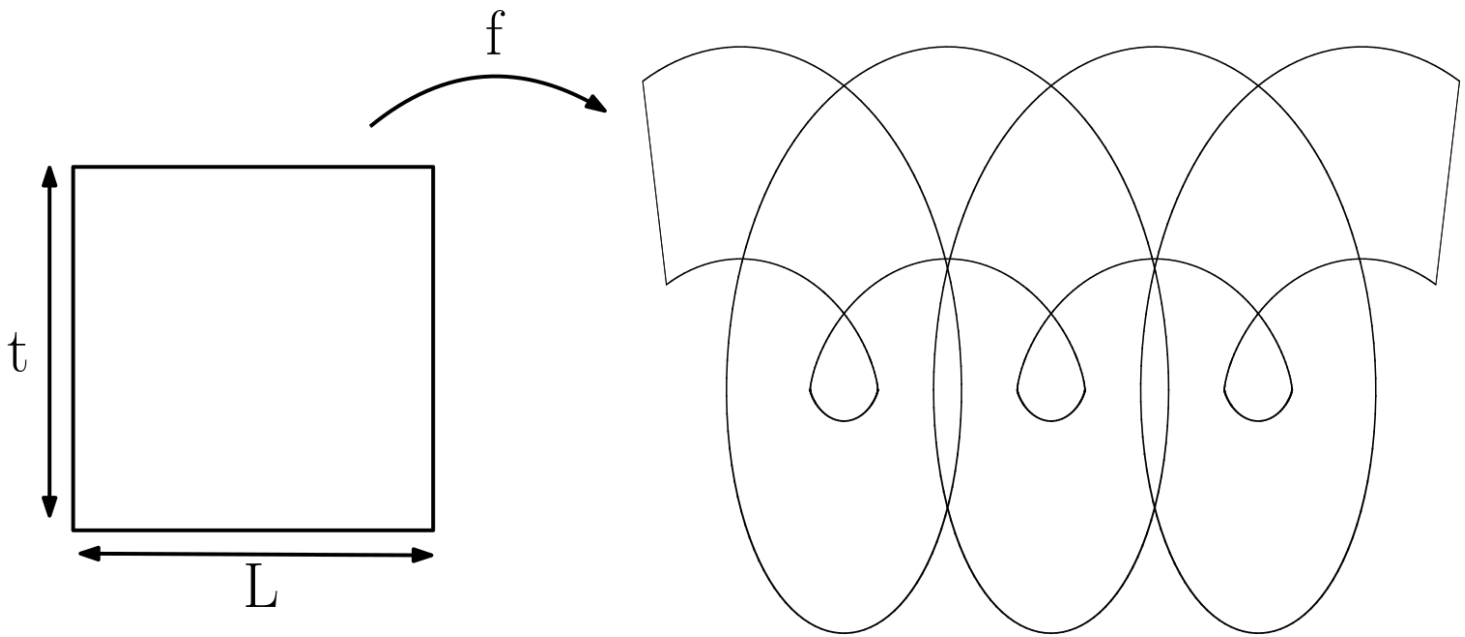


Figure: Modelization

Sweeped area with fakes

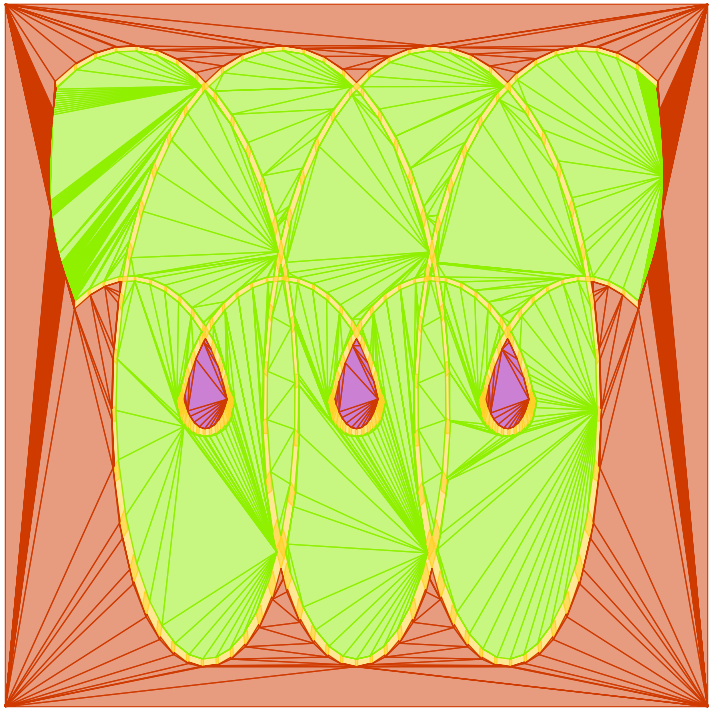


Figure: Sweeped area

Inner and outer approximation

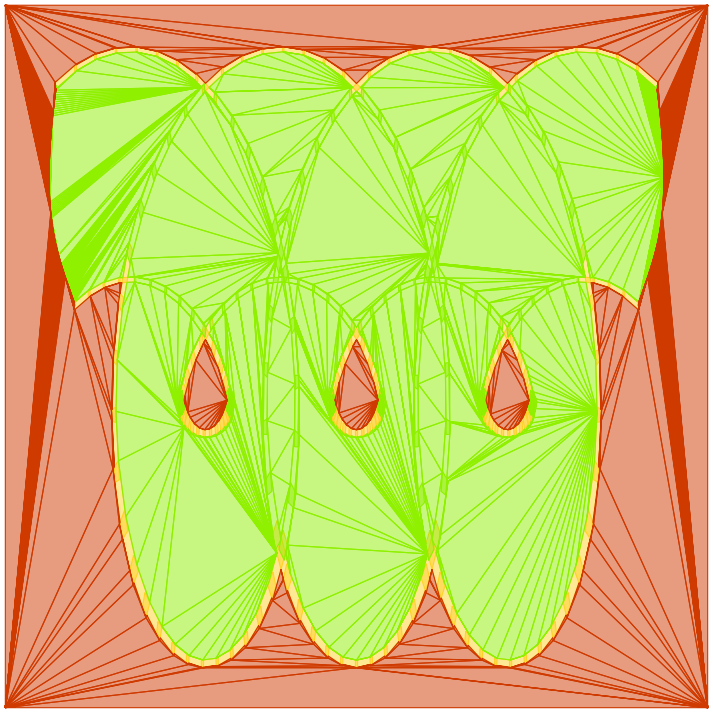


Figure: Fake boundaries removed

Conclusion

- Inner and outer approximation of the workspace
- Method based on parallelepipeds and Delaunay triangulation
- To be tested in higher dimensions

Bibliography

[1] Ignazi A., Manifold decomposition for interval analysis, SWIM 2023

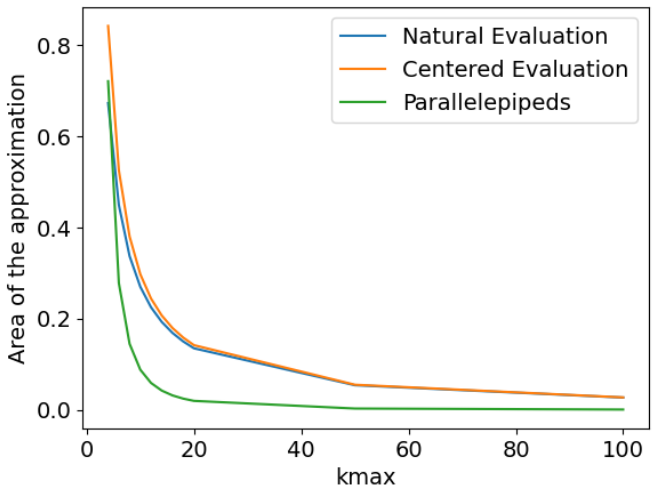
[2] Rohou S., Desrochers B., Le Bars F., The Codac Library: A Catalog of Domains and Contractors, Acta Cybernetica - Special Issue of SWIM 2022, 2024.

arle

[3] Susan Hert and Michael Seel. dD Convex Hulls and Delaunay Triangulations. In CGAL User and Reference Manual. CGAL Editorial Board, 6.0.1 edition, 2024.

Thank you for listening

Convergence



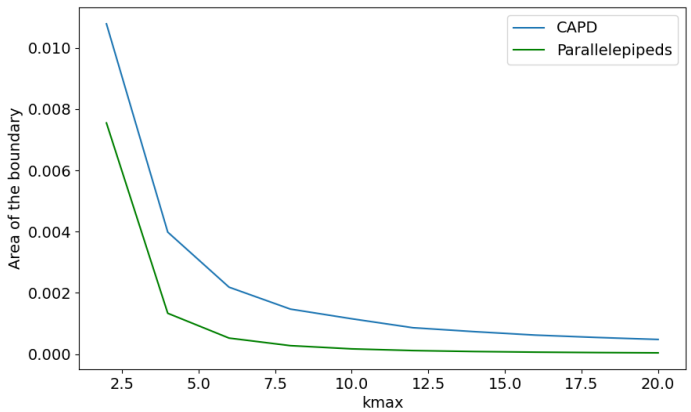


Figure: Convergences for the Henon map and the pendulum

Convergence speed

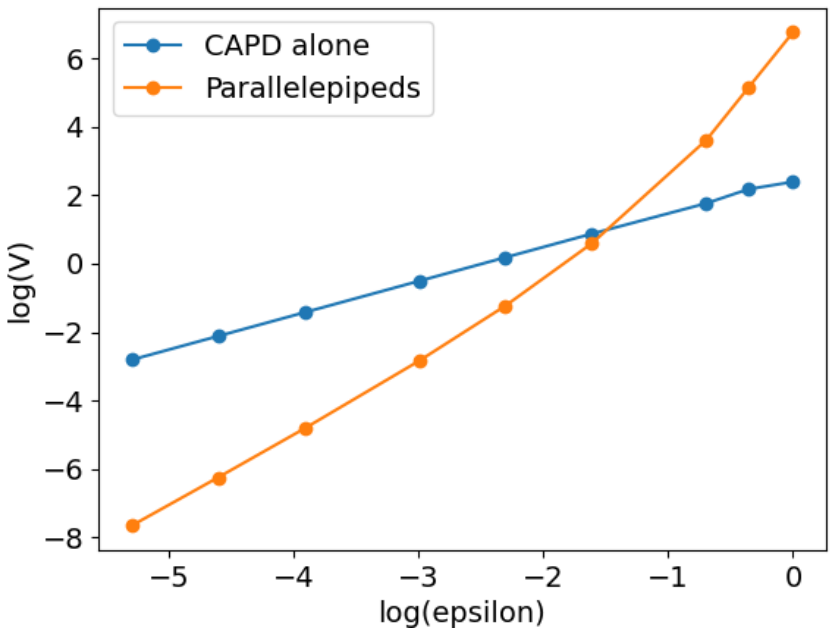


Figure: Convergence speed for the Lorenz system

Computation times

Dimension	Example	Computation time
2D	Henon map	0.01 s
	Pendulum (25 steps)	1.74 s
	Pendulum (last step)	0.21 s
	Robotic arm	0.01s
3D	Lorenz (0.05 s)	2.72 s
	Lorenz (0.1 s)	3.92 s
	Battery model (15 steps)	10 min
	Battery model (last step)	78 s
	Robotic arm	1.90 s

QR decomposition

Figure: Parallelepiped going “to infinity”

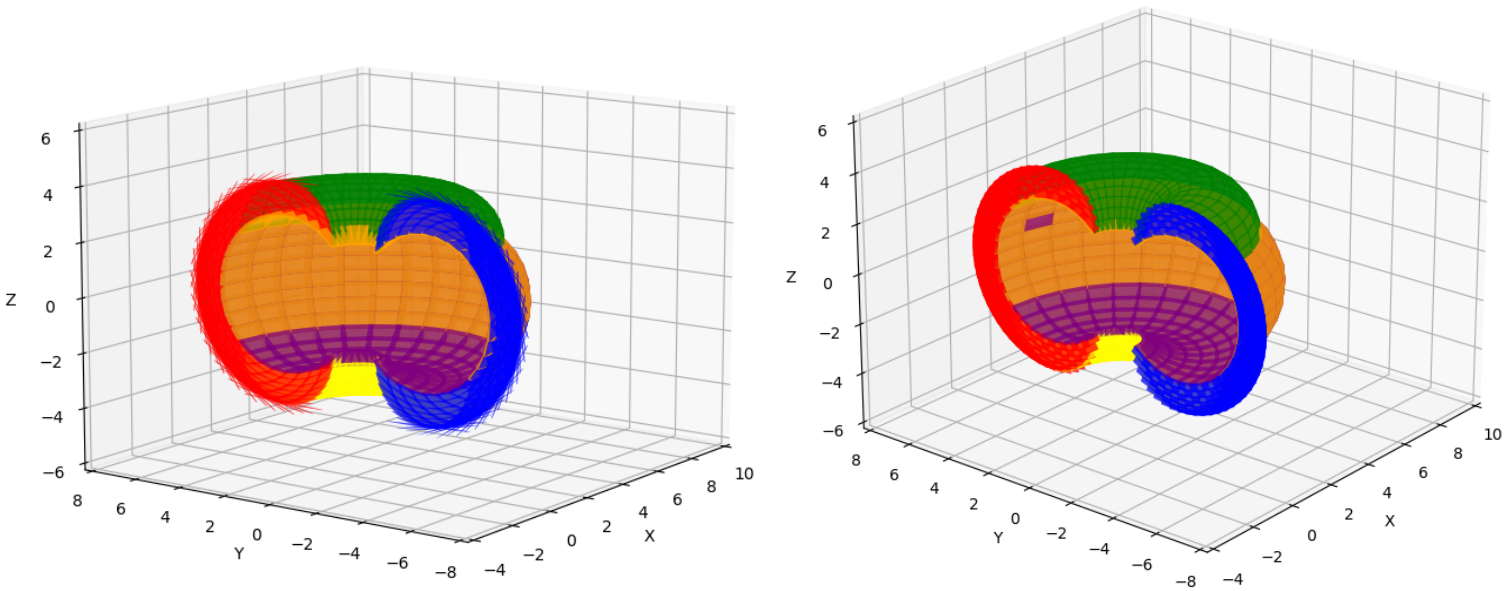


Figure: QR decomposition