Adaptative parallelepipedic approximation of the image of a set by a nonlinear function

Maël Godard, Luc Jaulin, Damien Massé

September 26, 2025

Introduction

We focus on the **direct problem**.

- Workspace of a robotic arm
- Observation (distance to landmarks)
- Reachability Analysis (not covered here)

Computing the exact image is not possible, we need to **enclose it in a set.**

Computation of the image of a set

We focus on the **direct problem**.

- Workspace of a robotic arm
- Observation (distance to landmarks)
- Reachability Analysis (not covered here)

Computing the exact image is not possible, we need to **enclose it in a set.**

Computation of the image of a set

We focus on the **direct problem**.

- Workspace of a robotic arm
- Observation (distance to landmarks)
- Reachability Analysis (not covered here)

Computing the exact image is not possible, we need to **enclose it in a set.**

Computation of the image of a set

We focus on the **direct problem**.

- Workspace of a robotic arm
- Observation (distance to landmarks)
- Reachability Analysis (not covered here)

Computing the exact image is not possible, we need to **enclose it in a set.**

Computation of the image of a set

We focus on the **direct problem**.

- Workspace of a robotic arm
- Observation (distance to landmarks)
- Reachability Analysis (not covered here)

Computing the exact image is not possible, we need to **enclose it in a set.**

- Boxes
- Zonotopes
- Ellipsoids
- $\cdots$

We choose **Parallelepipeds** as an intermediate between boxes and zonotopes

- Boxes
- Zonotopes
- Ellipsoids
- $\cdots$

We choose **Parallelepipeds** as an intermediate between boxes and zonotopes

Sets

- Boxes
- Zonotopes
- Ellipsoids
- ...

We choose **Parallelepipeds** as an intermediate between boxes and zonotopes

- Boxes
- Zonotopes
- Ellipsoids
- $\cdots$

We choose **Parallelepipeds** as an intermediate between boxes and zonotopes

Definition (Parallelepiped)

A *parallelepiped* is a subset of $\mathbb{R}^n$ of the form

$$\langle \mathbf{y} \rangle = \bar{\mathbf{y}} + \mathbf{A} \cdot [-1, 1]^m = \{\bar{\mathbf{y}} + \mathbf{A} \cdot \mathbf{x} \,|\, \mathbf{x} \in [-1, 1]^m\}$$

With $m \leq n$

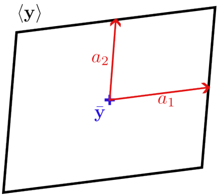


Figure 1: 2D parallelepiped

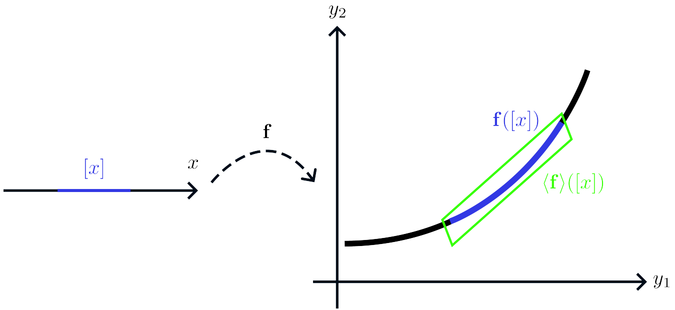
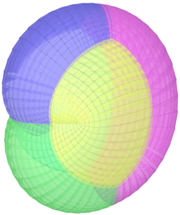


Figure 2: Parallelepiped inclusion function

PEIBOS stands for **P**arallelepipedic **E**nclosure of the **I**mage of the **BO**undary of a **S**et.



https://godardma.github.io/subpages/libs/parallelepiped.html

[1] Maël Godard, Luc Jaulin, Damien Massé, Inner and outer approximation of the image of a set by a nonlinear function, *International Journal of Approximate Reasoning*, Volume 187, 2025.

Need for adaptivity

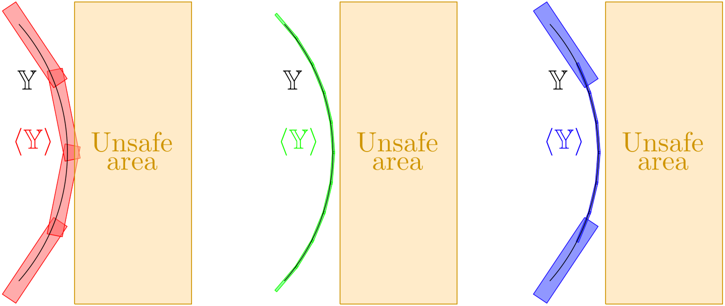


Figure 3: Bad, good and adaptative approximations

Adaptative parallelepipedic approximation

Constraint

Definition (Constraint)

A constraint c is a boolean function defined by :

$$\begin{array}{rccc} c: & \mathbb{R}^n & \rightarrow & \{0,1\} \\ & \mathbf{y} & \mapsto & \begin{cases} 1 & \text{if } \mathbf{y} \text{ satisfies the constraint} \\ 0 & \text{Otherwise} \end{cases} \end{array}$$

Definition

A constraint is verified on a set if it is verified on every point of it

$$\begin{array}{rccc} c: & \mathcal{P}(\mathbb{R}^n) & \rightarrow & \{0,1\} \\ & \mathbb{Y} & \mapsto & \begin{cases} 1 & \text{if } \forall \mathbf{y} \in \mathbb{Y},\ c(\mathbf{y}) = 1 \\ 0 & \text{Otherwise} \end{cases} \end{array}$$

Definition (Constraint)

A constraint c is a boolean function defined by :

$$\begin{aligned} c: \quad \mathbb{R}^n &\rightarrow \{0,1\} \\ \mathbf{y} &\mapsto \begin{cases} 1 & \text{if } \mathbf{y} \text{ satisfies the constraint} \\ 0 & \text{Otherwise} \end{cases} \end{aligned}$$

Definition

A constraint is verified on a set if it is verified on every point of it

$$\begin{aligned} c: \quad \mathcal{P}\left(\mathbb{R}^n\right) &\rightarrow \{0,1\} \\ \mathbb{Y} &\mapsto \begin{cases} 1 & \text{if } \forall \mathbf{y} \in \mathbb{Y},\ c(\mathbf{y}) = 1 \\ 0 & \text{Otherwise} \end{cases} \end{aligned}$$

Contractor

Definition

A contractor $\mathcal{C}_c$ associated to a constraint c is a function $\mathcal{C}_c : \mathbb{IR}^n \to \mathbb{IR}^n$ contracting a box with respect to the constraint c. It satisfies

- $\forall\, [\mathbf{x}] \in \mathbb{IR}^n,\ \mathcal{C}_c([\mathbf{x}]) \subseteq [\mathbf{x}]$ (Contractance)
- $\forall \mathbf{x} \in [\mathbf{x}],\ c(\mathbf{x}) \implies \mathbf{x} \in \mathcal{C}_c\left([\mathbf{x}]\right)$ (Consistency)

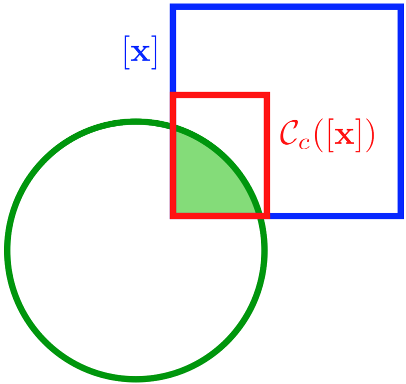


Figure 4: Contraction

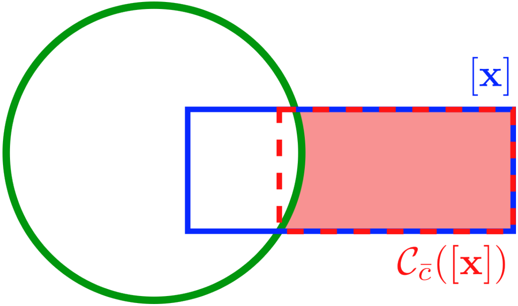


Figure 5: Contractor on the complementary

Inputs

- A function $\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^n$, $m < n$
- A box $[\mathbf{x}_0] \in \mathbb{IR}^m$
- A contractor $\mathcal{C}_{\bar{c}} : \mathbb{R}^n \to \mathbb{R}^n$ associated to the complementary of the constraint $c : \mathbb{R}^n \to \{0, 1\}$
- A resolution $\epsilon \in \mathbb{R}^+$ with a lower limit $\epsilon_{lim} \in \mathbb{R}^+$
- A list of Parallelepipeds $\mathcal{L}_P$.

Algorithm

Algorithm 1 Adaptative parallelepiped enclosure

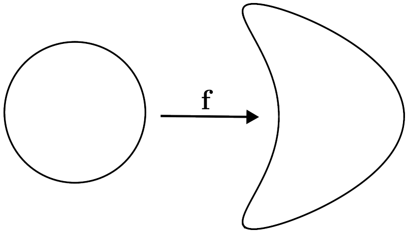
```
Output the list $\mathcal{L}_P$ completed in place
Notation Ad-PEIBOS_step($\mathbf{f}, [\mathbf{x}_0], \mathcal{C}_{\bar{c}}, \epsilon, \epsilon_{lim}, \mathcal{L}_p$)

if $\epsilon < \epsilon_{lim}$ then
    Raise an alarm
else
    Split $[\mathbf{x}_0]$ in a list of boxes $\mathcal{L}_\mathbf{x}$ of diameter $\epsilon$ or less
    for $[\mathbf{x}]$ in $\mathcal{L}_\mathbf{x}$
        Compute $\langle \mathbf{y} \rangle$ the parallelepiped enclosing $\mathbf{f}([\mathbf{x}])$
        if $\mathcal{C}_{\bar{c}}(\langle \mathbf{y} \rangle) = \emptyset$ // The constraint $c$ is satisfied on $\langle \mathbf{y} \rangle$
            Store $\langle \mathbf{y} \rangle$ in $\mathcal{L}_p$
        else
            Ad-PEIBOS_step($\mathbf{f}, [\mathbf{x}], \mathcal{C}_{\bar{c}}, \frac{\epsilon}{2}, \epsilon_{lim}, \mathcal{L}_p$)
```

Illustration

For graphical purposes, we consider the Henon map defined by :

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} x_2 + 1 - a x_1^2 \\ b x_1 \end{pmatrix}, a = 1.4, b = 0.3$$



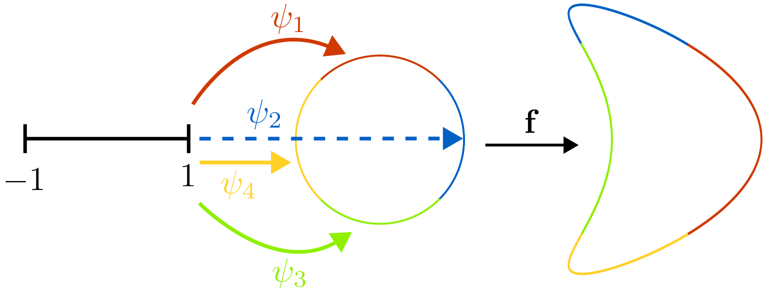


Figure 6: Atlas of the unit circle

[2] Arthur Ignazi, Remy Guyonneau, Sébastien Lagrange, Sébastien Lahaye, Box atlas: An interval version of atlas, *International Journal of Approximate Reasoning*, Volume 183, 2025.

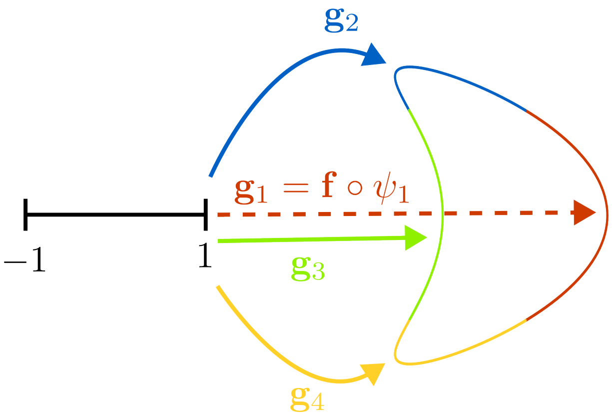
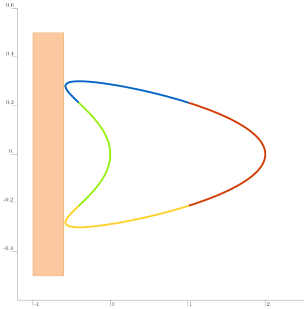


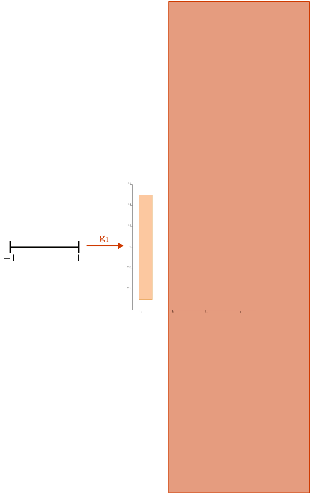
Figure 7: $\mathbf{g}_i$ functions

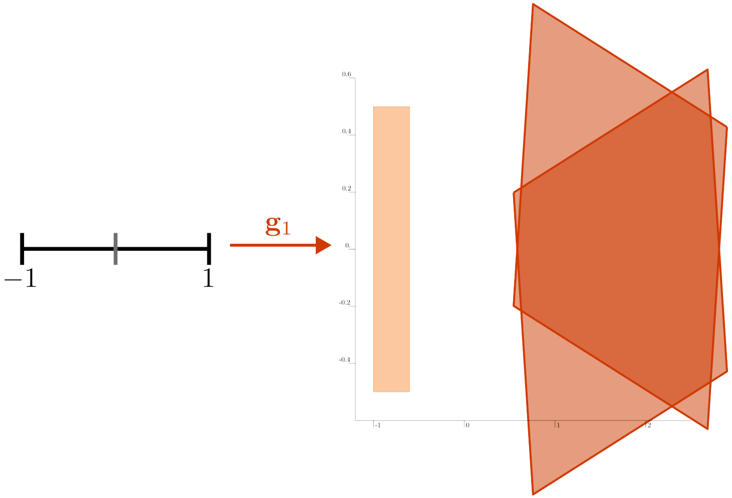
The constraint is

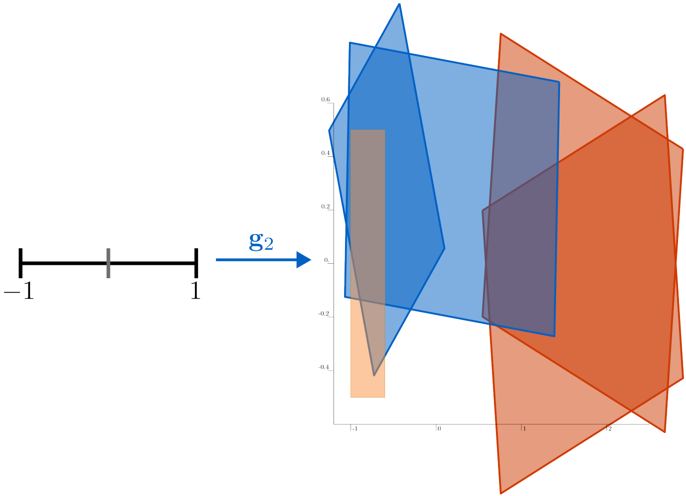
$$\begin{aligned} c: \quad \mathbb{R}^2 &\rightarrow \quad \{0,1\} \\ \mathbf{y} &\mapsto \quad y_2 < -0.6 \end{aligned}$$



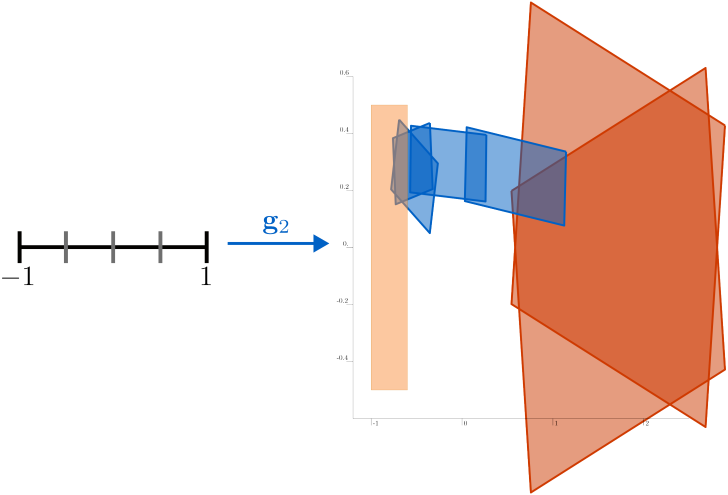
Step by step



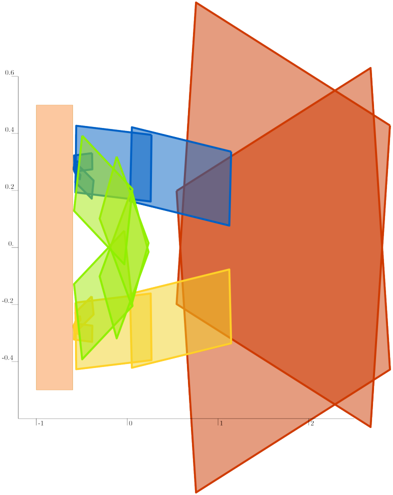

−1
1
g1
0.6
0.4
0.2
0.
-0.2
-0.4
-1
0
1
2



−1
1
g₂
0.6
0.4
0.2
0.
-0.2
-0.4
-1
0
1
2



-1
1
$\mathbf{g}_2$
0.6
0.4
0.2
0.
-0.2
-0.4
-1
0
1
2


0.6
0.4
0.2
0.
-0.2
-0.4
-1
0
1
2

We use the **small resolution** everywhere. Computation time goes from **3ms** to **6ms**.

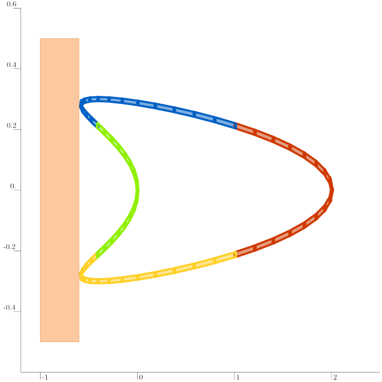


Figure 8: Small resolution enclosure of the image set

Additional example

The position of the effector is defined by

$$\mathbf{z} = \mathbf{f}\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} = \begin{pmatrix} \cos(\alpha_1)\cdot(l_1 + l_2\cos(\alpha_2) + l_3\cos(\alpha_2+\alpha_3)) \\ \sin(\alpha_1)\cdot(l_1 + l_2\cos(\alpha_2) + l_3\cos(\alpha_2+\alpha_3)) \\ l_2\sin(\alpha_2) + l_3\sin(\alpha_2+\alpha_3) \end{pmatrix}$$

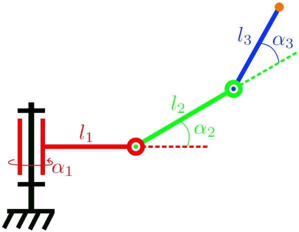


Figure 9: Robotic arm

Workspace of the arm

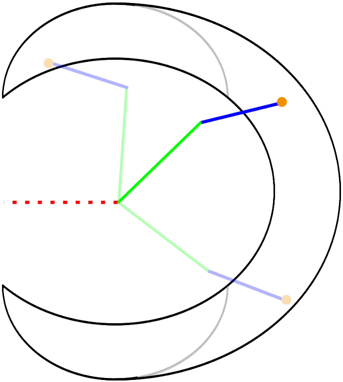


Figure 10: 2D workspace

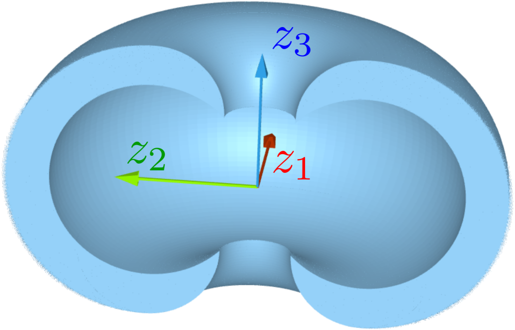


Figure 11: Revolution of the workspace

$$\begin{cases} c_1(\mathbf{z}) = \left(\sqrt{{z_2}^2 + z_3^2} < 1.2\right) & \text{(radius of the cylinder)} \\ c_2(\mathbf{z}) = (z_1 < 1.055) & \text{(depth of the cylinder)} \end{cases}$$

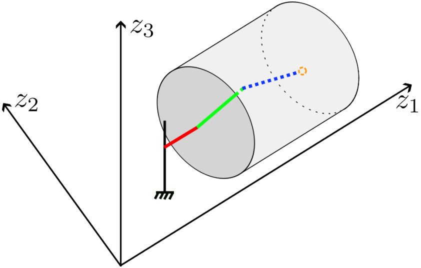


Figure 12: Arm in a cylinder

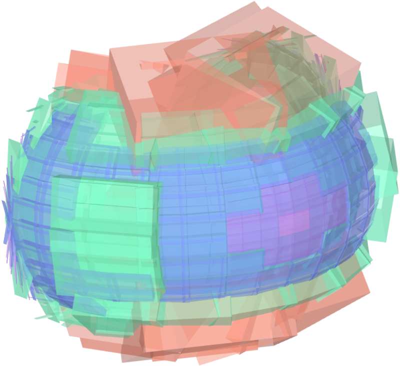


Figure 13: Adaptative enclosure

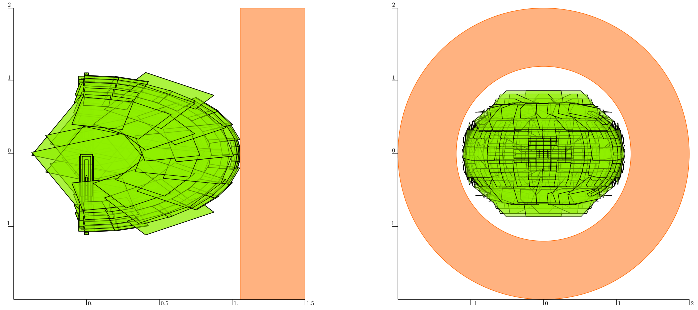


Figure 14: Projections in the (z_1, z_2) and (z_2, z_3) planes

Once again we use the **fine resolution** everywhere. We go up from **0.1s** (adaptative method) to **4s**.

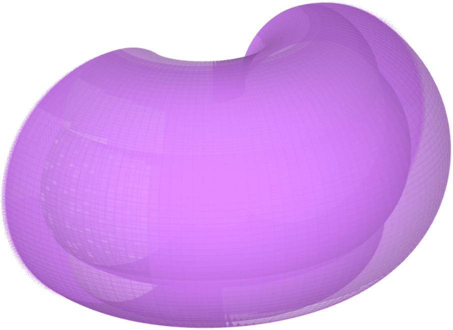


Figure 15: Fine enclosure

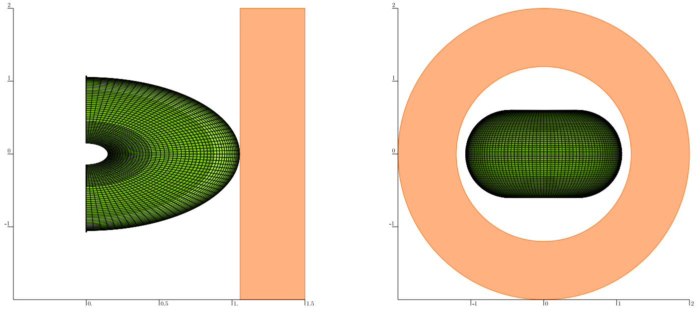


Figure 16: Projections in the (z_1, z_2) and (z_2, z_3) planes

Conclusion

Conclusion ...

- Adaptative method for parallelepiped approximation
- Constraint approach to refine the approximation where needed
- 2D and 3D examples, works in n-D
- Applicable to dynamical systems (ODE)

- Adaptative method for parallelepiped approximation
- Constraint approach to refine the approximation where needed
- 2D and 3D examples, works in n-D
- Applicable to dynamical systems (ODE)

Conclusion ...

- Adaptative method for parallelepiped approximation
- Constraint approach to refine the approximation where needed
- 2D and 3D examples, works in n-D
- Applicable to dynamical systems (ODE)

Conclusion ...

- Adaptative method for parallelepiped approximation
- Constraint approach to refine the approximation where needed
- 2D and 3D examples, works in n-D
- Applicable to dynamical systems (ODE)

- Application to reachability
- PEIBOS joining Codac
- $\cdots$ Ad-PEIBOS joining too?

- Application to reachability
- PEIBOS joining Codac
- $\cdots$ Ad-PEIBOS joining too?

- Application to reachability
- PEIBOS joining Codac
- $\cdots$ Ad-PEIBOS joining too?

Thank you for your attention