



**HAL**  
open science

## Plotting Options with SpaceEx

Nikolaos Kekatos

► **To cite this version:**

| Nikolaos Kekatos. Plotting Options with SpaceEx. 2021. hal-03091955

**HAL Id: hal-03091955**

**<https://hal.science/hal-03091955>**

Preprint submitted on 1 Jan 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Plotting options with SpaceEx

**Nikolaos Kekatos**

*Verimag Laboratory, University of Grenoble Alpes*  
nikolaos.kekatos@univ-grenoble-alpes.fr

This document provides several options to visualize and plot the SpaceEx<sup>1</sup> results. SpaceEx results correspond to reachable sets and the visualization is possible for 2D and 3D sets. Note that SpaceEx generates .gen files for 2D output, i.e., lists of vertices of polyhedra separated by a blank line. For 3D output, SpaceEx generates .jvx files.

Herein, we focus on two examples, a pendulum model and a timed bouncing ball. The model files are enclosed in the accompanied zip file.

## 1 Web Interface

In the web interface, it is possible to visualize both two-dimensional and three-dimensional graphs. Also, the SpaceEx server can generate GIF and pdf images of 2D graphs.

## 2 Plotutils/Graph

For command line users, the easiest option is to use the `graph` utility from the `Plotutils` package<sup>2</sup>. This package is available for all major Linux distributions and for MacOS. If not installed, follow online instructions<sup>3</sup>. MacOS users could write in the command line `$ brew install wget` (to install Homebrew) and `$ brew install plotutils` (to install Plotutils).

To generate a PNG bitmap, write in the terminal `$ graph -T png -C -B -q0.5 myfile.gen > myfile.png`. To generate a PDF file, write `$ graph -T ai -C -B -q0.5 myfile.gen > myfile.pdf`

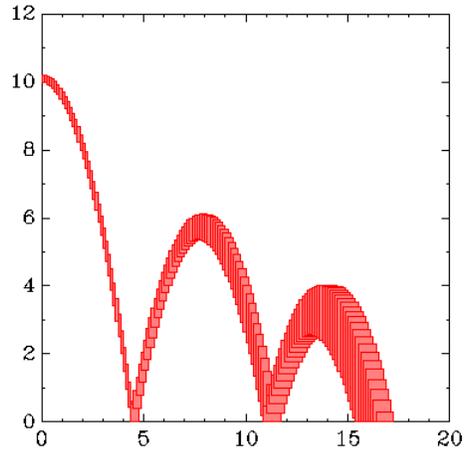
---

<sup>1</sup><http://spaceex.imag.fr/>

<sup>2</sup><https://www.gnu.org/software/plotutils>

<sup>3</sup><https://www.gnu.org/software/plotutils/manual/en/plotutils.html>

For the timed bouncing ball, we get the following plot.

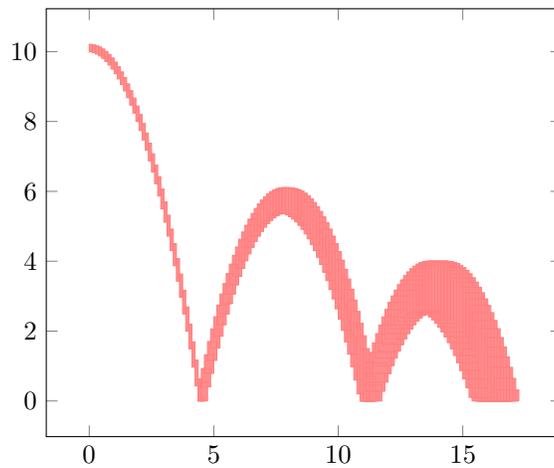


### 3 LaTeX

The second option is to use LaTeX and employ the TikZ package. In particular,

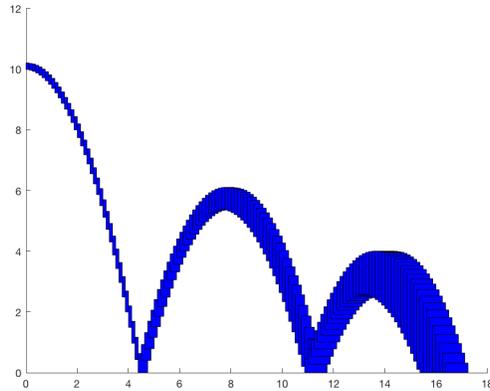
```
\usepackage{tikz}
\usepackage{pgfplots}
...
\begin{tikzpicture}
\begin{axis}
\addplot [color=red,fill=red!90!white,opacity=0.5] table {myfile.gen};
\end{axis}
\end{tikzpicture}
```

Replacing `myfile.gen` with `bball_timed.gen`, we get the TikZ image below.

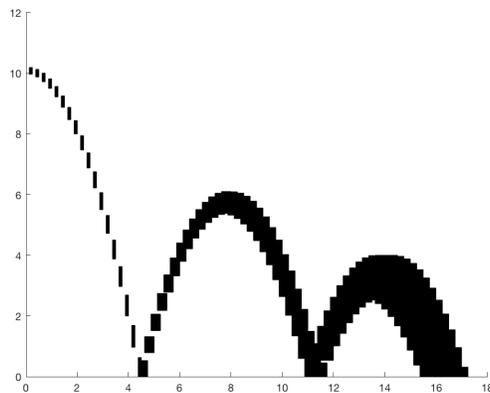


## 4 MATLAB

Another option is to run MATLAB scripts for 2D or 3D plotting. For the 2D graphs, it is possible to use the script `plot_2d_vertices.m`<sup>4</sup>. For the timed bouncing ball, we get the following plot.



A revised script (containing two main functions) is included in the zip file. There are extra functionalities, such as save option, animated display of the reachable sets, verbosity levels, ranges, polytope selections, reduced computation time. For the timed bouncing ball, assume that we do not want the entire flowpipe (all the polytopes). We can only plot half of it (evenly distributed) by running `template_plotting_bball`.



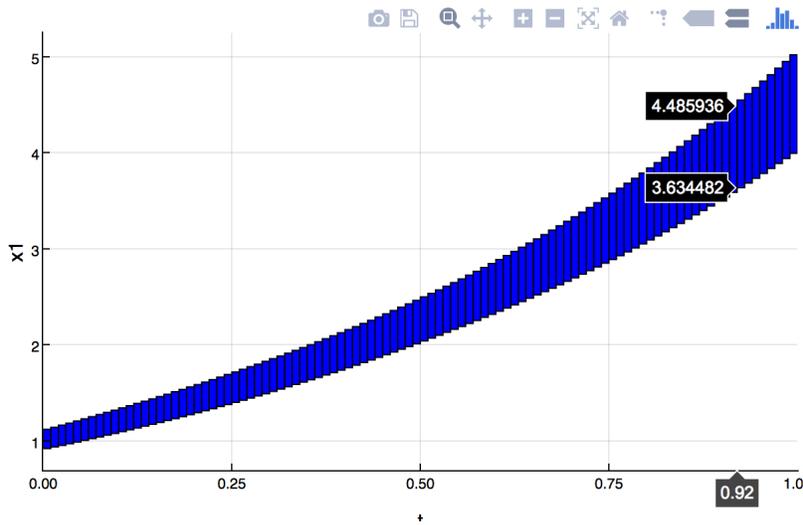
---

<sup>4</sup>It is available at [http://spacex.imag.fr/sites/default/files/downloads/plot\\_2d\\_vertices.m](http://spacex.imag.fr/sites/default/files/downloads/plot_2d_vertices.m)

## 5 Julia

LazySets.jl<sup>5</sup> is part of JuliaReach<sup>6</sup>, a project developing tools for reachability computations of dynamical systems in Julia<sup>7</sup>. It is developed by Marcelo Forets, Christian Schilling, and Frédéric Viry.

This is a feature-rich alternative and there are many plotting backends available. We refer to the documentation of `LazySets.jl` for further details. The following plot (for a randomly generated model) was obtained with the `Plotly` backend.



## 6 Java

Another option is to use Java. The project 3PLIB<sup>8</sup> is a Java Library specialized to planar projections of convex polyhedra. It is created by Frédéric Viry and provides a backend for visualization of high-dimensional reach set approximations computed with SpaceEx.

<sup>5</sup><https://github.com/juliareach/LazySets.jl>

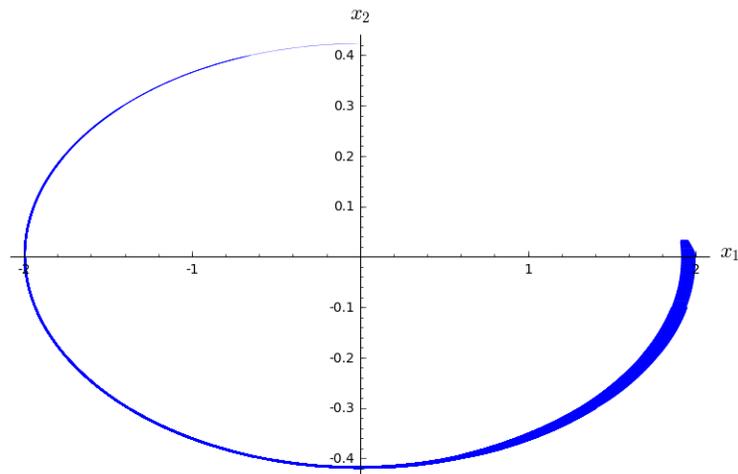
<sup>6</sup><https://github.com/juliareach>

<sup>7</sup><https://julialang.org/>

<sup>8</sup><https://3plib.wordpress.com/>

## 7 Python

The next option is to use a Python script that sequentially reads the polygons as polyhedra objects and requires SageMath<sup>9</sup>. For the pendulum example, we use the simple Python script `plot_2d_vertices.ipynb`. It is written by Marcelo Forets and it produces the results very fast. It is suitable for big models with many generated polytopes. For the pendulum example, we get the following plot.



---

<sup>9</sup><http://www.sagemath.org/>