SOFTWARE FOR VISUALIZING AND ANALYZING DYNAMICAL SYSTEMS IN ℝ2

Alec McGlasson and Adam Case(advisor)

Drake University

[alec.mcglasson@drake.edu](mailto:alec.mcglasson@drake.edu), [adam.case@drake.edu](mailto:adam.case@drake.edu)

**INTRODUCTION**

Drawing visual representations of dynamical systems by hand is typically a very challenging task. Researchers often use computer graphing software to automate this process. This paper describes software, developed by the authors in C#, that can be used to define functions F: ℝ2 → ℝ2 and generate their trajectories on a 2-D plane. This software, *DSGraph*, uses the library mXParser to parse the definitions of functions specified by the user. This software also makes use of ScottPlot, a library used to create 2D graphs, to visualize trajectories of a dynamical system. Future work will be discussed, including the development of analytical tools that will aid researchers in this field.

**DYNAMICAL SYSTEMS IN ℝ2**

A dynamical system in ℝ2 is defined by a function F: ℝ2 → ℝ2. An *n-trajectory* of F, given an initial condition (X0, Y0), is a sequence (X0, Y0), (X1, Y1), (X2, Y2), . . . , (Xn, Yn) such that

F(X0, Y0) = (X1, Y1), F(X1, Y1) = (X2, Y2), . . . , F(Xn-1, Yn-1) = (Xn, Yn).

Mathematicians are often interested in graphing these trajectories to provide a visual representation of a dynamical system. For example, the *Hénon attractor*, discovered by Michel Hénon [4] is a dynamical system T: ℝ2 → ℝ2 defined by

T(x, y) = (1 + y – 1.4x2, 0.3x).

By graphing trajectories of T, given a particular initial condition, we obtain the image found in Figure 1 (generated in DSGraph).

**THE SOFTWARE**

DSGraph was programmed in C# using the Microsoft Visual Studio .Net Development Framework. Visual Studio’s GUI customization features helped to expedite the development of DSGraph’s user interface. The software uses mXParser [2], a math library used to parse and calculate user-defined functions in ℝ2, and ScottPlot [3], a library that allows the visualization of trajectories of dynamical systems.

ScottPlot and mXParser provide the core functionality of DSGraph. The mXparser library gives us the ability to define and calculate functions in a way that intuitive to mathematicians and those who are not familiar with programming. ScottPlot gives the user the ability to plot multiple functions on an interactive 2D plane. The integration of these libraries significantly reduced the development time of DSGraph.

MXparser’s functionality is essential to the back end of DSGraph because it can create user-defined functions by parsing a string that is entered into a text box located in the user interface. After these functions are defined, they can be calculated with any input by calling a built-in function. This library also provides the user with common trigonometric, unary, and binary functions, which adds to DSGraph’s versatility.

ScottPlot’s functionality is essential to the front end of DSGraph because it is the main viewing panel. In ScottPlot, the points of a plot are defined by populating two arrays, where the first array contains the x values and the second array contains the y-values. The trajectories of a dynamical system are plotted using a PlottableScatter object, which is a basic scatter plot. ScottPlot allows the user to update a trajectory by changing the values in the arrays related to it. Another feature of ScottPlot is its user control functionality, which allows the user to drag the plot around, zoom, define a viewing window by either highlighting a region by middle-clicking and dragging or going into ScottPlot’s settings window.

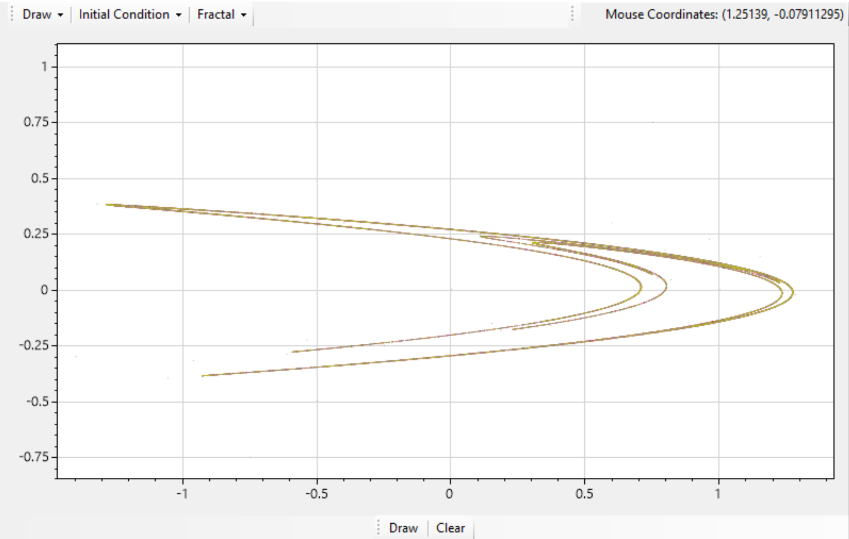


Figure 1: The Hénon attractor generated in DSGraph

**FUTURE WORK**

DSGraph is still being developed and is currently in its alpha stage. The software currently supports a variety of tools for drawing the trajectories of dynamical systems in ℝ2. However, future versions will include additional tools to aid researchers in the analysis of these systems.

**Trajectory compression.** Developing a tool that approximates the compressibility of individual trajectories may yield insights into the orbit complexities and topological entropies of dynamical systems that exist within computable metric spaces [1]. This tool will likely use standard algorithms (e.g., LZ77) for compressing strings of data.

**Drawing in spaces other than ℝ2.** Currently, DSGraph only produces visual representations of dynamical systems in ℝ2. Providing researchers with the ability to define and draw dynamical systems in ℝ3 would be valuable to researchers who study 3D dynamical systems. In addition, allowing users to visualize dynamical systems in the complex plane would be of interest to mathematicians who study the Mandelbrot Set or Julia sets.

**REFERENCES**

[1] Galatolo, S., Hoyrup, M., Rojas, C., Effective symbolic dynamics, random points, statistical behavior, complexity and entropy, Information and Computation, 208, (1), 23-41, 2010.

[2] Gromada, M., mXparser, 2017, <http://mathparser.org/>, retrieved February 2020.

[3] Harden, S., ScottPlot, 2020, <https://github.com/swharden/ScottPlot>, retrieved February 2020.

[4] Hénon, M., A Two-dimensional Mapping with a Strange Attractor, *Communications in Mathematical Physics*, 50, (1), 69-77, 1976.