Hénon maps

The Hénon maps are a generalization of the function $z \mapsto z^2 + c$ to two complex variables. The Hénon map $H : \mathbb{C}^2 \to \mathbb{C}^2$ is given by

$$H \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x^2 - ay + c \\ x \end{pmatrix},$$

with inverse

$$H^{-1} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{a} (y^2 + c - x) \\ y \end{pmatrix},$$

where $a$ and $c$ are both complex parameters.
Related sets

- $K^+$ and $K^-$ are the sets of points with bounded orbits under $H$ and $H^{-1}$, respectively.
- The filled Julia set is $K = K^+ \cap K^-$. 
- $J^+$ and $J^-$ are the boundaries of $K^+$ and $K^-$, respectively.
- $J = J^+ \cap J^-$ is the Julia set.
Related functions

Let

\[
G^+ \left( \begin{pmatrix} x \\ y \end{pmatrix} \right) = \lim_{n \to \infty} \frac{1}{2^n} \log + \left\| H^n \left( \begin{pmatrix} x \\ y \end{pmatrix} \right) \right\|
\]

\[
G^- \left( \begin{pmatrix} x \\ y \end{pmatrix} \right) = \lim_{n \to \infty} \frac{1}{2^n} \log + \left\| H^{-n} \left( \begin{pmatrix} x \\ y \end{pmatrix} \right) \right\|.
\]

\(G^+\) and \(G^-\) measure how quickly iterates of \(H\) (the Hénon map) and \(H^{-1}\) go to infinity.

The surfaces we studied were

\[
X_t = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{C}^2 \mid G^+ \left( \begin{pmatrix} x \\ y \end{pmatrix} \right) = G^- \left( \begin{pmatrix} x \\ y \end{pmatrix} \right) = t \right\},
\]

where \(t\) is a real nonnegative parameter. As \(t \to 0\), \(X_t\) approaches the Julia set \(J\).
What is known about $X_t$

For large $t$, it is known that $X_t$ is homeomorphic to a torus (and as a result has Euler characteristic 0).

Why: If $x$ and $y$ are both large, then $G^+(x,y) \approx \log |x|$ and $G^-(x,y) \approx \log |y|$, so if $t$ is large, then $X_t$ consists of points where $\log |x| \approx \log |y|$, which means that $|x| \approx |y| \approx e^t$, and a point of $X_t$ is determined by the arguments of $x$ and $y$. This roughly describes a torus.
What does $X_t$ look like?
Strategy for computing $X_t$

- **Goal:** Write a program, Cubism, which, given particular values for $a$, $c$, and $t$, generates an approximation of $X_t$ and then uses the approximation to compute the Euler characteristic of each connected component of $X_t$.

- Take a region $R$ in $\mathbb{R}^4$ ($\mathbb{C}^2$), subdivide this region into many four-dimensional cubes, and determine which cubes contain a part of the surface $X_t$.

- The collection of cubes $S$ that are found is the desired approximation of $X_t$. 
Strategy for analysis of $X_t$

- In the case of the surfaces $X_t$, we can classify the connected components of the surfaces by their Euler characteristics.

- Since $S$ should be a thickening of $X_t$, we can compute the Euler characteristic of $X_t$ by computing the Euler characteristic $\chi$ of $S$ using the formula

$$\chi(S) = \sum_{k=0}^{4} (-1)^k n_k,$$

where $n_k$ is the number of $k$-dimensional faces in $S$.

- Also, we color the three-dimensional projections of $X_t$ by using the missing dimension.
Finding $S$

- Make some subdivision of $R$ into four-dimensional cubes.
- Run through each two-dimensional face that touches at least one four-dimensional cube in $R$.
- The intersections of a two-dimensional face with the sets $c^+ = \{G^+ = t\}$ and $c^- = \{G^- = t\}$ are each one-dimensional curves. If these curves intersect each other in the face in question, then all 4 four-dimensional cubes which touch this face are added to $S$.
- How to *efficiently* determine whether or not $c^+$ intersects $c^-$ in a particular face?
Vertex decisions

- Examine the signs of the functions $G^+ - t$ and $G^- - t$ at the vertices of the face.
- Certain combinations of signs at the vertices can ensure that $c^+$ will intersect $c^-$ on a face.

For example, say that the signs on a face are as below:

\[
\begin{array}{cc}
G^+ : + & G^+ : - \\
G^- : + & G^- : + \\
\end{array}
\]

\[
\begin{array}{cc}
G^+ : + & G^+ : - \\
G^- : - & G^- : - \\
\end{array}
\]
**Vertex decisions: a “yes” face**

In this case, we can be sure that $c^+$ and $c^-$ intersect in that face, looking something like this:

In the case above and in similar cases (when we can be sure that there is an intersection), we mark the face in question as a “yes”.
Vertex decisions: a “maybe” face

On the other hand, this combination of signs

\[
\begin{array}{c|c}
G^+ : + & G^+ : - \\
G^- : + & G^- : + \\
\end{array}
\]

could manifest itself in any of the following situations:
Problems with vertex decisions

If this algorithm marks a face as a “yes”, then \( c^+ \) and \( c^- \) definitely intersect within the face. But even if we consider \( S \) to be the set of all “yes” cubes and all “maybe” cubes, we still miss cubes that should be in \( S \).

Either \( G^+ - t \) or \( G^- - t \) could change signs twice, resulting in signs at the vertices that are identical to the case where the function does not change sign at all. (See the diagram at right.)
A picture with gaps

Using only the vertex decision method resulted in approximations of $X_t$ that had gaps in them (cubes that should have been in $S$ but weren’t):
Getting around problems with vertex decisions

- Given an infinite amount of time (and/or computational power), we could fix the problem by making a finer subdivision of $R$.
- But Cubism takes time $O(N^4)$, if $N$ is the number of subdivisions on a side, so it wasn’t reasonable to simply increase $N$.
- Instead, we needed a better algorithm.
Newton’s Method

- Using a two-dimensional version of Newton’s method, we attempt to find a common root of the functions $G^+ - t$ and $G^- - t$, with the initial point in the center of the face.

- We iterate Newton’s method a fixed number of times (5) and then try to determine whether Newton’s method is converging to a root.

- If it is, then there is an intersection of $c^+$ and $c^-$ in the face, and we mark the face as a “yes”.
Determining whether Newton’s Method is converging

- It is difficult to determine whether or not Newton’s method is really converging to a root within the face. (Could mistake convergence of Newton’s Method to a root on the face in question for converging to a root that is on another face.)

- Our criteria for convergence: if the final iterate of Newton’s method is still within the face it started in, \( and \) both \( G^+ \) and \( G^- \) are within \( \epsilon \) of \( t \), then we mark this face as a “yes”. (Set \( \epsilon = .01 \).)

- The second condition attempts to ensure that Newton’s method is in fact converging, and the first condition attempts to ensure that Newton’s method is converging to a root inside the face it started in.
Subdivision and Refinement

- If we choose an initial subdivision $N$ (so that each side of $R$ is divided into $N$ subdivisions), then we end up with $N^4$ cubes.

- A larger $N$ results in a better approximation of the surface $X_t$, so we want to run Cubism for as large an $N$ as possible.

- We use a method that allows us to start with a reasonable initial subdivision (i.e., not too large), and then “refine” the resulting cubes into smaller and smaller cubes.
Subdivision and Refinement, cont.

Start with some initial subdivision, and let $S_0$ be the resulting set $S$ of cubes marked as a “yes” or a “maybe”. In the lower-dimensional diagram at right, $S_0$ is the set of shaded boxes:

For each four-dimensional cube in $S_0$, perform another subdivision, and let $S_1$ be the resulting collection of cubes. Repeat as desired. At right, $S_1$ is the set of shaded cubes:
Subdivision and Refinement, cont.

- Note that if subdivisions $N_0, N_1, \ldots, N_k$ are used, then the final thickening of $X_t$ is the set $S_k$.

- In theory, we should obtain the same thickening of $X_t$ by doing this procedure or by choosing the sole subdivision to be $N_0 \cdot N_1 \cdot N_2 \cdots N_k$.

- In practice, this doesn’t quite work—if both the vertex decision algorithm and the Newton’s method algorithm mark a face a “no” that should be a “yes” or “maybe”, then not only will that cube not be included in $S$, but that cube will also not be refined (further subdivided).