THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF MATHEMATICS

STRUCTURE OF ATTRACTOR REGION FOR NATURAL EXTENSION MAP ASSOCIATED WITH P-ADIC CONTINUED FRACTIONS

ADAM ZYDNEY
Spring 2011

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Mathematics
with honors in Mathematics.

Reviewed and approved* by the following:

Svetlana Katok
Professor of Mathematics
Thesis Supervisor, Honors Advisor

Misha Guysinsky
Senior Lecturer
Thesis Supervisor

* Signatures are on file in the Schreyer Honors College.

Abstract

This thesis investigates the behavior of repeated iterations of a map F on two p-adic numbers, ultimately proving the existence of an attractor region for this map and describing that region. A description is also given of the behavior of the map on points inside and outside of the attractor region. First, the p-adic numbers are introduced as an alternative (to the real numbers) completion of the rationals, and definitions are given for canonical expansions, p-adic integers, and p-adic units. Also laid out are the means for arithmetic of p-adic numbers relevant to computing F.

With the addition of a non-injective "digit reversing" map from p-adic numbers to real numbers, a computer program (written in Java) produces several graphs of the behavior of F for various initial points. Using these plots as a guideline, analytic proofs are then developed to rigorously show the behavior of F, specifically the existence and description of its attractor region.

Table of Contents

List of Figures

Acknowledgements

I would like to thank my advisor, Professor Svetlana Katok, for her guidance and support, as well as for proposing this project. I would also like to thank Misha Guysinsky for helping with the specifics of this research, which was done in Summer 2009 as part of Penn State's Research Experience for Undergraduates program in collaboration with Natasha Potashnik and Steve Wolf. Lastly, I am grateful to my family and friends for all of their support throughout my undergraduate career.

1. Introduction to p-adic Numbers

The real numbers $\mathbb{R}$ are the completion of the rational numbers using the Euclidean norm. The p-adic numbers, denoted $\mathbb{Q}_p$, are the completion of the rationals using the p-adic norm defined below.

1.1. Norm and Distance

For any prime p, the order of a non-zero integer x is defined as the greatest power of p which divides x. Put another way,

$$\text{For } x = c \cdot p^n, \text{ where } c \in \mathbb{Z},\ n \in \mathbb{N}, \text{ and } (c, p) = 1,\ ord_p\, x = n.$$

For integers $x \in \mathbb{Z}$, the p-adic norm is defined as

$$|x|_p = \begin{cases} 0 & \text{if } x = 0, \\ p^{-ord_p x} & \text{if } x \neq 0, \end{cases}$$

and for rational numbers $x = a/b$, $a, b \in \mathbb{Z}$, $|x|_p = |a|_p/|b|_p$.

For any integer $x = c\,p^n$ as defined before, $|x|_p = p^{-n}$.

Notice that $|\cdot|_p$ can take only a discrete set of values, $\{p^n, n \in \mathbb{Z}\} \cup \{0\}$.

Since $\mathbb{Q}$ is a field, the distance induced by the p-adic norm is

$$d_p(x, y) = |x - y|_p.$$

1.2. p-adic Expansions

Recall that a sequence $\{x_n\}$ converges to a if

$$\lim_{n\to\infty} d(x_n, a) = 0.$$

So whether a sequence converges or not depends on the distance being used.

Consider the sequence $\{x_n\} = \{p^n\}$. Under the Euclidean norm (absolute value), this sequence clearly diverges. But under the p-adic norm, we know that

$$d_p(x_n, 0) = |x_n|_p = |p^n|_p = p^{-n}.$$

$\lim_{n\to\infty} p^{-n} = 0$, so the sequence $\{p^n\}$ actually converges to 0!

Consider the infinite series

$$d_{-m}p^{-m} + d_{-m+1}p^{-m+1} + \cdots + d_0 + d_1 p + d_2 p^2 + \cdots\ ,$$

where $d_i \in \{0, 1, \ldots, p-1\}$ and $d_{-m} \neq 0$. The partial sums of this series, $S_n = \sum_{-m}^{n} d_i p^i$, form a Cauchy sequence.

Because $\mathbb{Q}_p$ is complete, S_n converges in $\mathbb{Q}_p$ (i.e., there is an $a \in \mathbb{Q}_p$ such that $d_p(S_n, a) \to 0$ as $n \to \infty$). So every series of the form $\sum_{-m}^{\infty} d_i p^i$ represents a single $a \in \mathbb{Q}_p$. As it turns out, the converse is also true − every element of $\mathbb{Q}_p$ can be represented by a unique series of that form, called the of a p-adic number, called the canonical p-adic expansion of a.

From now on, every number $x \in \mathbb{Q}_p$ will be expressed as an expansion in base p:

$$x = \ldots a_n \ldots a_1 a_0 . a_{-1} a_{-2} \ldots a_{-m} = \sum_{i=-m}^{\infty} a_i p^i,$$

where $a_i \in \{0, 1, \ldots, p-1\}$ and $a_{-m} \neq 0$. The dot between a_0 and a_{-1} is called a "radix point," the generalization of the decimal point to base p expansions.

There are several important properties of this expansion:

- Expansions have a finite number of digits to the *right* of the radix point, but they can have an infinite number of digits to the *left*.
- The order and the norm of x are immediately ascertainable from this expansion: $ord_p x = -m$ and $|x|_p = p^m$.
- No $\pm$ sign is needed – even negative numbers can be expressed this way.
- This representation is unique – there is exactly one expansion for every number.

For positive rational numbers whose ordinary (Euclidean) base p expansions terminate, the p-adic expansions are identical. For example, for $k \in \mathbb{N}$, the p-adic expansion of p^k is $1\underbrace{0\ldots0}_{k \text{ zeros}}$, and $\frac{1}{p^k}$ is $0.\underbrace{0\ldots0}_{k \text{ zeros}}1$.

Rational numbers whose real base p expansions do not terminate are more complicated.

1.3. *p*-adic Integers and Units

The set of p-adic integers $\mathbb{Z}_p$ is the completion of the integers with respect to the p-adic norm. They can be equivalently defined as any of the following:

$$\begin{aligned} \mathbb{Z}_p &= \{\, x \in \mathbb{Q}_p \,|\, ord_p x \geq 0 \,\} \\ &= \{\, x \in \mathbb{Q}_p \,|\, |x|_p \leq 1 \,\} \\ &= \{\, x \in \mathbb{Q}_p \,|\, x = \ldots a_2 a_1 a_0 . \,\} \end{aligned}$$

It is worth noting that there are numbers in $\mathbb{Z}_p \cap \mathbb{Q}$ which are not in $\mathbb{Z}$. These include any fractions a/b where $a, b \in \mathbb{Z}$, $b > 0$, and b is relatively prime to p; the solutions to

$x^2 = 2$, $x \in \mathbb{Q}_p$, $p \neq 2$; and numbers such as $\sum_{i=0}^{\infty} p^{i(i+3)/2} = \ldots 1000100101$, which have no real analogues at all.

The set of p-adic units $\mathbb{Z}_p^\times$ is the set of invertible elements of $\mathbb{Z}_p$. Equivalently,

$$\begin{aligned}
\mathbb{Z}_p^\times &= \{\, x \in \mathbb{Q}_p \,|\, ord_p x = 0 \,\} \\
&= \{\, x \in \mathbb{Q}_p \,|\, |x|_p = 1 \,\} \\
&= \{\, x \in \mathbb{Q}_p \,|\, x = \ldots a_2 a_1 a_0.,\, a_0 \neq 0 \,\}
\end{aligned}$$

Any p-adic number x can be written as $x = p^n u$, $u \in \mathbb{Z}_p^\times$, where $n = ord_p x$. Also, $x \in p\mathbb{Z}_p$ if and only if $x^{-1} \in \mathbb{Q}_p \setminus \mathbb{Z}_p$. One simple explanation of this is that

$$|x|_p |x^{-1}|_p = |xx^{-1}|_p = |1|_p = 1,$$

so $|x|_p < 1$ if and only if $|x^{-1}|_p > 1$.

1.4. Arithmetic of p-adic Numbers

Addition, subtraction, and multiplication can all be calculated with p-adic expansions in much the same way as they are with Euclidean base p expansions.

Finding $-x$ for $x = \ldots a_2 a_1 a_0$ amounts to finding the $(p-1)$'s complement of x up until the left-most non-zero digit. For example, the p-adic expansion of -1 is an infinite tail of $(p-1)'s$ to the left of the radix point. It is convenient to define the digit $q = p-1$ so that -1 can be expressed as $\ldots qqq$. Note that any subtraction can be done using addition because p-adic expansions do not include $+$ or $-$ signs. For example, $x - 1 \equiv x + \ldots qqq$.

Division of p-adic numbers is a little different from Euclidean division. Rather than provide a general division algorithm however, we will work exclusively with the algorithm for finding a reciprocal. One must only worry about finding reciprocals of p-adic units because for $x \notin \mathbb{Z}_p^\times$, $x = p^n u$, $u \in \mathbb{Z}_p^\times$, and $x^{-1} = p^{-n} u^{-1}$.

To find the reciprocal of $u = \ldots a_2 a_1 a_0.$, $a_0 \neq 0$:

1. Let $d = \ldots d_1 d_0 = u$.
2. For each digit a_i of u (starting with $i = 0$):
    i. Find the digit c_i for which $c_i \cdot d_0 \equiv 1 \pmod{p}$.
    ii. Redefine d as $d = p(d_{old} - c_i \cdot u) \in \mathbb{Z}_p$.
3. The p-adic unit $\ldots c_2 c_1 c_0. = u^{-1}$.

2. The Natural Extension Map

The main focus of this thesis is the properties of the *natural extension map* $F\colon \mathbb{Q}_p \times \mathbb{Q}_p \to \mathbb{Q}_p \times \mathbb{Q}_p$ given by

$$F(x,\, y) = \begin{cases} \left(\frac{1}{x},\, \frac{1}{y}\right) & \text{if } x \in p\mathbb{Z}_p \\ \left(x - \frac{1}{|x|_p},\, y - \frac{1}{|x|_p}\right) & \text{if } x \in \mathbb{Q}_p \setminus \mathbb{Z}_p \end{cases}$$

In particular, we wish to describe a set which is an attractor region for F.

To prove that a set $A \subset \mathbb{Q}_p^2$ is an attractor region, we must prove

Condition 1. For every $(x,\, y) \in \mathbb{Q}_p^2\ \exists\ N \geq 0$ s.t. $F^N(x,\, y) \in A$.
Condition 2. If $(x,\, y) \in A$, then for any $n > 0$, $F^n(x,\, y) \in A$.
Condition 3. There is no subset $A' \subset A$ for which Conditions 1 and 2 hold.

Note that F is not well defined when $x = 0$. Also, the notations $(x',\, y') = F(x,\, y)$ and $(x^{(n)},\, y^{(n)}) = F^n(x,\, y)$ are often used.

2.1. Motivation From Continued Fractions

The map F is the natural extension of the one dimensional map $f : \mathbb{Q}_p \to \mathbb{Q}_p$ given by

$$f(x) = \begin{cases} 1/x & \text{if } x \in p\mathbb{Z}_p \\ x - \frac{1}{|x|_p} & \text{if } x \in \mathbb{Q}_p \setminus \mathbb{Z}_p \end{cases}$$

This function is analogous to the real function $g : \mathbb{R}^+ \to \mathbb{R}^+$

$$g(x) = \begin{cases} 1/x & \text{if } 0 < x < 1 \\ x - 1 & \text{if } x \geq 1 \end{cases}$$

which has significant connections to real continued fractions as described by Katok and Ugarcovici [2].

2.2. Terminating Cases

In some cases, $F^n(x,\, y)$ does not end up in A because the iteration of F terminates first. That is, there is an $N > 0$ for which $F^{N+1}(x,\, y)$ is not well-defined. For the natural extension map, this is when $F^N(x,\, y) \in \{0\} \times \mathbb{Q}_p$.

Several points will eventually encounter this problem. Any $x = \sum_{i=-n}^{0} a_i p^i = a_0.a_1 \ldots a_{-n}$ will eventually lead to $x^{(m)} = 0$ because F only subtracts from the one's place and from places to the right of the radix point.

Although it is convenient to say that $F : \mathbb{Q}_p^2 \to \mathbb{Q}_p^2$, F should really be defined only on the subset of $\mathbb{Q}_p^2$ for which $F^n(x, y)$ is well-defined for all $n > 0$. In all future references to $\mathbb{Q}_p$ and $\mathbb{Q}_p^2$, it is assumed that points for which $F^n(x, y)$ is eventually not well-defined are not included.

3. Graphing p-adic Numbers

Graphing a p-adic number is inherently difficult because $\mathbb{Q}_p$ is unordered. We make heavy use of the following (non-injective) map $\varphi\colon \mathbb{Q}_p \to \mathbb{R}^+$ to investigate F:

$$\varphi(\ldots a_1 a_0.a_{-1} \ldots a_{-m}) = a_{-m} \ldots a_{-1} a_0.a_1 \ldots$$
$$\varphi\left(\sum_{i=-m}^{\infty} a_i p^i\right) = \sum_{i=-m}^{\infty} a_i p^{-i}$$

Since we will be graphing points on a plane, we will also use $\varphi\colon \mathbb{Q}_p^2 \to \mathbb{R}^{+2}$ defined as simply $\varphi(x, y) = (\varphi(x), \varphi(y))$.

Because $\mathbb{Z}_p = \mathbb{Z}_p^\times \sqcup p\mathbb{Z}_p$, any p-adic number is in one of three sets: $\mathbb{Z}_p^\times$, $p\mathbb{Z}_p$, or $\mathbb{Q}_p \setminus \mathbb{Z}_p$. The map φ can easily differentiate between these three sets:

$$\begin{array}{llll}
\text{For } x \in p\mathbb{Z}_p, & ord_p x > 0, & |x|_p < 1, & \text{and } \ 0 \le \varphi(x) < 1. \\
\text{For } x \in \mathbb{Z}_p^\times, & ord_p x = 0, & |x|_p = 1, & \text{and } \ 1 \le \varphi(x) < p. \\
\text{For } x \in \mathbb{Q}_p \setminus \mathbb{Z}_p, & ord_p x < 0, & |x|_p > 1, & \text{and } \ p \le \varphi(x) < \infty.
\end{array}$$

3.1. Multiple Expansions

One significant difference between $\mathbb{Q}_p$ and $\mathbb{R}$ is that one real number can have two different expansions while p-adic numbers cannot. Consider the following two expansions of a rational number r in base p:

$$\begin{array}{ll}
r_1 = a_{-m} \ldots a_{-1} a_0.a_1 \ldots a_{n-1} a_n, & a_n \neq 0 \\
r_2 = a_{-m} \ldots a_{-1} a_0.a_1 \ldots a_{n-1} a'_n qqq \ldots, & \text{where } a'_n = a_n - 1
\end{array}$$

The pre-image of r under φ consists of two distinct p-adic numbers:

$$\begin{aligned}
\varphi(x_1) &= \varphi(a_n a_{n-1} \ldots a_1 a_0.a_{-1} \ldots a_{-m}) = r_1 \\
\varphi(x_2) &= \varphi(\ldots qqq a'_n a_{n-1} \ldots a_1 a_0.a_{-1} \ldots a_{-m}) = r_2
\end{aligned}$$

Since F operates either by inversion or by subtraction of $\frac{1}{|x|_p}$, we only need to consider how x_1 and x_2 will be treated by these two operations. Specifically, we need to see if x_1 and x_2 have the same norm.

The order of x is the smallest (most negative) i for which $a_i \neq 0$. If $i \neq n$, then a_i is the same for x_1 and x_2, and $|x_1|_p = |x_2|_p = p^{-i}$. If $ord_p x_1 = n$, there are two possibilities: (1) $a_n = 1$, in which case $|x_1|_p = p^{-n}$ and $|x_2|_p = p^{-n-1}$, or (2) $a_n \neq 1$, in which case $|x_1|_p = |x_2|_p = p^{-n}$.

So $|x_1|_p \neq |x_2|_p$ occurs only when $ord_p x_1 = n$ and $a_n = 1$.

If $n \geq 0$,

$$x_1 = 1\underbrace{0\ldots0}_{n \text{ zeros}} = p^n$$

$$x_2 = \ldots qq\underbrace{00\ldots0}_{n+1 \text{ zeros}} = -p^{n+1}$$

If $n < 0$,

$$x_1 = 0.\underbrace{0\ldots0}_{n-1}1 = p^n$$

$$x_2 = \ldots qqq.\overbrace{q\ldots q} = -p^n$$

$F^n(x_1, y)$ will terminate within two iterations for both cases above. $F^n(x_2, y)$ will be periodic in x.

3.2. Graphical Interpretation of F

Here we consider the possible relationships between $\varphi(x, y)$ and $\varphi(F(x, y))$ when F operates by subtraction. Let

$$x = \ldots a_1 a_0.a_{-1}\ldots a_{-n} \text{ and}$$

$$y = \ldots b_1 b_0.b_{-1}\ldots b_{-m}.$$

Then $(x', y') = (x - \frac{1}{|x|_p}, y - \frac{1}{|x|_p}) = (x - p^{-n}, y - p^{-n})$. Also let

$$\Delta x = \varphi(x') - \varphi(x) \text{ and}$$

$$\Delta y = \varphi(y') - \varphi(y).$$

This notation is a little odd in that Δx is a real number even though x is p-adic, but writing $\Delta\varphi(x)$ would be even worse.

Theorem 1. Let $\Delta x = \varphi(x') - \varphi(x)$ and $\Delta y = \varphi(y') - \varphi(y)$, and assume that F operates by subtraction.

(i) In all cases, $\Delta x = -|x|_p$.
(ii) If $b_n \neq 0$, then $\Delta y < 0$ and $\Delta y = \Delta x$.
(iii) If $b_n = 0$, then $\Delta y > 0$ and $|\Delta y| > |\Delta x|$.

These statements each have a graphical interpretation:

(i) F always moves the image under φ of (x, y) to the left by $|x|_p$.
(ii) When $b_n \neq 0$, $\varphi(x, y)$ moves downward along a line of slope 1.
(iii) When $b_n = 0$, $\varphi(x, y)$ moves upward along a line whose slope has an absolute value greater than 1.

Proof of Part (i): This is pure digit manipulation:

$$\begin{aligned}
\varphi(x') &= (a_{-n}-1)a_{-n+1}\ldots a_{-1}a_0.a_1 \quad \ldots \\
\varphi(x) &= \quad\quad a_{-n}\ a_{-n+1}\ldots a_{-1}a_0.a_1 \quad \ldots \\
\Delta x &= \quad\quad -1 \quad\ 0 \quad \ldots\ 0\ \ 0\,. \\
&= -p^n \\
&= -|x|_p \text{ because } n = -ord_p x
\end{aligned}$$

□

Proof of Part (ii): Since $ord_p x = n$, b_n must be non-zero if $ord_p x = ord_p y$. Then

$$\begin{aligned}
x &= \ldots a_1 a_0.a_{-1}\ldots a_{-n+1}a_{-n} \\
y &= \ldots b_1 b_0.b_{-1}\ldots b_{-n+1}b_{-n} \\
\text{Subtract} &\quad\quad 0\,.\ 0\ \ldots \quad 0 \quad 1 \\
y' &= \ldots b_1 b_0.b_{-1}\ldots b_{-n+1}(b_{-n}-1)
\end{aligned}$$

$$\begin{aligned}
\varphi(y') &= (b_{-n}-1)b_{-n+1}\ldots b_0.b_1\ldots \\
\varphi(y) &= \quad\quad b_{-n}\ b_{-n+1}\ldots b_0.b_1\ldots \\
\Delta y &= \quad\quad -1 \quad\ 0 \quad \ldots 0\,. = \Delta x
\end{aligned}$$

If $ord_p x < ord_p x$, we must specify that $b_n \neq 0$. In this case,

$$\begin{aligned}
x &= \ldots a_1 a_0.a_{-1}\ldots a_{-n+1} \quad a_{-n} \\
y &= \ldots b_1 b_0.b_{-1}\ldots b_{-n+1} \quad b_{-n} \quad \ldots b_{-m} \\
\text{Subtract} &\quad\quad 0\,.\ 0\ \ldots \quad 0 \quad\quad 1 \\
y' &= \ldots b_1 b_0.b_{-1}\ldots b_{-n+1}(b_{-n}-1)\ldots b_{-m}
\end{aligned}$$

$$\begin{aligned}
\varphi(y') &= b_{-m}\ldots(b_{-n}-1)b_{-n+1}\ldots b_0.b_1\ldots \\
\varphi(y) &= b_{-m}\ldots \quad b_{-n} \quad b_{-n+1}\ldots b_0.b_1\ldots \\
\Delta y &= \quad\quad\quad -1 \quad\quad 0 \quad \ldots 0\,. = \Delta x
\end{aligned}$$

□

Proof of Part (iii): As stated before, b_n cannot equal 0 if $ord_p x = ord_p y$. If $ord_p x < ord_p y$, we must specify that $b_n = 0$. Then

$$
\begin{array}{rl}
x = & \ldots a_1 a_0 . a_{-1} \ldots a_{-n+1} a_{-n} \\
y = & \ldots b_1 b_0 . b_{-1} \ldots b_{-n+1} \;\; 0 \;\; \ldots b_{-m} \\
\text{Subtract} & \quad 0\,.\; 0 \;\ldots \quad 0 \quad 1 \\
y' = & \ldots b'_1 b'_0 . b'_{-1} \ldots b'_{-n+1} \;\; q \;\; \ldots b_{-m}
\end{array}
$$

$$
\begin{array}{rl}
\varphi(y') = & b_{-m} \ldots q b'_{-n+1} \ldots b'_0 . b'_1 \ldots \\
\varphi(y) = & b_{-m} \ldots 0 b_{-n+1} \ldots b_0 . b_1 \ldots \\
\Delta y = & \qquad q\, 0 \quad \ldots 0\,. \\
\Delta y > & \qquad 1\, 0 \quad \ldots 0\,. = -\Delta x
\end{array}
$$

If $ord_p x > ord_p y$, b_n must be 0, and

$$
\begin{array}{rlllll}
x = & \ldots a_1 a_0 . a_{-1} \ldots & a_{-m} & a_{-m-1} \ldots b_{-n+1} b_{-n} & & \\
y = & \ldots b_1 b_0 . b_{-1} \ldots & b_{-m} & & & \\
\text{Subtract} & \quad 0\,.\; 0 \;\ldots & 0 & 0 \quad \ldots & 0 & 1 \\
y' = & \ldots b_1 b_0 . b_{-1} \ldots & (b_{-m} - 1) & q \quad \ldots & q & q
\end{array}
$$

$$
\begin{array}{rlll}
\varphi(y') = & q \ldots \quad q & (b_{-m} - 1) & b_{-m+1} \ldots b_0 . b_1 \ldots \\
\varphi(y) = & & b_{-m} & b_{-m+1} \ldots b_0 . b_1 \ldots \\
\Delta y = & q \ldots (q-1) & q & 0 \quad \ldots 0\,. \\
\Delta y > & & 1 & 0 \quad \ldots 0\,. = -\Delta x
\end{array}
$$

$\square$

4. Describing the Attractor Region

The attractor region A is comprised of three disjoint sections:

$$A = A^* \bigsqcup \Big((\mathbb{Q}_p \setminus \mathbb{Z}_p) \times p\mathbb{Z}_p \Big) \bigsqcup \Big\{ (x,\, y) \in \mathbb{Q}_p^2 \;\Big|\; y = x \Big\}$$

Although A can be dealt with using p-adic expansions exclusively, it is useful and quite interesting to examine the image of A under φ. See Figures 2 through 5 on pages 10 through 13 for more graphs of $\varphi(A)$ for various $\mathbb{Q}_p$ at various magnifications.

4.1. Using Real Image

A^* is most easily defined via its image under φ and the new function $\ell : \mathbb{N} \to \mathbb{N}$

$$\ell(k) = \begin{cases} k - 1 & \text{if } k \leq p \\ k \bmod p & \text{if } k > p \text{ and } p \nmid k \\ p\,\ell(k/p) & \text{if } k > p \text{ and } p \mid k \end{cases}$$

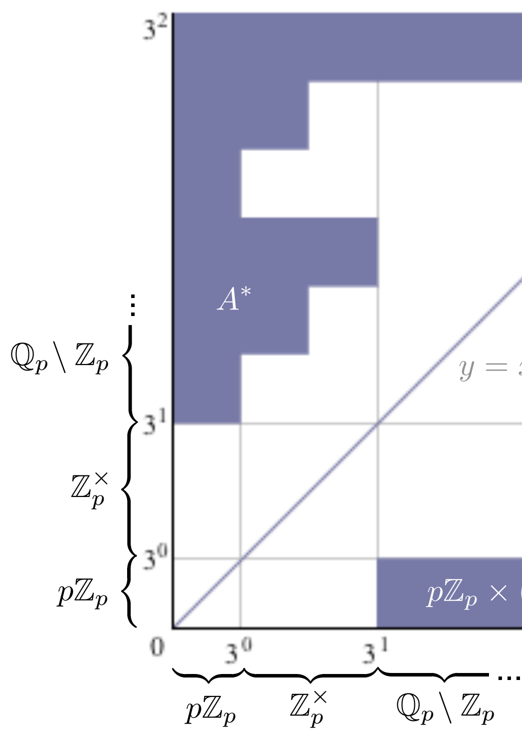

FIGURE 1. Attractor Region for $\mathbb{Q}_3$ out to 3^2

Graphically, $\ell(k)$ is the length of a bar which spans $\varphi(y) \in [k-1, k)$ for $k > p$. For $k \le p$, $\ell(k)$ has no physical significance $-$ its value is simply what is necessary for $p\ell(k/p)$ to evaluate correctly. A^* is then the set of points for which $\varphi(x) < \ell(k)$.

$$A^* = \left\{ (x,\, y) \in \mathbb{Q}_p^2 \;\middle|\; y \notin \mathbb{Z}_p,\, \varphi(x) < \ell\big(\lfloor \varphi(y) \rfloor + 1\big) \right\}$$

This definition of A^* is not ideal because it depends on the map φ. Also, what $\varphi(x) \le \ell\big(\lfloor \varphi(y) \rfloor + 1\big)$ says about the p-adic numbers x and y is unclear.

4.2. Using p-adic Expansions

There is an alternative description of A^* which uses only p-adic expansions. This is an improvement over only describing $\varphi(A^*)$, but this purely p-adic description is really no more elegant or intuitive.

Let $y = \ldots y_1 y_0 . y_{-1} \ldots y_{-N} \in \mathbb{Q}_p \setminus \mathbb{Z}_p$. Find a digit c and a non-negative integer m as follows:

- If $y_0 \ne q$, then $c = y_0 + 1$ and $m = 0$.

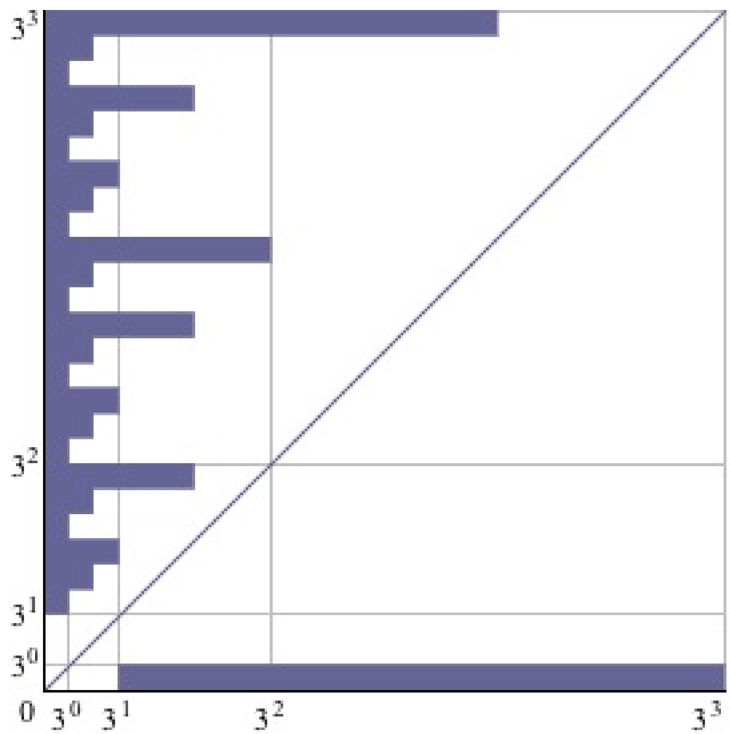

FIGURE 2. Attractor Region for $\mathbb{Q}_3$ out to 3^3

- If $y_0 = q$,
  - Find M such that $y_{-i} = q$ for $i = \{0, \ldots, M-1\}$ and $y_{-M} \neq q$.
  - If $M = N + 1$, then $c = q$ and $m = M - 1$.
  - If $M = N$, then $c = y_{-M}$ and $m = M$.
  - If $M < N$, then $c = y_{-M} + 1$ and $m = M$.

Let $x = \ldots a_1 a_0 . a_{-1} \ldots a_{-n}$.

$$A^* = \left\{ (x,\, y) \in \mathbb{Q}_p^2 \;\middle|\; n < m \right\} \bigcup \left\{ (x,\, y) \in \mathbb{Q}_p^2 \;\middle|\; n = m,\, a_{-n} < c) \right\}$$

This description was derived from the first description of A^*. One relation between the two descriptions is that $\ell(k) = cp^m$.

5. Unit Squares

Since the attractor region's image is made up entirely of unit squares, it is sensible to look at the pre-image under φ of unit squares and intervals.

Theorem 2. F maps unit squares to unit squares, provided that the original unit square is not in the first column.

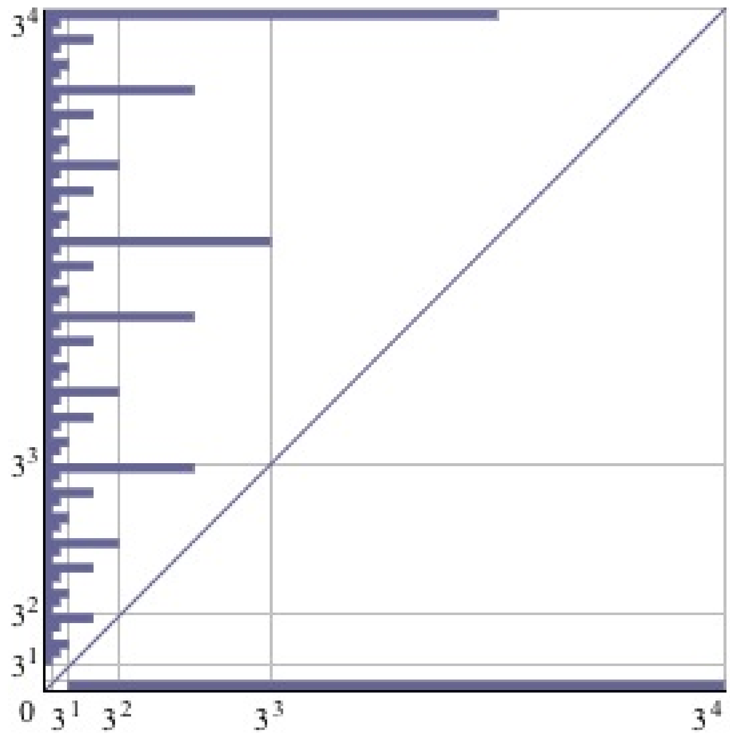

FIGURE 3. Attractor Region for $\mathbb{Q}_3$ out to 3^4

Proof: Consider a unit interval $[k-1,\, k)$, $k \geq 1$. Let $k-1 = a_n \dots a_1 a_0$ in base p. If $\varphi(x) \in [k-1, k)$, then $\varphi(x) = k-1+r$ for some $r \in [0,1)$. Let $r = 0.r_{-1}r_{-2}\dots$ in base p. Then

$$\begin{aligned}
\varphi(x) &= \quad a_n \dots a_1 a_0.r_{-1}r_{-2}\dots \\
x &= \dots r_{-2}r_{-1}a_0.a_1\dots a_n \\
x &= \quad \dots x_2x_1x_0.x_{-1}\dots x_{-n}, \qquad \text{where } x_i = \begin{cases} r_{-i} & i > 0 \\ a_{-i} & i \leq 0 \end{cases}
\end{aligned}$$

By similar reasoning, if $\varphi(y) \in [j-1, j)$, then $j = b_m \dots b_0$, $\varphi(y) = j-1+s$ for some $s \in [0,1)$, and $y = \dots s_{-2}s_{-1}b_0.b_1\dots b_m$. Now $\varphi(x,\, y)$ is in the unit square $[k-1,\, k)\times[j-1,\, j)$.

Assume that $j, k > 1$ (this means the square is not in the first column). From the expansion of x, it is clear that $|x|_p = p^n$, so F will subtract p^{-n} from x and y to get

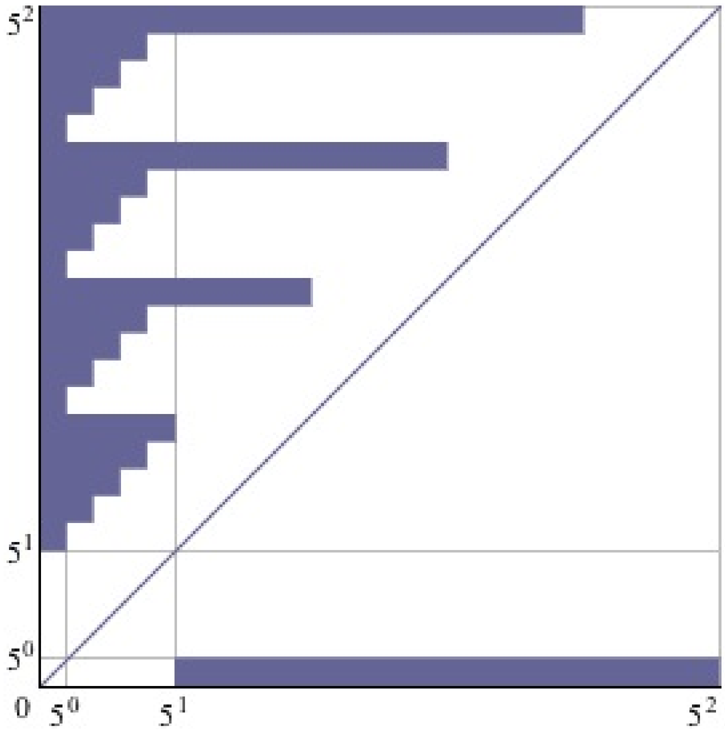

FIGURE 4. Attractor Region for $\mathbb{Q}_5$ out to 5^2

x' and y'. Because $k-1>0$ and $j-1>0$, the subtraction algorithm will stop before r_{-1} and s_{-1}, leaving

$$\begin{aligned} x' &= \ldots r_{-2}r_{-1}a_0'.a_1'\ldots a_n' \\ y' &= \ldots s_{-2}s_{-1}b_0'.b_1'\ldots b_m' \\ \varphi(x') &= a_n'\ldots a_1'a_0'.r_{-1}r_{-2}\ldots = k'+r \text{ for some } k' \\ \varphi(y') &= b_m'\ldots b_1'b_0'.s_{-1}s_{-2}\ldots = j'+s \text{ for some } j' \end{aligned}$$

Thus $(x',\, y') \in [k'-1,\, k') \times [j'-1,\, j')$, which is also a unit square. So F does map unit squares to unit squares, provided that the original unit square is not in the first column. $\square$

In fact, this proof yields an even stronger statement. The translation

$$\Delta x = \varphi(x') - \varphi(x) = (k'+r) - (k+r) = k'-k$$

does not depend on r (and likewise Δy does no depend on s), so F moves every point within a unit square identically to the same location within the new unit square. Figure 6 shows an example in which green points are mapped to blue points.

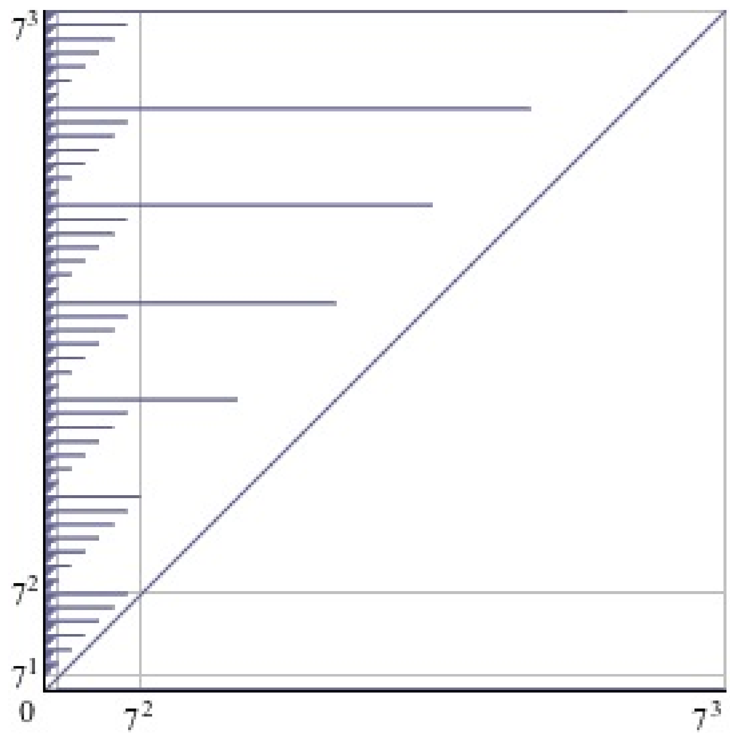

FIGURE 5. Attractor Region for $\mathbb{Q}_7$ out to 7^3

6. Ending Up in the Attractor Region

Condition 1 of A being an attractor region is that for every $(x, y) \in \mathbb{Q}_p^2$ $\exists$ $N \geq 0$ such that $F^N(x, y) \in A$.

6.1. Moving Left

F only inverts when $x \in p\mathbb{Z}_p$, which is when $\varphi(x) < 1$. If $\varphi(x) \geq 1$, F subtracts, which is represented graphically by moving to the left. $\varphi(y')$ could move up or down relative to $\varphi(y)$, but subtracting from the leftmost digit of x leads to a decrease in the rightmost digit of $\varphi(x)$, so $\varphi(x') < \varphi(x)$ for all $\varphi(x) \geq 1$.

Recall that $\mathbb{Q}_p = p\mathbb{Z}_p \sqcup \mathbb{Z}_p^\times \sqcup (\mathbb{Q}_p \setminus \mathbb{Z}_p)$. The first Quadrant (which is the codomain of φ) can therefore be divided easily into nine sections as shown in Figure 7 (the shaded areas are part of the attractor region).

If F moves $\varphi(x, y)$ leftwards until $\varphi(x) < 1$, then $F^n(x, y)$ must at some point end up in one of the three leftmost regions: R_1, R_4, or R_7. $R_1 \subset A$, so we only need to show that points in R_4 and R_7 eventually end up in the attractor region.

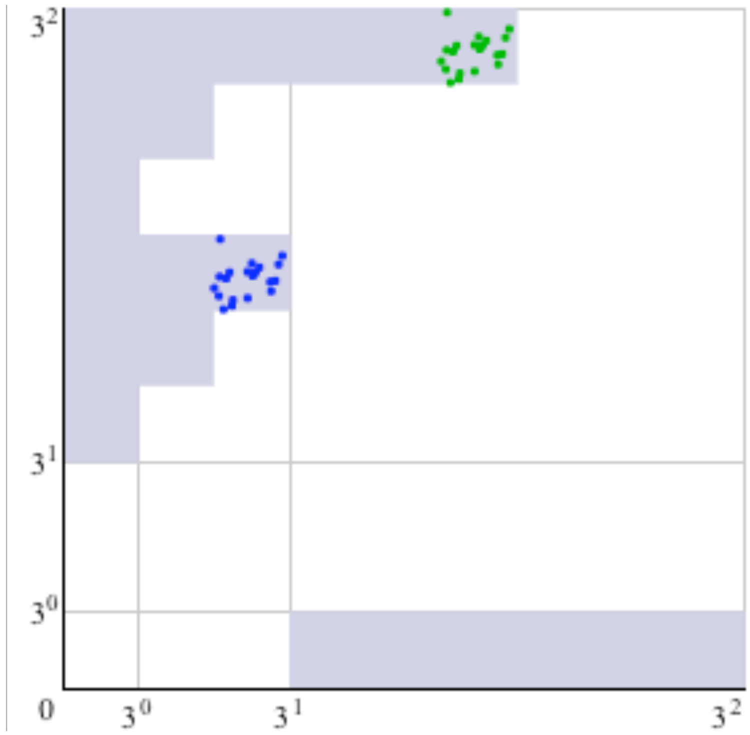

FIGURE 6. Squares Map to Squares, $\mathbb{Q}_3$ out to 3^2

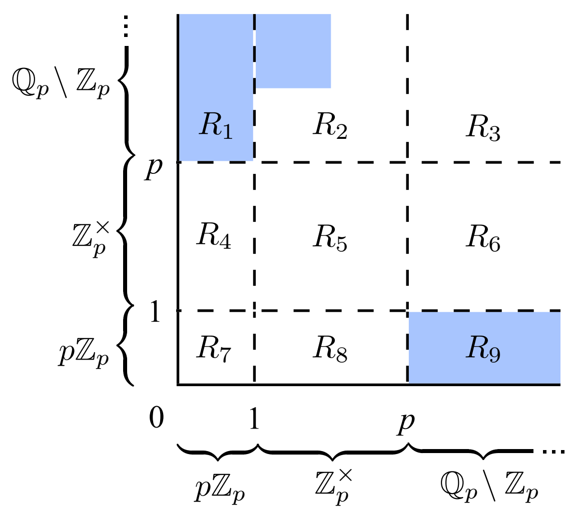

FIGURE 7. Division of the First Quadrant

6.2. Moving Right

For R_4, $x \in p\mathbb{Z}_p$ and $y \in \mathbb{Z}_p^\times$. The next point will be

$$(x',\, y') = \left(\frac{1}{x},\, \frac{1}{y}\right) \in (\mathbb{Q}_p \setminus \mathbb{Z}_p) \times \mathbb{Z}_p^\times = R_6.$$

So the next question is how points in R_6 are mapped by F. Figure 8 shows an example of the iteration of F for a point in R_6 until $F^n(x, y) \in A$.

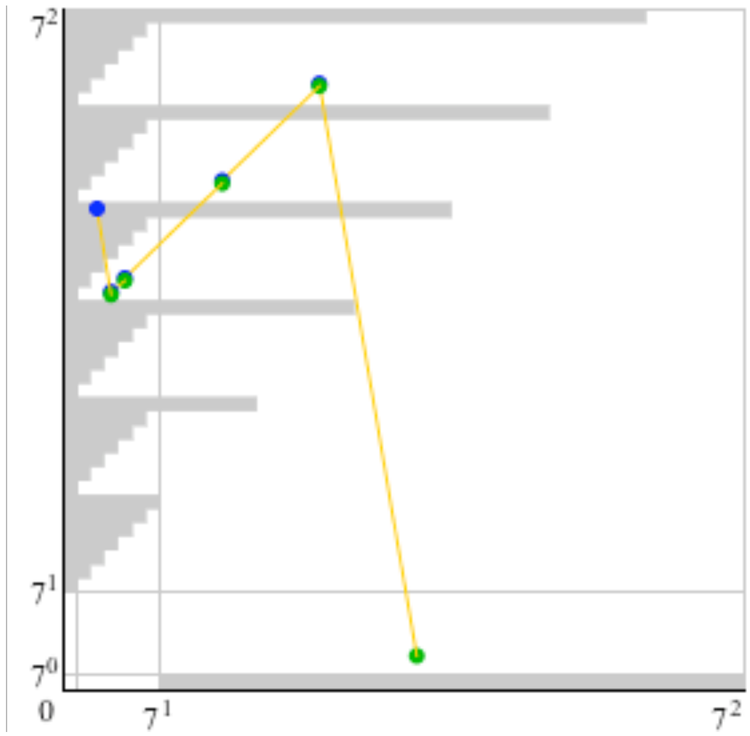

FIGURE 8. Iteration for a Point in R_6, $\mathbb{Q}_7$ out to 7^2

Theorem 3. Given any $(x, y) \in R_6$, $F^n(x, y) \in R_1$ for some $n > 0$.

This will actually be proven by demonstrating an even stronger statement:

Claim. Given $(x, y) \in R_6$, not only is $F^N(x, y) \in R_1$ for some $N > 0$, but $y^{(n)} \notin \mathbb{Z}_p$ for any $n < N$. In other words, the image $\varphi(F^n(x, y))$ hits the leftmost column before passing below the line $\varphi(y) = p$.

Proof: For $(x, y) \in R_6$, $|x|_p > 1$. Because p-adic norms can only be zero or powers of p, $|x|_p \geq p$. By Part (i) of Theorem 1, $\Delta x = \varphi(x') - \varphi(x) = -|x|_p$, so $\Delta x \leq -p$.

By Parts (ii) and (iii), either $\Delta y = \Delta x$ (moving down) or $\Delta y < -\Delta x$ (moving up). If the former is true,

$$\begin{aligned} \varphi(y') - \varphi(y) &\leq -p \\ \varphi(y') &\leq \varphi(y) - p. \end{aligned}$$

But $\varphi(y) < p$, so this would mean that $\varphi(y') < 0$, which is impossible. Thus $\varphi(x, y)$ must move up, and it will do so along a line whose slop is less than -1. Therefore, $\varphi(x^{(n)}, y^{(n)})$

would have to move back down along a line of slope greater than $+1$ in order for $y^{(N)}$ to be less than p for some N. Part (ii) of Theorem 1 states that so long as F operates by subtraction, $\varphi(x^{(n)}, y^{(n)})$ can only move downwards along a line whose slope is exactly 1, so $\varphi(y^{(n)})$ cannot be less than p for any n unless F operates by inversion instead of subtraction. This would require $F^n(x, y)$ to be in R_1, R_4, or R_7, but only R_1 can be reached without $\varphi(y^{(n)})$ being less than p. So $F^n(x, y) \in R_1$ for some n. $\square$

6.3. The $y = x$ Exception

We still have to deal with how points in R_7 end up in the attractor region. F maps $(x, y) \in R_7 = (p\mathbb{Z}_p)^2$ to $(x', y') \in R_3 = (\mathbb{Q}_p \setminus \mathbb{Z}_p)^2$ by inversion. If $(x^{(n)}, y^{(n)})$ ends up in R_1 or R_4 from there, we already know that $F^N(x, y)$ will be in A for some N. So what about if the points ends up back in R_3?

Theorem 4. If F maps $(x, y) \in R_3 \mapsto (x', y') \in R_7$, then either $F^N(x, y) \in R_1 \cup R_4$ for some $N > 0$ or $y = x$. In other words, $(x^{(n)}, y^{(n)})$ cannot go back and forth between R_3 and R_7 indefinitely unless $y = x$.

Proof: This is proven by showing that for any point $(x, y) \in R_7$ for which $y \neq x$, the distance between x and y increases as a result of the eventual inversion. This is in fact true for any normed field, as described generally by the following lemma.

Lemma. Let X be a field and let $d : X \to \mathbb{R}^+$ be a distance. For $x, e \in X \setminus \{0\}$ such that $|x| < 1$ and $|x + e| < 1$, then

$$d\left(\frac{1}{x}, \frac{1}{x+e}\right) > d(x, x+e).$$

Proof of Lemma: First, note that the original distance $d(x, y) = d(x, x + e) = |e|$. Now describe the distance of the inverses in terms of the original distance.

$$\begin{aligned} d\left(\frac{1}{x}, \frac{1}{x+e}\right) &= \left|\frac{1}{x} - \frac{1}{x+e}\right| \\ &= \left|\frac{x+e}{x(x+e)} - \frac{x}{x(x+e)}\right| \\ &= \left|\frac{x+e-x}{x(x+e)}\right| \\ &= \frac{|e|}{|x||x+e|} \end{aligned}$$

$$
\begin{aligned}
d\left(\frac{1}{x},\, \frac{1}{x+e}\right) &= \frac{d(x,\, x+e)}{|x||x+e|} \\
&= \alpha \cdot d(x,\, x+e),\ \text{where } \alpha = \frac{1}{|x||x+e|}
\end{aligned}
$$

Since $|x| < 1$ and $|x+e| < 1$ by assumption, α must be greater than 1, and $d\left(\frac{1}{x},\, \frac{1}{x+e}\right)$ must be greater than $d(x,\, x+e)$. $\square$

To apply this to our situation, let $y = x + e$. Since $e \neq 0$, $y \neq x$. Also, $|x| < 1$ and $|x + e| < 1$ means that $|x|_p < 1$ and $|y|_p < 1$, which puts $(x,\, y)$ in R_7. The equation

$$
d\left(\frac{1}{x},\, \frac{1}{x+e}\right) > d(x,\, x+e) \text{ becomes } d\left(\frac{1}{x},\, \frac{1}{y}\right) > d(x,\, y),
$$

so for any $(x,\, y) \in R_7$ for which $x \neq y$, $F^n(x,\, y)$ will diverge from $y = x$, eventually resulting in $F^N(x,\, y) \in R_1 \cup R_4$ for some N. $\square$

7. Staying in the Attractor Region

7.1. From the Left to the Bottom

Points in R_1, the leftmost column of A, are mapped by F to R_9, the bottom strip in A. $R_1 = p\mathbb{Z}_p \times (\mathbb{Q}_p \setminus \mathbb{Z}_p)$, so for $(x,\, y) \in R_1$,

$$
(x',\, y') = \left(\frac{1}{x},\, \frac{1}{y}\right) \in (\mathbb{Q}_p \setminus \mathbb{Z}_p) \times p\mathbb{Z}_p = R_9.
$$

7.2. From the Bottom to the Top

Theorem 5. Points in the bottom strip (R_9) are mapped by F to bars whose images are just below p^n. Specifically, if $y \in p\mathbb{Z}_p$ and $\varphi(x) \in [p^{n-1},\, p^n)$, then $\varphi(y') \in [p^n - 1,\, p^n)$ and $\varphi(x') = \varphi(x) - p^{n-1}$.

Figures 9 and 10 show examples of this mapping. In these figures, green points are connected to their images (blue points) by orange lines. Because each unit square maps to a unit square (Theorem 2), the connections are shown for just one point in each unit square.

Proof:

If $\varphi(x) \in [p^{n-1},\, p^n) = [1\underbrace{0\ldots0}_{n-1},\, 1\underbrace{00\ldots0}_{n})$, then $\varphi(x) = a_{n-1}\ldots a_1 a_0$, $a_{n-1} \neq 0$.

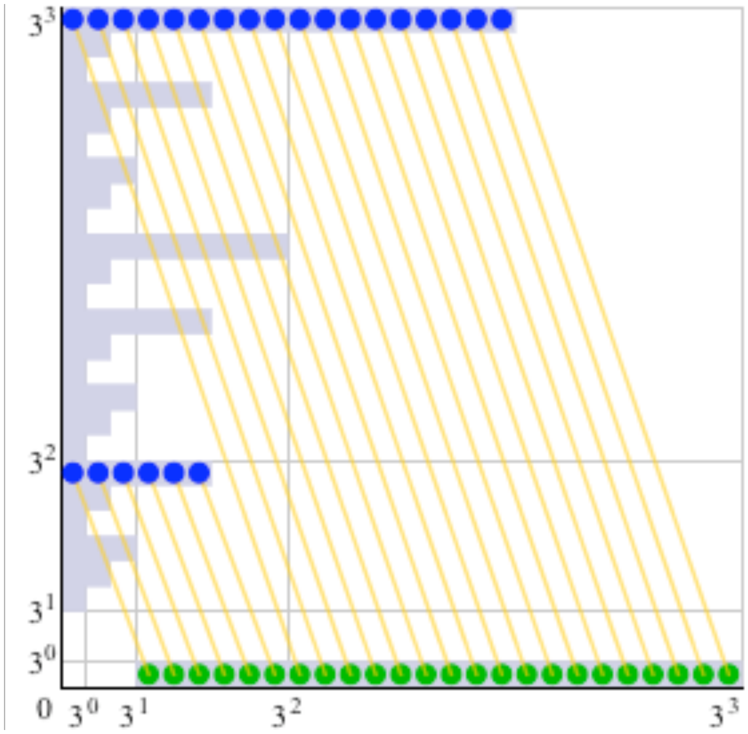

FIGURE 9. Bottom Strip Maps to Top Strips, $\mathbb{Q}_3$ out to 3^3

Let $k = \varphi(x) - p^{n-1} = (a_{n-1} - 1)a_{n-2} \ldots a_1 a_0$. Let $\varphi(y) = 0.b_1 b_2 \ldots$. To find y',

$$\begin{array}{rl} x = & \qquad a_0.a_1 \ldots a_{n-1} \\ y = & \ldots b_1\, 0\, . \\ \text{Add} & \ldots q\ q\, .\, q \ldots q \\ y' = & \ldots c_1\, q\, .\, q \ldots q \end{array}$$

So $\varphi(y') = \underbrace{q \ldots qq}_{n}.c_1 \ldots \in [p^n - 1,\, p^n)$.

To find x',

$$\begin{array}{rll} x = & a_0.a_1 \ldots a_{n-2}\, a_{n-1}, & \quad a_{n-1} \neq 0 \\ \text{Subtract} & 0\, .\, 0 \ldots \quad 0 \qquad 1 & \\ x' = & a_0.a_1 \ldots a_{n-2}\, (a_{n-1} - 1) & \end{array}$$

So $\varphi(x') = (a_{n-1} - 1)a_{n-2} \ldots a_1 a_0 = k = \varphi(x) - p^{n-1}$. $\square$

7.3. From the Top to the Left

Theorem 6. If $(x,\, y) \in A^*$, $x \notin p\mathbb{Z}_p$, then $\varphi(x') < \varphi(x)$ and $\varphi(y') < \varphi(y)$. In other words, F moves $\varphi(x,\, y)$ down and to the left.

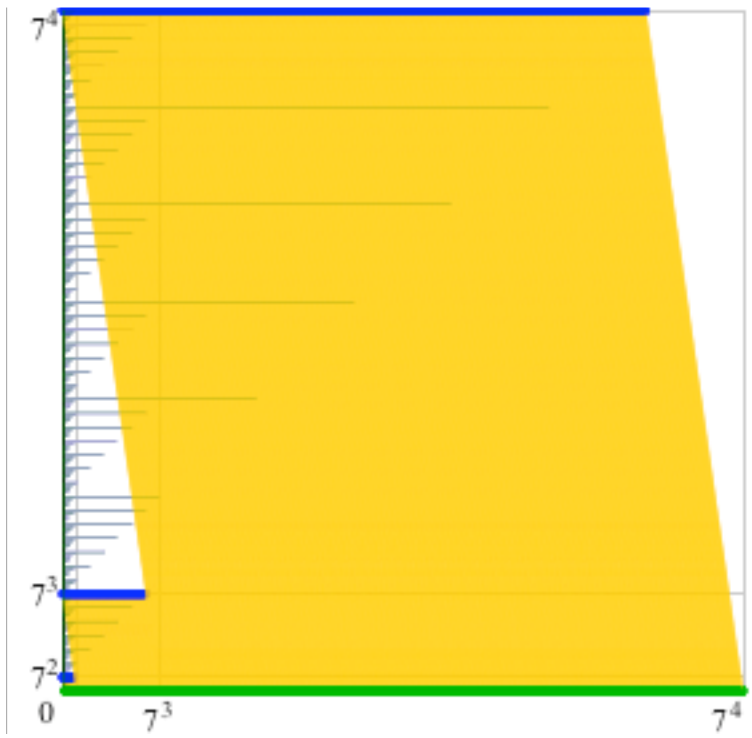

FIGURE 10. Bottom Strip Maps to Top Strips, $\mathbb{Q}_7$ out to 7^4

Proof: As the proof of Part (ii) of Theorem 1 showed, mapping down and to the left occurs when $ord_p x \leq ord_p y$. Figure 11 shows several examples of this mapping (green points map to their connected blue points).

Let $k = \lfloor \varphi(y) \rfloor + 1$. Because the bars in A^* don't start until above $\varphi(y) = p$, $k \geq p + 1$. Using $\ell(k)$ as defined on page 8,

$$p \leq k - 1 < \varphi(y) < k$$
$$1 < \varphi(x) < \ell(k)$$

If $\varphi(y) = b_n \dots b_1 b_0 . b_{-1} \dots$, the fact that $\varphi(y) \geq p$ means that $b_i \neq 0$ for some $i > 0$. Then $y = \dots b_{-1} b_0 . b_1 \dots b_n$, and $ord_p y > 0$.

If $k \not\equiv 0 \pmod{p}$, $\ell(k) = k \bmod p$.

$$1 < \varphi(x) < k \bmod p < p$$
$$\varphi(x) = a_0 . a_{-1} \dots, \quad a_0 \neq 0$$
$$x = \dots a_{-1} a_0 .$$
$$ord_p x = 0.$$

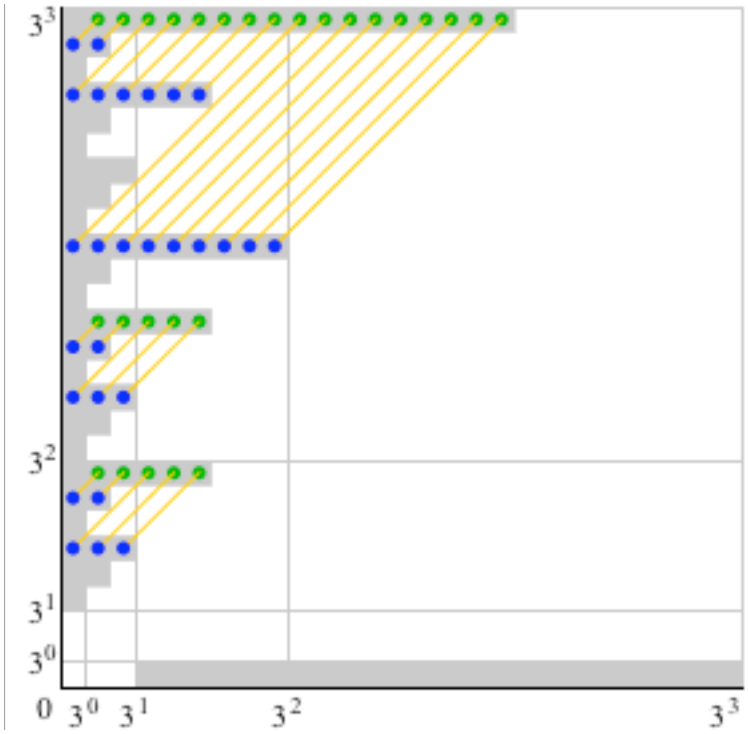

FIGURE 11. Down and to the Left, $\mathbb{Q}_3$ out to 3^3

If $k \equiv 0 \pmod p$, then $k = cp^m$ for some $c > 0$ relatively prime to p and some exponent $m \geq 0$. Via its recursive formula,

$$\ell(k) = p\ell(cp^{m-1}) = p^2\ell(cp^{m-2}) = \cdots = p^m\ell(c).$$

This formula holds for all $m \geq 0$. Since c is coprime to p, $\ell(c)$ is $c - 1$ if $c \leq p$ or $c \bmod p$ if $c > p$. Either way,

$$\begin{aligned} 1 < \varphi(x) < \ell(k) &= p^m\ell(c) \\ \varphi(x) &= a_m \ldots a_1 a_0, \quad a_m < \ell(c) \\ x &= a_0.a_1 \ldots a_m \\ ord_p x &\leq 0. \end{aligned}$$

Whether k is divisible by p or not, $ord_p x \leq 0$ and $ord_p y > 0$, so by Part (ii) of Theorem 1, $\Delta x = \Delta y < 0$, which is movement down and to the left. $\square$

8. Conclusion and Future Work

This thesis does conclusively prove the existence of an attractor region for the given p-adic natural extension map. This region is describable in purely p-adic terms, but that description is motivated by its image under the digit-reversing map φ.

There are three apparent areas of the natural extension map F which are not fully explored in this thesis. The set of all initial points for which F^n is eventually undefined for some n (that is, when $x^{(n-1)} = 0$) is not fully described here. Page 4 gives one form of x-value for which $x^{(m)} = 0$ for some m, but there may or may not be other x-values with this property. Additionally, the set of all initial points for which the behavior of F is eventually periodic is not described here, nor is the exact motion of points with periodic orbits.

The digit manipulation techniques used here may be helpful in answering the above questions. The computer storage, arithmetic, and graphing techniques (the code for which is partially shown in the Appendix) may have applications for any number of further problems involving p-adic analysis.

References

[1] Svetlana Katok, *p-adic Analysis Compared with Real*, AMS Student Mathematical Library **37** (2007).

[2] Svetlana Katok and Ilie Ugarcovici, *Structure of Attractors for (a, b)-continued Fraction Transformations*, Journal of Modern Dynamics **4** (2010).

[3] Matthew Moore, *p-adic Continued Fractions*, Undergraduate Research Assistantship, Univ. of Arizona, May 4, 2006.

Appendix

The following Java classes are all written by Adam Zydney for the purpose of experimental data collection involving p-adic numbers.

A zero value is represented by a `PadicZero` object. p-adic integers are stored as a finite sequence of digits in the `PadicInteger` class. Every other p-adic number is stored using the `PadicNumber` class as a product $u \times p^m$, where u is a p-adic integer and m is an integer (positive or negative). All three of these classes extend the base class `Padic`, which defines some common methods and leaves others as abstract.

All `Padic` objects also internally store the value of p, and attempts to do arithmetic operations on numbers with two different values for p will result in a runtime error. Because only a finite number of digits are used for any number, a `Padic` object can only *perfectly* represent a small subset of p-adic numbers.

Note that this appendix does not include various other classes and execution code used in the creation of this thesis (code to, for example, create a graph with prime power axis markings, draw the attractor region, plot points, or iterate the natural extension map). While essential to the development of this thesis, such code would be less useful for others wanting to do computer simulations with p-adic numbers.

```
public abstract class Padic
{
    public abstract Padic plus(Padic other);
    public Padic minus(Padic other) { return this.plus(other.negative()); }
    public abstract Padic times(Padic other);
    public abstract Padic timesPrimeToPower(int m);
    public Padic dividedBy(Padic other) { return this.times(other.reciprocal()); }

    public abstract Padic negative();
    public abstract Padic reciprocal();

    public abstract int getOrder();
    public abstract int getPrime();
    public double getNorm() { return Math.pow(getPrime(), -getOrder()); }

    public double distanceTo(Padic other) { return this.minus(other).getNorm(); }
    public static double distance(Padic a, Padic b) { return a.distanceTo(b); }

    public abstract double flipToReal();

    protected abstract PadicUnit asUnit();

    // If possible, returns this number as a PadicNumber. For a PadicZero, returns null.
    protected abstract PadicNumber asNumber();

    protected abstract boolean isZero();

    // Returns the digits of this number's p-adic expansion in String format.
    // This string may contain a radix point.
    protected abstract String digitString();

    // Returns a representation of this number's p-adic expansion. An attempt is made to
    // replacing most of the periodic segment of an expansion with an ellipsis (...).
    public String toString()
    {
        String str = digitString();
```

```
    if (str.charAt(0) == '.') return "0" + str;
    if (!str.contains(".")) str += ".";

    for (int L = 1; L < str.length()/3; L++)
    {
        // L is period length
        String period = str.substring(0, L);
        String triple = period + period + period + period + period;
        if (str.startsWith(triple))
        {
            str = str.substring(5*L);
            while (str.startsWith(period))
                str = str.substring(L);
            //int reps = Math.max(2*L, 3);
            //return "..." + repeat(period, reps/L) + str;
            return "(" + period + ")" + str;
        }
    }
    return str;
}

public static Padic convert(int n, int p) { return build(Integer.toString(n, p), p); }

public static Padic convertRational(int a, int b, int p)
{
    Padic numerator = Padic.convert(a, p);
    Padic denominator = Padic.convert(b, p);
    return numerator.dividedBy(denominator);
}

public static Padic convertFromFlip(int n, int p)
{
    String s = Integer.toString(n, p);
    s = (new StringBuilder(s)).reverse().toString();
    return build(s, p).timesPrimeToPower( 1 - s.length() );
}

public static Padic convertFromFlipDual(int n, int p) { return null; }

protected static Padic create(char[] digits, int p) { return build(new String(digits), p); }

// Creates a p-adic number from a given finite p-adic expansion. This expansion may contain
// a radix point and may begin with a negative sign.
public static Padic build(String finite, int p)
{
    if (finite.charAt(0) == '-')
    {
        Padic x = buildPeriodic("0", finite.substring(1), p);
        return x.negative();
    }
    else
        return buildPeriodic("0", finite, p);
}
```

```
// Creates a p-adic number from a given periodic p-adic expansion. This expansion may contain
// a radix point in the non-repeating section, but not a negative sign. The p-adic number
// returned has the period appended to the right of the finite p-adic expansion as many times
// as allowed by the precision of this data type. Because Padic objects use a finite number
// of digits internally, the resulting number is only an approximation of the true periodic
// p-adic number.
public static Padic buildPeriodic(String period, String finite, int p)
{
    if (finite.contains("."))
    {
        while (finite.charAt(finite.length()-1) == '0')
            finite = finite.substring(0, finite.length()-1);

        int place = finite.indexOf(".");
        int k = finite.length() - place - 1;

        String f2 = finite.substring(0, place) + finite.substring(place+1);
        return new PadicNumber(new PadicUnit(period, f2, p), -k);
    }
    else
    {
        finite = period + finite;

        int m;
        for (m = 0; finite.charAt(finite.length()-1) == '0' && finite.length()>1; m++)
            finite = finite.substring(0, finite.length()-1);

        try {
            if (Integer.parseInt(period, p) == 0 && Integer.parseInt(finite, p) == 0)
                return new PadicZero(p);
        } catch (Exception ex) { }

        PadicUnit u = new PadicUnit(period, finite, p);
        return (m == 0) ? u : new PadicNumber(u, m);
    }
}

protected static String repeat(String s, int n)
{
    StringBuilder sb = new StringBuilder();
    for (int i = 0; i < n; i++)
        sb.append(s);
    return sb.toString();
}

protected static int parseDigit(char digit, int p)
{
    int d;
    try { d = Integer.parseInt(Character.toString(digit), p); }
    catch (NumberFormatException ex) { d = 0; }
    return d;
}
```

```
    public static void setUnitPrecision(int length) { PadicUnit.LENGTH = length; }

    public static int getUnitPrecision() { return PadicUnit.LENGTH; }

    public abstract boolean equals(Padic other);
}

public class PadicZero extends Padic
{
    private int prime;

    public PadicZero(int p) { prime = p; }

    public Padic plus(Padic other) { return other; }
    public Padic times(Padic other) { return this; }
    public Padic timesPrimeToPower(int m) { return this; }
    public Padic negative() { return this; }

    public Padic reciprocal() { throw new ArithmeticException("/ by zero"); }

    public int getOrder() { return Integer.MAX_VALUE; }

    public int getPrime() { return prime; }

    public double flipToReal() { return 0; }

    protected PadicUnit asUnit() { return null; }

    protected PadicNumber asNumber() { return null; }

    protected boolean isZero() { return true; }

    protected String digitString() { return "0"; }

    public String toString() { return "0"; }

    public boolean equals(Padic other) { return other.isZero(); }
}

public class PadicUnit extends Padic
{
    protected char[] digits;
    protected int prime;

    protected static int LENGTH = 300;

    protected PadicUnit(char ch, int p) { this(Character.toString(ch), p);   }

    public PadicUnit(char[] ds, int p) { digits = ds; prime = p; }

    protected PadicUnit(int value, int p) { this("0", Integer.toString(value, p), p); }

    protected PadicUnit(String finite, int p) { this("0", finite, p); }
```

```
protected PadicUnit(String period, String finite, int p)
{
    prime = p;
    digits = new char[LENGTH];
    int index = LENGTH - 1;
    for (int i = finite.length()-1; i >= 0 && index >= 0; index--)
        digits[index] = finite.charAt(i--);
    for (int i = period.length()-1; index >= 0; index--)
    {
        digits[index] = period.charAt(i);
        if (i == 0)
            i = period.length()-1;
        else
            i--;
    }
}

public Padic plus(Padic other)
{
    if (other.isZero())
        return this;
    if (other.asUnit() == null)
        return other.asNumber().plus(this);

    // Now adding two p-adic units. The result may or may not be a p-adic unit.

    char[] sum = addByDigits(this.digits, other.asUnit().digits, prime);
    return Padic.create(sum, prime);
}

protected static char[] addByDigits(char[] adigits, char[] bdigits, int p)
{
    char[] sum = new char[LENGTH];
    char carry = '0';
    for (int i = LENGTH-1; i >= 0; i--)
    {
        String s = addDigits(adigits[i], bdigits[i], carry, p);
        carry = s.charAt(0);
        sum[i] = s.charAt(1);
    }
    return sum;
}

private static String addDigits(char d1, char d2, char d3, int p)
{
    int a = Padic.parseDigit(d1, p);
    int b = Padic.parseDigit(d2, p);
    int c = Padic.parseDigit(d3, p);
    String sum = Integer.toString(a+b+c, p);
    return (sum.length() == 1) ? "0" + sum : sum;
}
```

```
public Padic times(Padic other)
{
    if (other.isZero())
        return other;

    PadicUnit u = other.asUnit();
    if (u != null)
        return timesUnit(u);
    else
        return asNumber().timesNumber(other.asNumber());
}

protected PadicUnit timesUnit(PadicUnit other)
{
    Padic temp = new PadicZero(prime);
    for (int i = 0; i < LENGTH; i++)
    {
        char digit = other.digits[LENGTH-1 - i];
        Padic partialProduct = this.timesDigit(digit);
        temp = temp.plus(partialProduct.timesPrimeToPower(i));
    }
    return temp.asUnit();
}


private Padic timesDigit(char digit)
{
    if (digit == '0') return new PadicZero(prime);
    if (digit == '1') return this;

    char[] sum = new char[LENGTH];
    char[] carry = new char[LENGTH];
    int d = Padic.parseDigit(digit, prime);

    for (int i = LENGTH - 1; i >= 0; i--)
    {
        int a = Integer.parseInt(Character.toString(digits[i]), prime);
        int s = (d*a) % prime;
        int c = (d*a)/prime;
        sum[i] = Integer.toString(s, prime).charAt(0);
        carry[i] = Integer.toString(c, prime).charAt(0);
    }

    Padic s = Padic.create(sum, prime);
    Padic c = Padic.create(carry, prime).timesPrimeToPower(1);
    return s.plus(c);
}

public Padic timesPrimeToPower(int m) { return new PadicNumber(this, m); }

public Padic negative()
{
    char[] flipped = new char[LENGTH];
```

```
    for (int i = 0; i < LENGTH; i++)
    {
        String ch = Character.toString(digits[i]);
        int f = (prime-1) - Integer.parseInt(ch, prime);
        flipped[i] = Integer.toString(f, prime).charAt(0);
    }
    PadicUnit u = new PadicUnit(flipped, prime);
    return u.plus(new PadicUnit("1", prime));
}

public Padic reciprocal()
{
    char[] rec = new char[LENGTH];

    char a = digits[LENGTH - 1];
    Padic d = new PadicUnit('1', prime);

    for (int i = 0; i < LENGTH;)
    {
        char c = modularSolve(a, d.asUnit());
        PadicUnit cx = this.timesUnit(new PadicUnit(c, prime));
        d = d.minus(cx);
        rec[LENGTH-1 - i] = c;

        if (d.isZero())
        {
            for(int j = 0; j < LENGTH-2-i; j++)
                rec[j] = '0';
            break;
        }

        int m = d.getOrder();
        d = d.asNumber().unit;
        for (int j = i+1; j < i+m && j < LENGTH; j++)
            rec[LENGTH-1 - j] = '0';
        i += m;
    }

    char[] recFixed = ATTEMPT_FIX(rec, prime);
    return new PadicUnit(recFixed, prime);
}

private static char[] ATTEMPT_FIX(char[] rec, int p)
{
    int cut = 10;
    int reps = 4;
    String str = (new String(rec)).substring(cut);

    for (int L = cut/reps+1; L < str.length()/reps; L++)
    {
        // L is period length
        String period = str.substring(0, L);
        String repeated = repeat(period, reps);
```

```
        if (str.startsWith(repeated))
        {
            String fix = repeated.substring(repeated.length() - cut);
            return (fix + str).toCharArray();
        }
    }
    return rec;
}

private static char modularSolve(char a, PadicUnit b)
{   return modularSolve(a, b.digits[LENGTH-1], b.getPrime());   }

// Solves a*c = b (mod p) for c.
private static char modularSolve(char a, char b, int p)
{
    int a0 = Integer.parseInt(Character.toString(a), p);
    int bi = Integer.parseInt(Character.toString(b), p) % p;
    for (int c = 0; c < p; c++)
        if (a0*c % p == bi)
            return Integer.toString(c, p).charAt(0);
    throw new ArithmeticException("Unsolvable modular equation: " + a +
            " * c = " + b + " (mod " + p +") for c");
}

public int getOrder() { return 0; }
public int getPrime() { return prime; }

public double flipToReal()
{
    double r = 0;
    for (int i = 0; i < LENGTH; i++)
    {
        char digit = digits[LENGTH-1 - i];
        int d = 0;
        try { d = Integer.parseInt(Character.toString(digit), prime); }
        catch (Exception ex) { }
        r += d*Math.pow(prime, -i);
    }
    return r;
}

protected PadicUnit asUnit() { return this; }
protected PadicNumber asNumber() { return new PadicNumber(this, 0); }
protected boolean isZero() { return false; }

// Returns u * p^L + v
protected static PadicUnit sumToUnit(PadicUnit u, int L, PadicUnit v)
{
    String upL = (new String(u.digits)) + repeat("0", L);
    upL = upL.substring(upL.length() - LENGTH);
    char[] sum = addByDigits(upL.toCharArray(), v.digits, u.prime);
    return new PadicUnit(sum, u.prime);
}
```

```
    protected String digitString()
    {
        String str = new String(digits);
        while (str.charAt(0) == '0')
            str = str.substring(1);
        return str;
    }

    public boolean equals(Padic other)
    {
        PadicUnit u = other.asUnit();
        return u != null && u.digits == digits;
    }
}

public class PadicNumber extends Padic
{
    protected PadicUnit unit;
    protected int order;

    public PadicNumber(PadicUnit u, int m) { unit = u; order = m; }

    public Padic plus(Padic other)
    {   return other.isZero() ? this : this.plusNumber(other.asNumber());   }

    public Padic plusNumber(PadicNumber other)
    {
        int m = getOrder(), n = other.getOrder();

        if (m < n)
            return other.plus(this);
        else if (m == n)
        {
            Padic s = unit.plus(other.unit);
            if (s.isZero())
                return s;
            PadicNumber t = s.asNumber();
            // s = x+y = w * p^k
            if (t.getOrder() == -m)
                return t.unit;
            else
                return new PadicNumber(t.unit, t.getOrder()+m);
        }
        else
        {
            //        u = unit * p^(m-n) + other.unit
            PadicUnit u = PadicUnit.sumToUnit(unit, m-n, other.unit);
            return new PadicNumber(u.asUnit(), n);
        }
    }

    public Padic times(Padic other)
    {   return other.isZero() ? other : this.timesNumber(other.asNumber());   }
```

```
public Padic timesNumber(PadicNumber other)
{
    PadicUnit u = this.unit.timesUnit(other.unit);
    int m = this.getOrder() + other.getOrder();
    return (m == 0) ? u : new PadicNumber(u, m);
}

public Padic timesPrimeToPower(int m)
{   return order+m == 0 ? unit : new PadicNumber(unit, order+m);   }

public Padic negative()
{   return new PadicNumber(unit.negative().asUnit(), order);   }

public Padic reciprocal()
{   return new PadicNumber(unit.reciprocal().asUnit(), -order);   }

public int getOrder() { return order; }

public int getPrime() { return unit.getPrime(); }

public double flipToReal()
{   return unit.flipToReal() * Math.pow(unit.getPrime(), -order);   }

protected PadicUnit asUnit() { return (order == 0) ? unit : null; }

protected PadicNumber asNumber() { return this; }

protected boolean isZero() { return false; }

protected String digitString()
{
    String str = unit.digitString();

    if (order == 0)
        return str;
    else if (order > 0)
        return str + repeat("0", order);
    else
    {
        int n = -order;
        if (n < str.length())
        {
            String iPart = str.substring(0, str.length() - n);
            String dPart = str.substring(str.length() - n);
            return iPart + "." + dPart;
        }
        else
        {
            int extraZeros = n - str.length(); // order is negative
            return "." + repeat("0", extraZeros) + str;
        }
    }
}
```

```
    public boolean equals(Padic other)
    {
        PadicNumber n = other.asNumber();
        return n != null && order == n.order && unit.equals(n.unit);
    }
}
```

As a small example of how this code can be used, the following lines print out the first 9 iterations of F starting with the initial 5-adic point $(\dots 4440, 0.1)$.

```
int prime = 5;
Padic x = Padic.convert(-prime, prime); // -p = -10 base p = ...qqq0.
Padic y = Padic.build("0.1", prime);

for (int i = 0; i < 2*prime; i++)
{
    System.out.println("F^" + i + "(x, y) = ( " + x + ", " + y + " )");

    Padic[] pt = F(x, y); // The static method F(Padic x, Padic y) is defined elsewhere.
    x = pt[0]; y = pt[1];
}
```

The output of this code shown below (with spacing altered to align radix points) gives an example of a periodic orbit: $F^9(x, y) = (x, y)$.

```
F^0(x, y) = ( (4)0.,   0.1 )
F^1(x, y) = (  (4).4, 10.  )
F^2(x, y) = (  (4).3,  4.4 )
F^3(x, y) = (  (4).2,  4.3 )
F^4(x, y) = (  (4).1,  4.2 )
F^5(x, y) = (  (4).,   4.1 )
F^6(x, y) = ( (4)3.,   3.1 )
F^7(x, y) = ( (4)2.,   2.1 )
F^8(x, y) = ( (4)1.,   1.1 )
F^9(x, y) = ( (4)0.,   0.1 )
```

Academic Vita

Adam Zydney

adam.zydney@gmail.com

Education

Pennsylvania State University, May 2011.
Bachelor of Science in Mathematics,
General Option, Computer Science Application
Thesis Title: Structure of Attractor Region for Natural Extension Map Associated with p-adic Continued Fractions
Thesis Supervisor: Svetlana Katok

Experience

State College Area School District Elementary Math Curriculum Selection Committee.
Member, Summer 2010 – Fall 2010.

John Hopkins Center for Talented Youth.
Teaching Assistant, Cryptology, Summer 2010.

Pennsylvania State University.
Grader, Honors Discrete Math, Honors Real Analysis, Fall 2009 – Fall 2010.

Mathematics Advanced Study Semester (MASS) Program.
Fellow, Fall 2009.

Penn State Dance Marathon.
Lead Database Developer, 2009 – 2011.

Penn State Mathematics Research Experience for Undergraduates.
Researcher, Summer 2009.

Penn State Student Support Services Program.
Tutor, Fall 2008.

Penn State Quiz Bowl Student Organization.
Founder and President, 2008 – 2011.

State College Area High School.
Co-author, Editor, Vector Calculus Textbook, Summer 2007.

Awards

Diamond of Wisdom Award, 2011
Outstanding MASS Research Projects, 2009
Dean's List, all semesters
Schreyer Honors College Scholar

Skills

LaTeX, Mathematica, MATLAB, PHP, MySQL,
HTML, Java, JavaScript, CSS, C++