Symbolic dynamics, the spider algorithm and finding certain real zeros of polynomials of high degree

Implementing non−linear dynamical systems theory by Mathematica.

T. M. Jonassen

Dept. of Computer Science, Faculty of Engineering, Oslo University College, Norway
torejo@hio.no

Abstract: We consider the kneading theory developed by Milnor and Thurston[MT] and the spider algorithm developed by Hubbard and Schleider[HS] to study certain properties of families of discrete dynamical systems, and how to implement this theory with Mathematica.We are concerned by identifying certain combinatorial and numerical properties of periodic critical orbits in one dimentional discrete dynamical systems generated by second order real polynomial maps by iteration, an important class of unimodal systems.

■ Introduction

A mapping f of an interval $I \subset \mathbb{R}$ given by $f : I \longrightarrow I$ give rise to a dynamical system on I by iteration. An orbit of a point $x_0 \in I$ is a sequence of points $\{x_i\}_{i\geq 0} \subset I$ given by $x_{i+1} = f(x_i)$. We use the standard notation $x_n = f^n(x_0)$, so f^n means the n times composition of f with itself. An orbit is called periodic of primitive period n if $x_{i+n} = x_i$ for some $n \geq 1$, but $x_{i+k} \neq x_i$ for $k = 1, 2, \ldots, n-1$. A **critical** periodic orbit is a periodic orbit containing the critical point of f. A critical periodic orbit is also called a **superstable** periodic orbit. A point is called critical if the derivative of the map is zero at the point.

Symbolic dynamics is a technique in the study of dynamical systems, in its simplest form converting orbits $\{x_i\}_{i\geq 0}$ into sequences of symbols from an alphabet $\mathbb{A} = \{\alpha_1, \alpha_2 \ldots, \alpha_k\}$, and in our unimodal case, the alphabet consists of three symbols so $\mathbb{A} = \{L, C, R\}$. The iteration process (the dynamical system) translates into a simple operator on the symbol space consisting of infinite words from $\mathbb{A}$. We will here only look at some elementary symbolic dynamics for simple one dimensional systems $f : I \longrightarrow I$ where f is a smooth function of the interval $I \subset \mathbb{R}$ such that f has only one critical point in I, and we may write $I = I_L \bigcup I_R$ such that f is decreasing on I_L and increasing on I_R. Let $I_L = [l, c)$ and $I_R = (c, r]$. The **address** of a point $x \in I$, denoted by $a(x)$ is given by $a(x) = L$ if $x \in I_L$, $a(x) = C$ if $x = c$ and $a(x) = R$ if $x \in I_R$. Since $x \in I \Rightarrow f(x) \in I$ by assumption, the orbit of a point $x_0 \in I$ is contained in I, and we may assign an infinite sequence of symbols $A(x_0) = \{s_i\}_{i\geq 0}$ to the orbit according to the rule $s_i = a(f^i(x_0))$. The sequence $A(x_0)$ is also called the **itinerary** of the point x_0. There is a natural mapping on the space of symbol sequences compatible with the dynamics on I, called the **shift map**, denoted by σ. We define $\sigma(\{s_i\}_{i\geq 0}) = \{s_{i+1}\}_{i\geq 0}$. Let $A(x_0) = \{s_i\}_{i\geq 0}$, then clearly $\sigma(A(x_0)) = A(f(x_0))$. Hence the action $x \mapsto f(x)$ corresponds to shifting the sequence of symbols one place to the left and forget the first symbol in the sequence $A(x)$. The critical point $c \in I$ plays a special role in the dynamics of f. The dynamics of the critical orbit is given by the symbol sequence $A(c)$. This sequence is called the **kneading sequence** of f and is denoted by $K(f) = A(c)$. We will here be concerned with superstable periodic orbits. This means that the kneading sequence of f is periodic. A periodic kneading sequence is written as the periodic data with an overbar as in the sequence $K(f) = \overline{\text{LRLLLLC}}$ of period 7. This sequence corresponds to an existing periodic orbit, while there is no orbit for the dynamical system corresponding to the sequence $\overline{\text{LRLLLRC}}$. A main problem is to decide if a given periodic symbol sequence corresponds to an orbit in the dynamical system $f : I \longrightarrow I$. A symbol sequence with a corresponding orbit in the dynamical system is called **admissible**. We apply the theory developed by Milnor and Thurston in [MT] and [X] to obtain an algorithm to decide if a given symbol sequence is admissible. This algorithm is based on the fact that the kneading sequence is minimal with respect to the lexicographic order denoted by $\preccurlyeq$ on the symbol space $\Sigma = \mathbb{A}^{\mathbb{N}}$. In particular, if σ is the shift map on the symbol space, then $K \preccurlyeq \sigma^i(K)$, $i = 1, \ldots, n-1$, for any admissible kneading sequence of length n.

The spider algorithm ([HS], [B]) was designed to study certain properties of the Mandelbrot set for families of dynamical systems $g_\alpha : \mathbb{C} \longrightarrow \mathbb{C}$. We apply this algorithm to real unimodal systems generated by second degree polynomials using the admissible kneading sequences. If the periodic kneading sequence is of length n this corresponds to find a certain real solution of a polynomial equation of degree 2^{n-1}. This leads to a numerical method to study of the parameterspace of unimodal systems generated by second degree polynomials. The Sharkovsky theorem ([S],[SMR]) is illustrated (special case "period 3 $\Rightarrow$ chaos", [LY]).

It is easy to see that it is sufficient to study a single representative among the non−degenerate second order polynomials, let $p_\theta(x) = x^2 + \theta$ and let $f_{\alpha,\beta,\gamma}(x) = \alpha x^2 + \beta x + \gamma$ where $\alpha \neq 0$. Then for any such map there is a homeomorphism (in fact of the simple form $h(x) = a x + b$) such that $p_\theta = h \circ f_{\alpha,\beta,\gamma} \circ h^{-1}$ where the quantities a, b and θ are dependent on α, β and γ. Hence $p_{\theta(\alpha,\beta,\gamma)}$ and $f_{\alpha,\beta,\gamma}$ is topological conjugate and have the same dynamics. We will use p_θ as the representative for the second order polynomials in the following.

■ Some simple properties of the family $x \mapsto x^2 + \theta$

We will list some simple properties of the dynamical system $p_\theta : x \mapsto x^2 + \theta$. All properties are easily proved using elementary calculus so the calculations are omitted here. The statements on symbolic dynamics and hyperbolicity can easily be proved using the techniques in [D] or a modification of the arguments in the next section. In the following let $x_l(\theta) = \left(1 - \sqrt{1 - 4\theta}\right)/2$, $x_r(\theta) = \left(1 + \sqrt{1 - 4\theta}\right)/2$ and $I(\theta) = [-x_r(\theta), x_r(\theta)]$ whenever these quantities are real ($\theta \leq \frac{1}{4}$). Clearly $x = 0$ is the only critical point of p_θ.

(1) If $\theta > 1/4$ then $\lim_{n\to\infty} p_\theta^n(x_0) = \infty$ for all $x_0 \in \mathbb{R}$.

(2) If $\theta < 1/4$ then p_θ has two fixed point given by $x_1 = x_l(\theta)$ and $x_2 = x_r(\theta)$.

(3) If $-2 \le \theta \le 1/4$ then the interval $I = I(\theta)$ is invariant under p_θ, that is, $p_\theta(I(\theta)) \subset I(\theta)$.

(4) If $\theta < -2$ then every periodic orbit of p_θ is repelling. p_θ has periodic orbits of every primitive order, but none of these contain the critical point.

(5) If $\theta < -2$ then p_θ has an invariant Cantor set $\Lambda_\theta \subset I(\theta)$ such that the restriction $p_\theta \mid \Lambda_\theta$ is topological conjugate to a one sided shift on two symbols. Furthermore, the set Λ_θ is hyperbolic. For any point $x_0 \in \mathbb{R} \setminus I(\theta)$ we have $\lim_{n\to\infty} p_\theta^n(x_0) = \infty$.

(6) The mapping $p : x \mapsto x^2 + \theta$ is topologically conjugate to the mapping $f : x \mapsto \alpha x^2 + \beta x + \gamma$ via the homeomorphism $h : x \mapsto a x + b$, $\alpha \ne 0$, with $a = \alpha$, $b = \beta/2$ and $\theta = \alpha\gamma + (2\beta - \beta^2)/4$ where $p = h \circ f \circ h^{-1}$.

■ Real spiders and the spider map

Consider the n–periodic orbit containing the critical point $x = 0$ under the map p_θ for a suitable choice of θ:

$$x_0 = 0 \mapsto x_1 \mapsto \cdots \mapsto x_{n-1} \mapsto x_n = 0$$

Since $x_{i+1} = p_\theta(x_i)$ we have $x_i \in p_\theta^{-1}(x_{i+1})$ and the correct point to choose in the fiber $p_\theta^{-1}(x_{i+1})$ is given by the kneading symbol at that location in the periodic orbit. In our case, the fiber $p_\theta^{-1}(y)$ is empty if $y < \theta$, contains exactly one point if $y = \theta$ and contains two points if $y > \theta$. The coding, $S_\theta(x_0) = \{s_i\}_{i\ge 1}$, of an orbit under p_θ is done according to the rule

$$s_i = \begin{cases} \text{L if } p_\Theta^i(x_0) < 0 \\ \text{C if } p_\Theta^i(x_0) = 0 \\ \text{R if } p_\Theta^i(x_0) > 0 \end{cases}$$

The kneading sequence of p_θ is the symbolic orbit of the critical point, $K(\theta) = S_\theta(0)$. In our setting the kneading sequence is periodic. We will use the notation $K(\theta) = \overline{\text{LR}\cdots\text{C}}$ to denote that the finite symbol sequence underneath the bar is repeated an infinite number of times. It is easily seen that not all symbol sequences are compatible with the underlying dynamical system. In fact, it can be shown that there is at most one order of points that is compatible with a kneading sequence.

A real spider is a very special case of the spiders defined on the Riemann sphere for complex systems. On the Riemann sphere a spider is an equivalence class of curvesystems connected in ∞, the "body" of the spider, and the curves going out from this point may be thought of as the "legs". The legs are used to impose an ordering of the points in $\mathbb{C}$. However, in $\mathbb{R}$ there is a natural ordering, so the space of real spiders associated with the dynamical system p_θ takes the form of n–tuples of real numbers subject to a set of inequalities $x_1 < x_{j_1} < \cdots < x_{j_{n-2}} < x_2$ where $x_n = 0$.

The spider space

Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ and $\boldsymbol{j} = (j_1, j_2, \ldots, j_n)$ be an index vector where $j_i \in \{1, \ldots, n\}$ with $j_i \ne j_k$ if $i \ne k$. Let $S_{\boldsymbol{j}} \subset \mathbb{R}^n$ be the subset $S_{\boldsymbol{j},k} = \{\boldsymbol{x} \in \mathbb{R}^n : x_{j_1} < x_{j_2} < \cdots < x_{j_n} \text{ and } x_{j_k} = 0\}$. The space $S_{\boldsymbol{j},k}$ equipped with the natural inherited topology from $\mathbb{R}^n$ is called the real spider space associated with $(\boldsymbol{j}, k)$. A mapping $\boldsymbol{\sigma} : S_{\boldsymbol{j},k} \longrightarrow S_{\boldsymbol{j},k}$ is called a spider mapping. We will later index the space $S_{j,k}$ by a periodic admissible kneading sequence, writing $S_{\boldsymbol{j},k} = S_K$.

Example 1

Consider the real spider space $S_{(1,3,2),2} = \{(x_1, x_2, x_3) \in \mathbb{R}^3 : x_1 < x_3 < x_2 \text{ with } x_3 = 0\}$ and let $\boldsymbol{\sigma}(x_1, x_2, x_3) = \left(-\sqrt{x_2 - x_1}, \sqrt{-x_1}, 0\right)$. The map $\boldsymbol{\sigma}$ is clearly well–defined on $S_{(1,3,2),2}$ as $x_2 > x_1$ and the first component in the image is negative and the second component is positive. Suppose $\boldsymbol{\sigma}$ has a fixed point $\boldsymbol{\sigma}(\boldsymbol{x}) = \boldsymbol{x}$. Then $x_1 = -\sqrt{x_2 - x_1}$, $x_2 = \sqrt{-x_1}$ and $x_3 = 0$. We find $x_1^2 = x_2 - x_1$, $x_2^2 = -x_1$ and $x_3 = 0$. By rearranging these equations we have $x_1 = x_1$, $x_2 = x_1^2 + x_1$ and $x_3 = 0$. This corresponds exactly to the orbit $0 \to \theta \to \theta^2 + \theta \to 0$, and hence any such fixed point corresponds to a superstable three–periodic orbit under $p_\theta(x) = x^2 + \theta$.

Example 2

Consider the real spider space
$S_{(1,4,5,3,2),3} = \{(x_1, x_2, x_3, x_4, x_5) \in \mathbb{R}^5 : x_1 < x_4 < x_5 < x_3 < x_2 \text{ with } x_5 = 0\}$ and let $\boldsymbol{\sigma}$ be the map defined by

$$\sigma(x_1, x_2, x_3, x_4, x_5) = \left(-\sqrt{x_2 - x_1}, \sqrt{x_3 - x_1}, \sqrt{x_4 - x_1}, -\sqrt{-x_1}, 0\right) = (y_1, y_2, y_3, y_4, y_5).$$

Let us first check that this map is well–defined. We will show that $\boldsymbol{x} \in S_{(1,4,5,3,2),3} \Rightarrow \boldsymbol{\sigma}(\boldsymbol{x}) \in S_{(1,4,5,3,2),3}$. Note first that $x_2 > 0$ and $x_1 < 0$ so $x_2 - x_1 = x_2 + (-x_1) > -x_1 > 0$ so $y_1 = -\sqrt{x_2 - x_1} < -\sqrt{-x_1} = y_4 < 0 = y_5$. Note that $x_3 - x_1 > x_4 - x_1 > 0$ since $x_3 > x_4$ and $x_4 > x_1$. Hence $y_2 = \sqrt{x_3 - x_1} > \sqrt{x_4 - x_1} = y_3 > 0$. In other words, $\boldsymbol{x} \in S_{(1,4,5,3,2),3} \Rightarrow \boldsymbol{\sigma}(\boldsymbol{x}) \in S_{(1,4,5,3,2),3}$, and the mapping is well–defined. Assume as in example 1 that $\boldsymbol{\sigma}$ has a fixed point $\boldsymbol{x}$, that is, $\boldsymbol{\sigma}(\boldsymbol{x}) = \boldsymbol{x}$. The fixed point equation gives us that $x_1^2 = x_2 - x_1$, $x_2^2 = x_3 - x_1$, $x_3^2 = x_4 - x_1$, $x_4^2 = -x_1$ and $x_5 = 0$. If we rewrite these equations doing some substitutions, we might write them as $x_1 = 0 + x_1$, $x_2 = x_1^2 + x_1$, $x_3 = x_2^2 + x_1 = (x_1^2 + x_1)^2 + x_1$, $x_4 = x_3^2 + x_1 = \left((x_1^2 + x_1)^2 + x_1\right)^2 + x_1$ and $x_5 = 0$. This is exactly a critical 5–perodic orbit for our polynomial family p_θ where

$$0 \to \theta \to \theta^2 + \theta \to (\theta^2 + \theta)^2 + \theta \to \left((\theta^2 + \theta)^2 + \theta\right)^2 + \theta \to 0.$$

Hence a fixed point for this spider map corresponds to a superstable 5–periodic orbit with kneading sequence $\overline{\text{LRRLC}}$.

Example 3

In the two preceding examples we have used spider maps with fixed points corresponding to periodic critical orbits obeying certain combinatorial properties of the orbits. In the two first cases we have of course used kneading sequences compatible with the dynamics of the system $x \mapsto x^2 + \theta$. We will now choose a map corresponding to a non–compatible kneading sequence (later to be called a non–admissible kneading sequence). Consider the sequence $\overline{\text{LRLRC}}$. This gives us the real spider space $S_{(1,4,5,3,2),3} = \{(x_1, x_2, x_3, x_4, x_5) \in \mathbb{R}^5 : x_1 < x_3 < x_5 < x_4 < x_2 \text{ with } x_5 = 0\}$, and suggest that we define the spider map as

$$\sigma(x_1, x_2, x_3, x_4, x_5) = \left(-\sqrt{x_2 - x_1}, \sqrt{x_3 - x_1}, -\sqrt{x_4 - x_1}, \sqrt{-x_1}, 0\right) = (y_1, y_2, y_3, y_4, y_5).$$

We will show that $\boldsymbol{\sigma} : S_{(1,4,5,3,2),3} \longrightarrow S_{(1,4,5,3,2),3}$ is ***not*** well–defined. Note that since $x_1 < 0$ and $x_2 > x_4 > 0$ we have $x_2 - x_1 > x_4 - x_1 > 0$ so $y_1 = -\sqrt{x_2 - x_1} < -\sqrt{x_4 - x_1} = y_3 < 0$. Hence $y_1 < y_3 < 0 = y_5$. However, $x_3 < 0$ so $x_3 - x_1 < 0 - x_1$, that is, $y_2 = \sqrt{x_3 - x_1} < \sqrt{-x_1} = y_4$ implying that $S_{(1,4,5,3,2),3}$ is not closed under $\boldsymbol{\sigma}$. As a consequence of this, some of the roots of the component functions may become complex after a few iterations of σ. Indeed, this is exactly what is happening as we shall see later on.

Finding admissible kneading sequences using the minimality of the kneading sequence with respect to the lexicographic order

We will apply the theory developed by Milnor and Thurston in [MT] to obtain an algorithm to decide if a kneading sequence is admissible. This algorithm is based on the fact that the kneading sequence is minimal with respect to the lexicographic order denoted by $\preccurlyeq$ (defined below). In particular, if σ is the shift map on the symbol space, then $K \preccurlyeq \sigma^i(K)$, $i = 1, \ldots, n-1$, for any admissible kneading sequence of length n.

The lexicographic order

Let $\mathbb{A} = \{L, C, R\}$ be a three letter alphabet with the ordering $L < C < R$, and let $\Sigma = \mathbb{A}^{\mathbb{N}}$ be the set of infinite words from $\mathbb{A}$ due to the following restriction. If W_a and W_b are two words in Σ containing the letter C, $W_a = W_1 \, C \, W_2$ and $W_b = W_3 \, C \, W_4$, where W_1 and W_2 do not contain the letter C, then $W_2 = W_4$.

Let $S \in \Sigma$ where we write $S = \{s_i\}_{i \geq 0}$. Assume that $s_k \neq C$ for $0 \leq k \leq n$. We define $\tau_n(S) = \sum_{i=0}^{n} \nu(s_i) \bmod 2$ where $\nu(s_i) = 1$ if $s_i = L$ and $\nu(s_i) = 0$ if $s_i = R$. In other words, as S is the sequence of addresses coming from the dynamical orbit $\{p_\theta^k(x)\}_{k \geq 0}$ the quantity τ_n determines the orientation properties for p_θ^n at the point x. Note that p_θ is decreasing (orientation reversing) for $x < 0$ (corresponding to the symbol L) and increasing (orientation preserving) for $x > 0$ (corresponding to the symbol R).

A signed lexicographic ordering, denoted by $\prec$ (less) and $\preccurlyeq$ (less or equal), for two elements $S, T \in \Sigma$ is now defined as follows: Assume that $s_i = t_i$ for $0 \leq i \leq n-1$, then $S \prec T$ if either $\tau_{n-1}(S) = 0$ and $s_n < t_n$, or $\tau_{n-1}(S) = 1$ and $s_n > t_n$. We write $S \preccurlyeq T$ if $S \prec T$ or $S = T$.

The following results are used to construct the general algorithm to find the admissible kneading sequences:

Lemma 1. Let $K(f)$ be a kneading sequence of an unimodal map $f : I \longrightarrow I$, let $A(x)$ be the itinerary of a point $x \in I$ and let σ denote the shift map on the symbol space. Then $K(f) \preccurlyeq \sigma^i(A(x))$ for all $x \in I$ and $i \geq 0$. In particular, $K(f) \preccurlyeq \sigma^i(K(f))$.

Proof: See [X] or [MT].

We may use the special case with $A(x) = K(f)$ of this lemma to decide if a given candidate S for a periodic kneading sequence of length n is admissible. We simply need to test if $S \preccurlyeq \sigma^i(S)$ for $1 \leq i \leq n-1$.

Mathematica code

We will first generate possible candidates for admissible periodic kneading sequences. This can be done as below. In order to reduce the number of candidates we exclude some sequences that clearly not can be candidates. Clearly all sequences must start with LR if $n \geq 3$, and it is easy to prove that sequences of the form $\overline{\text{LRLWRC}}$ and $\overline{\text{LRLRWC}}$ where W is a word from $\{L, R\}$ including the empty word can not be admissible for $n \geq 5$. We may then code as follows:

```
nLs[n_] := StringJoin[Table["L", {n}]];
nRs[n_] := StringJoin[Table["R", {n}]];
BaseString[{n_, m_}] := nLs[n] <> nRs[m];
Generators[n_] :=
  Map[BaseString, Table[{n - i, i}, {i, 0, n}]];
KneadSeq[1] := {"C"}; KneadSeq[2] := {"LC"};
KneadSeq[3] := {"LRC"};
KneadSeq[n_Integer] :=
  Map["LR" <> # <> "C" &, Map[StringJoin, Flatten[Map[
      Permutations, Map[Characters, Generators[n - 3]]], 1]]];
ExclusionRuleOne[n_Integer] := Map["LRL" <> # <> "RC" &,
   Map[StringJoin, Flatten[Map[Permutations,
      Map[Characters, Generators[n - 5]]], 1]]];
ExclusionRuleTwo[n_Integer] := Map["LRLR" <> # <> "C" &,
   Map[StringJoin, Flatten[Map[Permutations,
      Map[Characters, Generators[n - 5]]], 1]]];
FilteredKneadSeq[n_Integer] := If[n < 5, KneadSeq[n],
   Complement[KneadSeq[n], Union[
     Flatten[{ExclusionRuleOne[n], ExclusionRuleTwo[n]}]]]];
```

We now need the signed lexicographic order for such strings. This can be done as follows:

```
LexOrder[x_, y_] := Module[{alphabet = {"L", "C", "R"}, px, py},
   px = First[Flatten[Position[alphabet, x]]];
   py = First[Flatten[Position[alphabet, y]]];
   Return[If[px == py, 0, If[px < py, -1, 1]]]];
SymbolValue[x_] := If[x == "L", 1, 0];
LxOrd[s_List, t_List, n_Integer] :=
  Module[{l = Length[s], i = 1, res = 0},
   If[s == t, Switch[n, -1, Return[0], 0,
     Return[False], 1, Return[True]]];
   While[(s[[i]] == t[[i]]) && (i < l),
    (res += SymbolValue[s[[i]]]; i++;)];
   If[EvenQ[res] && (LexOrder[s[[i]], t[[i]]] == -1), Switch[n,
     -1, Return[-1], 0, Return[True], 1, Return[True]]];
   If[OddQ[res] && (LexOrder[s[[i]], t[[i]]] == 1), Switch[n,
     -1, Return[-1], 0, Return[True], 1, Return[True]]];
   Switch[n, -1, Return[1], 0, Return[False],
    1, Return[False]]];
LexicographicOrder[s_List, t_List] := LxOrd[s, t, -1];
LGOrderLess[s_List, t_List] := LxOrd[s, t, 0];
LGOrderLessOrEqual[s_List, t_List] := LxOrd[s, t, 1];
s_ ≺ t_ := LGOrderLess[Characters[s], Characters[t]];
s_ ≼ t_ := LGOrderLessOrEqual[Characters[s], Characters[t]];
```

We then need some code to test our candidates above and return the admissible sequences among these. This can be done as follows:

```
MinimalSeqQ[s_] :=
  If[Max[Map[LexicographicOrder[Characters[s], #] &,
      Map[Characters,
       Table[StringJoin[RotateLeft[Characters[s], i]],
        {i, 1, Length[Characters[s]] - 1}]]]] ≤ 0, True, False];
GenerateAdmissibleKneadingSequences[n_Integer] :=
  Module[{lgl, gl, al = {}},
   If[n ≤ 6, Return[FilteredKneadSeq[n]]];
   lgl = Length[pl = FilteredKneadSeq[n]];
   Do[If[MinimalSeqQ[pl[[i]]], al = {al, pl[[i]]}], {i, 1, lgl}];
   Return[Flatten[al]]];
```

The following example shows what we wanted obtain. Due to the large number of the elements of words in the output we only count the number of words in each list:

```
{Length[FilteredKneadSeq[14]],
 Length[GenerateAdmissibleKneadingSequences[14]]}

{1280, 585}
```

■ The ordering of points in an admissible periodic kneading sequence (or forming the real spiders)

Let $K(f) = \overline{\text{LRWC}}$ be an admissible kneading sequence with $|K(f)| = n$ and W a word from the alphabet $\{L, R\}$ with $|W| = n - 3$. Let the corresponding dynamical orbit be $\{x_1, x_2, \ldots, x_n\}$, where $x_n = 0$ in our case. The problem is to order the points in the orbit on the real line $x_{j(1)} < x_{j(2)} < \cdots < x_{j(n)}$, in other words to find a bijective function $j : \{1, \ldots, n\} \longrightarrow \{1, \ldots, n\}$ based on the kneading information. Let $\#(K(f), L) = n_L$ and $\#(K(f), R) = n_R$ denote the number of L and R in the word LRW respectively. Clearly $n = n_L + n_R + 1$, and $n_L, n_R \geq 1$ if $n \geq 3$. We have some trivial information about the function j. Clearly $j(1) = 1$, $j(2) = n_L$ and $j(1 + n_L) = j(n - n_R) = n$. We will compute the image of j with the same trick as we used to compute admissible kneading sequences, we will sort points according to the lexicographic order. The following lemma from [MT] relates the order of points in the dynamic space with the order of addresses in the symbol space:

Lemma 2. Let $f : I \longrightarrow I$ be a unimodal map where the critical point is a minima, and let $x, y \in I$ with $x < y$ and let $A(x)$ and $A(y)$ denote their itineraries. Then $A(x) \preccurlyeq A(y)$ with respect to the signed lexicographical order.

Proof: See [MT] section 3.

We may apply this lemma to find the ordering of points in the dynamical space of an admissible kneading sequence given by $K(f) = \overline{\text{LRWC}}$ where W is a word from the alphabet $\{L, R\}$ in the following way. The symbols in the periodic word $\overline{\text{LRWC}}$ are asigned to symbolic points $x_1, x_2, \ldots, x_n$ in the dynamic space, that is, to indices $1, 2, \ldots, n$, and these are split into three groups according to their symbol in the kneading sequence. For example, the sequence $\overline{\text{LRLLRLC}}$ is mapped to $\{\{1, 3, 4, 6\}, \{7\}, \{2, 5\}\}$. The problem is then reduced to sort the first and third group according to their relative positions in the dynamic space. Now we just compare two versions of the symbol sequence $\overline{\text{LRWC}}$ by rotating left the correct number of times according to the symbol position in the string, so this symbol becomes the first symbol, given by the indices we already have found. We then apply the lemma above to determine their relative position in the dynamic space.

Mathematica code

The following code gives a version of the map j above operating on words corresponding to dynamical orbits:

```
SplitLCR[s_] := Map[Flatten,
   {Position[Characters[s], "L"], Position[Characters[s], "C"],
    Reverse[Position[Characters[s], "R"]]}];
MySortFunction[s_, u_Integer, w_Integer] :=
  If[LexicographicOrder[RotateLeft[Characters[s], u - 1],
     RotateLeft[Characters[s], w - 1]] ≤ 0, True, False];
JSortMap::"nonsequence" = "Non-admissible sequence: `1`.";
JSortMap[s_] := Module[{ll, lc, rl, nlst},
   If[! MinimalSeqQ[s], Message[JSortMap::"nonsequence", s];
    Return[Range[Length[Characters[s]]]]];
   nlst = SplitLCR[s];
   lc = nlst⟦2⟧;
   If[Length[nlst⟦1⟧] ≥ 2, ll =
     Sort[nlst⟦1⟧, MySortFunction[s, #1, #2] &], ll = nlst⟦1⟧];
   If[Length[nlst⟦3⟧] ≥ 2, lr = Sort[nlst⟦3⟧,
      MySortFunction[s, #1, #2] &], lr = nlst⟦3⟧];
   Return[Flatten[{ll, lc, lr}]]];
```

Here is an example with the admissible sequence $\overline{\text{LRLLRLC}}$:

```
JSortMap["LRLLRLC"]
```

```
{1, 4, 6, 3, 7, 5, 2}
```

We will use the function **JSortMap** to produce a function to generate a suitable element of the spider space associated with a given admissible kneading sequence $\overline{K}$ to be used as an initial point for the spider algorithm in the next section in order to generate the dynamical orbit for the system p_θ. We have chosen this spider to be equally spaces point in each of the intervals $[-2, 0)$ and $(0, 2]$ according to the numbers of L and R in the word K. The programming here is straight forward, we only need to find an "inverse" to the map described by **JSortMap**.

```
LRCount[s_] :=
  Map[Length, Map[Flatten, {Position[Characters[s], "L"],
     Position[Characters[s], "R"]}]];
LRCList[s_] := Module[{n, l, r},
   n = LRCount[s];
   l = Table[-2 + 2 i / n[[1]], {i, 0, n[[1]] - 1}];
   r = Table[2 i / n[[2]], {i, 1, n[[2]]}];
   Return[Flatten[{l, 0, r}]]];
InitSpider[s_] := Module[{ss, spider, index},
   ss = LRCList[s];
   spider = Range[Length[ss]];
   index = Map[Reverse,
     Sort[Table[{JSortMap[s][[i]], i}, {i, 1, Length[ss]}]]]; Do[
    spider[[index[[i, 2]]]] = ss[[index[[i, 1]]]], {i, 1, Length[ss]}];
   Return[spider]];
```

Here is an example with the admissible sequence $\overline{\text{LRLLRLC}}$:

```
InitSpider["LRLLRLC"]
```

$$\left\{-2, 2, -\frac{1}{2}, -\frac{3}{2}, 1, -1, 0\right\}$$

■ *Mathematica* implementation of a spider map

The simple examples in the third section of this paper suggest how we should define the spider map associated with a periodic kneading sequence. Consider the periodic dynamical sequence

$$0 \to x_1 \to x_2 \to \cdots \to x_{n-1} \to 0$$

where we have $x_{i+1} = x_i^2 + x_1$ for $0 \le i < n-1$ with $x_n = x_0 = 0$. Hence we have $x_i^2 = x_{i+1} - x_1$, so $x_i = s_i \sqrt{x_{i+1} - x_1}$ where $s_i \in \{-1, 0, 1\}$ if the corresponding kneading symbol is L, C or R.

The implementation of the real spider map choosing correct roots according to a given kneading sequence is easily done in *Mathematica*. Here, we will not perform any error or sanity checks, so our map simply takes the form

```
RealSpiderMap[k_List][l_List] :=
 Table[SpiderRoot[l[[1]], k[[i]]][l[[Mod[i, Length[l]] + 1]]],
  {i, 1, Length[l]}]
```

where **SpiderRoot** is the map returning the correct root according to the symbol in the kneading sequence. In our case, this map can be defined by

```
SpiderRoot[θ_, sym_][x_] := If[sym == "C", 0.0,
  If[sym == "L", N[-√(x - θ)], If[sym == "R", N[√(x - θ)]]]]
```

■ The Spider Algorithm

We will briefly describe the Spider Algorithm for the system $x \mapsto x^2 + \theta$.

We have seen in example 1 and 2 above that a critical periodic orbit is a fixed point for the spider map constructed by choosing the correct roots according to the combinatorics of the dynamical orbit. Example 1 shows that this fixed, in the special case of a 3−periodic orbit, is stable. One might hope that this is true in general for any critical periodic orbit, and hence this suggests the following algorithm:

Problem: Find a parameter value $\theta_K \in [-2, 0]$ such that the real dynamical system $x \mapsto x^2 + \theta_K$ has a periodic kneading sequence K.

The Real Spider Algorithm:
A) Choose a finite string K of length n of symbols where the first two symbols is LR and the $n-3$ next symbols is chosen from the two−letter alphabet $\{L, R\}$ and the last symbol is C.
B) Form the map $\boldsymbol{\sigma} : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ where the k−th component function is $y_k = s_k \sqrt{x_{k+1} - x_1}$ and $s_k = -1$ if the k−th symbol in the string is L, $s_k = 1$ if symbol is R and $s_k = 0$ if the symbol is C.
C) Choose a vector $\mathbf{x_0} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ where the points are ordered according to the dynamics of the periodic orbit for the dynamical system $x \mapsto x^2 + \theta$.
D) Form the sequence $\boldsymbol{x_{i+1}} = \boldsymbol{\sigma}(\mathbf{x_0})$ and stop the iteration process when the sequence of vectors converge to some point $\boldsymbol{y} \in \mathbb{R}^n$ (or in $\mathbb{C}^n$).
E) The parameter θ_K with the desired critical orbit is given by $\theta_K = y_1$.

In *Mathematica* this is implemented as the iteration process indicated below:

```
First[
    FixedPoint[
        RealSpiderMap[Characters[kneading_sequence]],spider]
     ]
```

Here **`kneading_sequence`** is a string of symbols and **`spider`** is an ordered list of real numbers. If the kneading sequence is not compatible with the dynamics then the returned number is non−real, that is, a number in $\mathbb{C} \setminus \mathbb{R}$.

We will define three functions associated with the numerical computation of the spider algorithm. These are **`SpiderIterationList[k,n]`**, **`SpiderFixedPoint[k,n]`** and **`CriticalParameter[k,n]`**. In all cases **`k`** is a kneading sequence, and **`n`** is an optional integer passed to **`FixedPoint`** or to **`FixedPointList`** controlling the maximum number of iterations in these functions. This is necessary in some cases because there is a "bit−flip" on the least significant bit at the fixed point, causing a non−stopping condition in **`FixedPoint`**. Note that it is checked if **`k`** is an admissible periodic kneading sequence. The function **`SpiderIterationList`** returns a list of the all steps in the iteration process of finding the fixed point of the spider map. The function **`SpiderFixedPoint`** returns simply the fixed point (the orbit) associated with the kneading sequence **`k`**, and **`CriticalParameter`** returns the first component of the fixed point, that is, the parameter θ for p_θ corresponding to the periodic kneading sequence.

```
SpiderIterationList[k_, n___Integer] := FixedPointList[
   RealSpiderMap[Characters[k]], InitSpider[k], n];
SpiderFixedPoint[k_, n___Integer] :=
  FixedPoint[RealSpiderMap[Characters[k]], InitSpider[k], n];
CriticalParameter[k_, n___Integer] :=
  First[SpiderFixedPoint[k, n]];
```

■ Examples

We will now give some examples using the code above. Some variations of the functions above are also used. We do not give the code for these. All code can be obtained from the author on request. We do not give the code for the graphics representations here either.

Example A

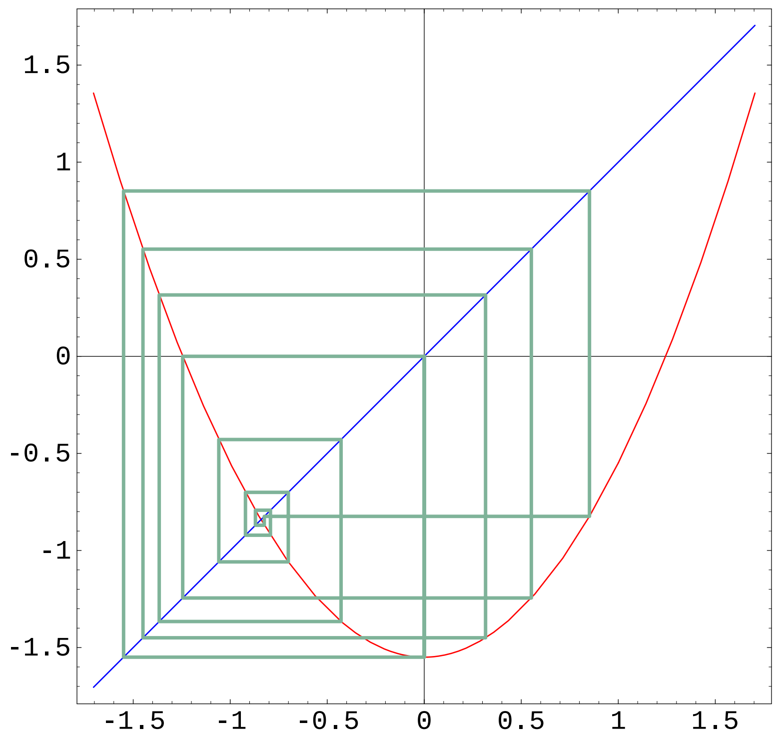
In this example we will display the dynamics of the orbit $\overline{\text{LRLLLLLLLLRLRLC}}$ of period 15 in two different ways. We first check that this string really represents an admissible periodic kneading sequence:

```
AdmissibleQ["LRLLLLLLLLRLRLC"]
```

```
True
```

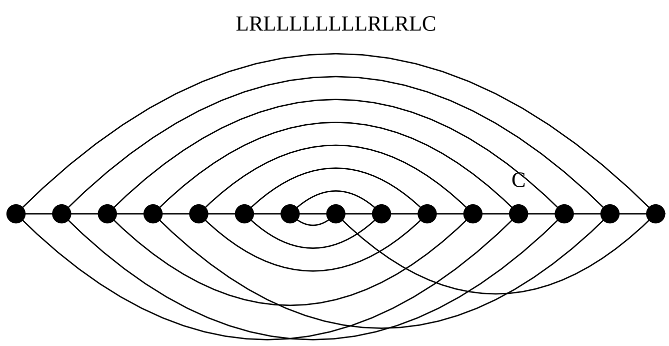
We can represent the superstable orbit $\overline{\text{LRLLLLLLLLRLRLC}}$ graphically in the usual way by drawing the graph (here in red) of p_θ for the value of θ corresponding to this sequence, draw the diagonal (here in blue) and then follow the orbit through the critical point with lines (here in green). It is easy to loose track of how the properties of the orbit are, by doing it this way. Below we will present a different method.

```
ShowDynamicOrbit["LRLLLLLLLLRLRLC"];
```



Another way is simply to display how point on the critical orbit move around under the map p_θ without taking the exact location in account, but just their relative locations, and which point is mapped to which, and in what direction. In the diagram below the points in the orbit of $\overline{\text{LRLLLLLLLLRLRLC}}$ are shown as black point along the horizontal axis with the critical point marked with a C. The curved lines represent how the point on the orbit are mapped, here curved lines above the horizon means a movement from left to right, and below the horizon, a movement from right to left.

```
ShowDiagramOrbit["LRLLLLLLLLRLRLC"];
```



Example B

In this example we compute all admissible sequences of length 9 and the corresponding critical parameter. Note that this corresponds to find certain real solutions of a polynomial of degree 2^8 . There are 28 admissible sequences of length 9. The function **`SortedAdmissibleSequences`** is described in the next example.

```
admseq = SortedAdmissibleSequences[9]
```

```
{LRRRRRRRC, LRRRRRRLC, LRRRRRLLC, LRRRRRLRC,
 LRRRRLLRC, LRRRRLLLC, LRRRRLRLC, LRRRRLRRC,
 LRRRLLRRC, LRRRLLRLC, LRRRLLLLC, LRRRLLLRC, LRRRLRLRC,
 LRRRLRLLC, LRRRLRRLC, LRRLLRRLC, LRRLLRLLC, LRRLLRLRC,
 LRRLLLLRC, LRRLLLLLC, LRRLLLRLC, LRRLRLRLC, LRRLRLLLC,
 LRRLRLLRC, LRLLRLLLC, LRLLRLRLC, LRLLLLRLC, LRLLLLLLC}
```

We use the function **`CriticalParameter`** to find the corresponding θ values for the dynamical system $x \mapsto p_\theta(x)$.

```
param = Map[CriticalParameter, admseq]
```

```
{-1.99994, -1.99949, -1.99859, -1.99722, -1.99542, -1.99313,
 -1.99038, -1.987, -1.98381, -1.97946, -1.97478, -1.96942,
 -1.96402, -1.95733, -1.94957, -1.93224, -1.92229, -1.91144,
 -1.90312, -1.89078, -1.87838, -1.84129, -1.82276,
 -1.78587, -1.69014, -1.65613, -1.59568, -1.55528}
```

Note that the solutions above are all the (real) zeros of the polynomial of degree 256 below corresponding to periodic orbits through zero of the dynamical system. The polynomial can be computed with **`Nest`**.

```
Short[Nest[pθ , 0, 9] // Expand, 12]
```

$$\theta + \theta^2 + 2\,\theta^3 + 5\,\theta^4 + 14\,\theta^5 + 42\,\theta^6 + 132\,\theta^7 + 429\,\theta^8 + 1430\,\theta^9 + 4606\,\theta^{10} + 14364\,\theta^{11} + 43810\,\theta^{12} + 131596\,\theta^{13} + 390964\,\theta^{14} + 1151240\,\theta^{15} + \ll 226 \gg + 1871297598686042576\,\theta^{242} + 224191664094990368\,\theta^{243} + 24788919621401424\,\theta^{244} + 2514273632010848\,\theta^{245} + 232268682367776\,\theta^{246} + 19378537561280\,\theta^{247} + 1445348279984\,\theta^{248} + 95166629216\,\theta^{249} + 5444445216\,\theta^{250} + 265070400\,\theta^{251} + 10676064\,\theta^{252} + 341440\,\theta^{253} + 8128\,\theta^{254} + 128\,\theta^{255} + \theta^{256}$$

The following table shows the periodic kneading sequence and the corresponding parameter θ:

```
TableForm[Table[
  {admseq[[i]], param[[i]] // InputForm}, {i, 1, Length[admseq]}],
 TableHeadings → {None, {"String", "Parameter"}}]
```

String	Parameter
LRRRRRRRC	-1.999943521765674
LRRRRRRLC	-1.999491438016398
LRRRRRLLC	-1.9985865888422067
LRRRRRLRC	-1.9972230246965785
LRRRRLLRC	-1.995419032661308
LRRRRLLLC	-1.9931302548789758
LRRRRLRLC	-1.990376381055951
LRRRRLRRC	-1.987004347515047
LRRRLLRRC	-1.983810249999715
LRRRLLRLC	-1.9794575048559522
LRRRLLLLC	-1.97478085890012
LRRRLLLRC	-1.9694191207308984
LRRRLRLRC	-1.9640243368201455
LRRRLRLLC	-1.9573250505356987
LRRRLRRLC	-1.949574903249391
LRRLLRRLC	-1.932243966576094
LRRLLRLLC	-1.9222857782462959
LRRLLRLRC	-1.9114446314734534
LRRLLLLRC	-1.9031167730155967
LRRLLLLLC	-1.890775424360235
LRRLLLRLC	-1.8783826015000962
LRRLRLRLC	-1.841288561509693
LRRLRLLLC	-1.8227563224922927
LRRLRLLRC	-1.7858656464106737
LRLLRLLLC	-1.6901422631188634
LRLLRLRLC	-1.6561325625742074
LRLLLLRLC	-1.5956809634397457
LRLLLLLLC	-1.5552827007685832

Example C

We will define a function **SortedAdmissibleSequences[n]** returning the admissible periodic kneading sequences of length **n** is sorted lexicographical order. In addition we have associated the symbols ≺ and ≼ with the lexicographical order.

```
SortedAdmissibleSequences[n_Integer] :=
 Sort[GenerateAdmissibleKneadingSequences[n],
  LGOrderLess[Characters[#1], Characters[#2]] &];
```

Here is an example with **SortedAdmissibleSequences** for sequences of length 15. There are 1091 different admissible sequences of this length. The function sorts these with respect to the lexicographical order. We have omitted 1081 strings in the output:

```
Short[SortedAdmissibleSequences[15], 6]
```

```
{LRRRRRRRRRRRRRC, LRRRRRRRRRRRRLC,
 LRRRRRRRRRRRLLC, LRRRRRRRRRRRLRC, LRRRRRRRRRRLLRC,
 <<1081>>, LRLLLLLLLRLRLLLC, LRLLLLLLLLLRLLLC,
 LRLLLLLLLLLRLRLC, LRLLLLLLLLLLRLC, LRLLLLLLLLLLLLLC}
```

We may use the symbols ≺ (Precedes) and ≼ (PrecedesSlantEqual) to test the lexicographic order of two strings. These relations work on any nonempty string from the alphabet $\{L, C, R\}$. The strings need not to be of equal length of course:

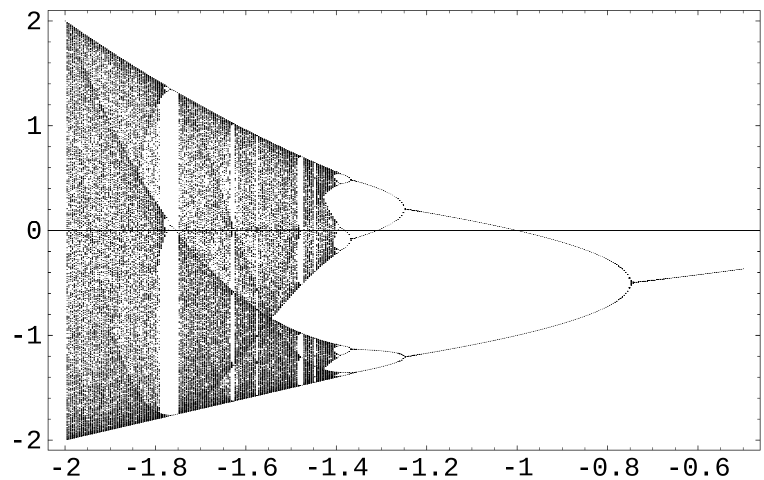
```
{"LRC" ≺ "LRLC", "LRLLLLLLLLLLRLC" ≼ "LRC", "CLLRC" ≺ "LLR"}
```

```
{True, False, False}
```

Example D

The bifurcation diagram in the figure below shows the attracting set of the critical orbit. We can not see the repelling periodic orbits in this diagram. The following theorem (the Sharkovsky theorem) states the relationship between co−existing periodic orbits without considering stability properties of these orbits.

```
ListPlot[
  LogisticBifucationDiagramData[-2, -0.5, 0.004, 0, 500, 400],
  Frame → True, PlotStyle → PointSize[0.001]];
```



The Sharkovsky ordering of ℕ. The natural numbers ℕ is ordered as follows by ⊳. The numbers *k* and *n* denote natural numbers (with respect to the usual ordering of ℕ), and the Sharkovsky ordering is given by:

$$3 \rhd 5 \rhd 7 \rhd 9 \rhd \cdots \rhd 2\,n+1 \rhd \cdots \rhd 2\cdot 3 \rhd 2\cdot 5 \rhd \cdots \rhd$$
$$2\cdot(2\,n+1) \rhd \cdots \rhd 2^k\cdot(2\,n+1) \rhd \cdots \rhd 2^k \rhd 2^{k-1} \rhd \cdots \rhd 8 \rhd 4 \rhd 2 \rhd 1$$

The Sharkovsky theorem. Let $f : I \longrightarrow I$ be a continuous map of some interval $I \subset \mathbb{R}$. If f has a periodic orbit of primitive period n, then f has periodic orbits of primitive period m for all m with $n \rhd m$ in the Sharkovsky ordering. In particular, if f has a periodic orbit of primitive period three, then f has periodic orbits of all periods.

See [S], [SMR] or [D] for a proof. As the reference [S] was written in Russian, and the result was unknown for a long time in the west. A proof of a special case of the Sharkovsky theorem, the theorem named "Period−3 implies chaos" was given in [LY] where the authors was unaware of the result in [S]. The proof in this case is however much easier than the proof of the Sharkovsky theorem.

Consider the dynamical system $x \mapsto x^2 + \theta$. The Sharkovsky ordering of ℕ has the odd numbers (1 excluded) as its greatest numbers in reverse order, $3 \rhd 5 \rhd 7 \rhd 9 \rhd \cdots \rhd (2\,n+1) \rhd \cdots$, $n \geq 1$. The Sharkovsky theorem implies that the first period $(2\,n+3)$−orbit must come into existence before or at the same time (with respect to the parameter) as a period $(2\,n+1)$−orbit as the parameter is varied from $\theta = -1$ to $\theta = -2$. As we have seen in our previous examples there are more than one admissible m−periodic kneading sequence if $m > 3$. Let θ_m denote the last occurrence of a superstable m−periodic orbit, m an odd number, for $\theta \in [-2, -1]$ with respect to the usual order in ℝ. We will find the sequence

$$\Theta_3 \leq \Theta_5 \leq \Theta_7 \leq \Theta_9 \leq \Theta_{11} \leq \cdots$$

using kneading sequences and the lexicographical order. Note that this sequence will not give the bifurcation points in the parameter space, but they will be quite good approximations, as the width of the windows containing the attracting periodic windows becomes very narrow at the left side of the bifurcation diagram above .

Let K be an admissible sequence of length $2n+1$ with $n>2$. It is easily shown that the maximal strings of this length are of the form LRL⋯LC. This means that these critical periodic orbits will be the first to appear when moving in the parameterspace from the right to the left (see the bifurcation diagram above). This fact will save us a lot of computations as we do not have to use the function **SortedAdmissibleSequences**. We may simply generate each sequence we need of length $2n+1$.

Equipped with the fact that the maximal odd kneading sequences are of the form $K=\overline{\text{LRL}\cdots\text{LC}}$ we may easily generate these for a consecutive sequence of odd numbers and compute the corresponding critical parameter using the spider algorithm:

```
UpperString[n_Integer] :=
  "LR" <> StringJoin[Table["L", {n - 3}]] <> "C";
kns = Table[UpperString[2 n + 1], {n, 1, 20}];
ind = Map[Length, Map[Characters, kns]];
sym = Table[θind[[i]] , {i, 1, Length[ind]}];
val = Map[CriticalParameter[#, 600] &, kns];
```

```
TableForm[Transpose[{sym, Map[InputForm, val]}],
 TableHeadings → {None, {"Parameter", "Value"}}]
```

Parameter	Value
θ_{3}	−1.7548776662466932
θ_{5}	−1.6254137251233038
θ_{7}	−1.5748891397523008
θ_{9}	−1.5552827007685832
θ_{11}	−1.547903761803955
θ_{13}	−1.5452017816926567
θ_{15}	−1.5442285601195278
θ_{17}	−1.5438809005277097
θ_{19}	−1.5437571734462723
θ_{21}	−1.5437132119079386
θ_{23}	−1.5436976024122815
θ_{25}	−1.5436920614376182
θ_{27}	−1.5436900947470673
θ_{29}	−1.543689396728154
θ_{31}	−1.543689148991056
θ_{33}	−1.543689061066118
θ_{35}	−1.543689029860556
θ_{37}	−1.543689018785357
θ_{39}	−1.5436890148546478
θ_{41}	−1.5436890134595964

Note that the calculations above corresponds to find particular non−trivial real solutions of polynomials of degrees in the range $\{2^2, 2^4, 2^6, \ldots, 2^{40}\}$.

Example E

We will consider what happens if the spider algorithm is applied to a non−admissible sequence. Consider the sequence $\overline{\text{LRLRC}}$ of length 5. This sequence is not admissible:

```
AdmissibleQ["LRLRC"]
```

```
False
```

In the following computation we apply the spider algorithm with this configuration, and we easily see that we should a initial spider of the form $\{-2, 2, -1, 1, 0\}$:

```
FixedPoint[
 RealSpiderMap[Characters["LRLRC"]], {-2, 2, -1, 1, 0}]

{-1.25637 - 0.380321 ⅈ, 0.177448 + 0.575325 ⅈ,
 -1.55588 - 0.17614 ⅈ, 1.13337 + 0.167784 ⅈ, 0.}
```

We obtain an orbit in $\mathbb{C}$. Even if we use a different initial spider we obtain the same orbit:

```
FixedPoint[RealSpiderMap[Characters["LRLRC"]],
 Table[Random[Real, {-2, 2}], {5}], 500]

{-1.25637 + 0.380321 ⅈ, 0.177448 - 0.575325 ⅈ,
 -1.55588 + 0.17614 ⅈ, 1.13337 - 0.167784 ⅈ, 0.}
```

This orbit is a critical orbit for the system $z \mapsto p_\theta(z)$ viewed as a complex dynamical system.

■ Conclusion

The main implementation issue in this work is not the implementation of the Spider Algorithm, which is trivial to implement, but rather to implement algorithms to decide if a given string of symbols is compatible with the dynamics of $x \mapsto x^2 + \theta$*, and how a dynamical orbit is ordered, using symbolic techniques, and how one can use symbolic dynamics to obtain numerical results. Mathematica provides excellent tools for this purpose.*

■ References

[B] D. A. Brown: *Using Spider Theory to Explore Parameter Spaces*. PhD−thesis, Cornell University. Advisor: J. H. Hubbard. **(2001)**

[**D**] R. Devaney: *An Introduction to Chaotic Dynamical Systems*. Addison−Wesley. (**1988**)

[HS] J. H. Hubbard and D. Schleicher: *The Spider Algorithm. In Complex Dynamics: The Mathematics behind the Mandelbrot and Julia sets*. AMS Lecture Notes. **(1994)**

[MT] J. Milnor and W. Thurston: *On Iterated Maps of the Interval*. Springer Lecture Notes in Mathematics, Vol. 1342, Springer−Verlag. **(1988)**

[S] A. N. Sharkovsky: *Coexistence of cycles of a continuous transformation of a line into itself*. Ukrain. Mat. Zhurn. 16. 1. **(1964)**

[LY] T. Y. Li and J. Yorke: *Period three implies chaos*. Amer. Math. Mon. 82. **(1975)**

[SMR] A. N. Sharkovsky, Yu. L. Maistrenko and E. Yu. Romanenko: *Difference Equations and Their Applications*. Kluwer Academic Press. **(1993)**

[X] H. Xie: *Grammatical Complexity and One Dimensional Dynamical Systems*. World Scientific, Singapore. **(1996)**