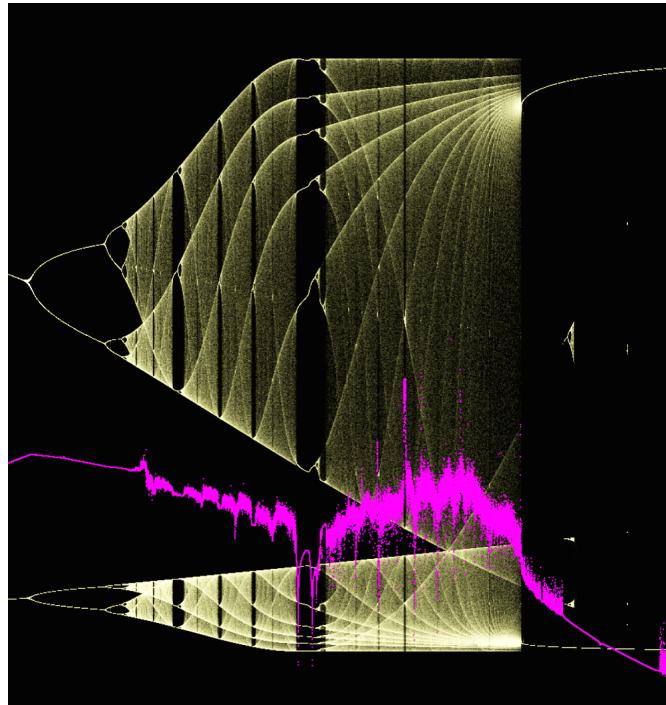


# Protokoll zum Versuch Nichtlineare Dynamik und Chaos

Nicolas Heimann, Jesse Hinrichsen

*Universität Hamburg*

2015



ABSTRACT TEXT TODO - vorläufig stand 300815

## 1 Einleitung

In diesem Protokoll haben wir die Eigenschaften von chaotischen System anhand der logistischen Gleichung, der Sinusabbildung, der Duffing-Gleichung und dem LDR Schwingkreis numerisch bestimmt und analysiert. Ebenfalls haben wir den LDR Schwingkreis in einem Experiment anhand eines analogen Oszilloskopes untersucht. Alle Plots und Simulationen in diesem Protokoll haben wir im Rahmen des Versuches selber implementiert. Dafür wählten wir als Programmiersprache Python2.7 und nutzten OpenGL4.1 und OpenCL für Visualisierungen und Berechnungen. Der Quellcode ist über github einsehbar: [https://github.com/keksnicoh/gl\\_plotting\\_experimental](https://github.com/keksnicoh/gl_plotting_experimental). Bei Abbildungen ist ein entsprechender Quellcodeverweis angegeben.

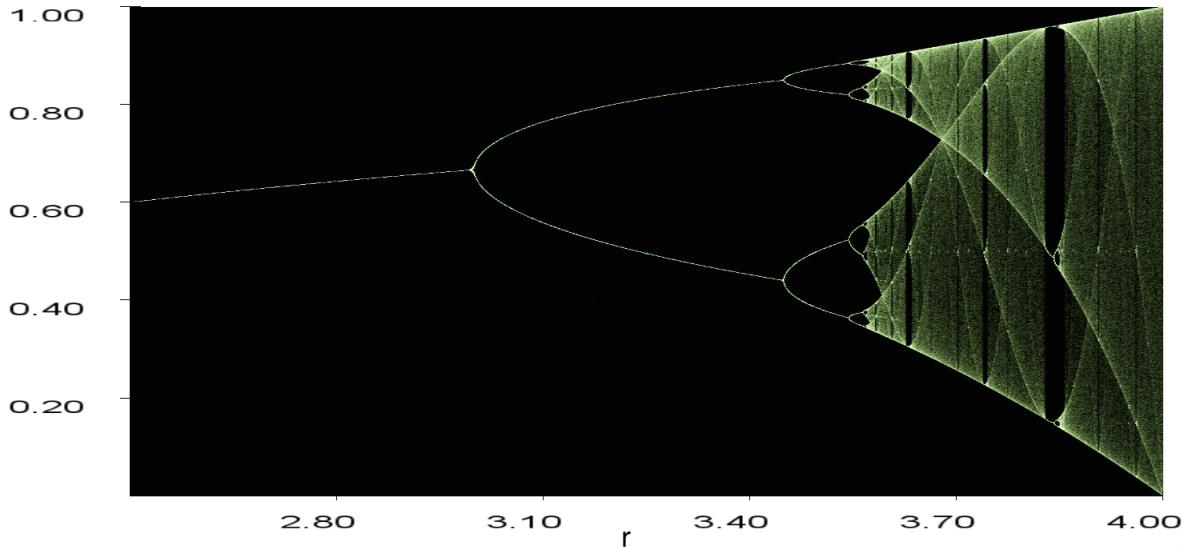
Im folgenden bezeichnet  $f^2(x) = f(f(x))$

# Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>1</b>
<b>2 Logistische Abbildung</b>	<b>2</b>
2.1 Fixpunkte / Stabilitätsbedingung . . . . .	3
2.2 Bifurkationspunkt . . . . .	4
2.3 Betrachtung von Extremwerten . . . . .	6
2.4 Lyapunov Exponent . . . . .	6
2.4.1 Konvergenzverhalten . . . . .	7
2.4.2 Betrachtung im chaotischen Bereich. . . . .	7
2.4.3 Vergleich der analytischen und renomierten Implementation . . . . .	8
2.5 Feigenbaumkonstante . . . . .	8
<b>3 Sinus Abbildung</b>	<b>10</b>
<b>4 Duffing-Gleichung</b>	<b>13</b>
4.1 Phasenraumdiagramm . . . . .	13
4.2 Poincareschnitt . . . . .	15
<b>5 LDR-Oszillator</b>	<b>15</b>
5.1 Theorie . . . . .	15
5.2 Numerische Berechnungen . . . . .	16
5.2.1 Phasenraumdiagramme . . . . .	17
5.2.2 Poincaré-Schnitt . . . . .	17
5.2.3 Bifurkationsdiagramm-Diagramm . . . . .	17
5.3 Versuchsaufbau . . . . .	23
5.4 Versuchsdurchführung . . . . .	23
5.4.1 Oszillation und Phasenraumdiagramm . . . . .	23
5.5 Bifurkationsdiagramm . . . . .	24
5.6 Vermessung der Bifurkationspunkte . . . . .	24
5.7 Zusammenfassung . . . . .	25
<b>6 Literatur</b>	<b>25</b>

## 2 Logistische Abbildung

Die logistische Abbildung ist gegeben durch  $f(x_n) = x_{n+1} = rx_n(1 - x_n)$ . Es zeigt sich das diese einfache Funktionsvorschrift bereits chaotisches Verhalten an den Tag legt welches wir im folgenden Abschnitt genauer untersucht haben. Zunächst haben wir ein Bifurkationsdiagramm der logistischen Abbildung erzeugt indem wir den Parameter  $r$  gegen Iterationspunkte aufgetragen haben. Dabei fixierten wir jeweils ein  $r$  und erzeugten eine Folge  $x_0 \dots x_{1000}$  von welcher wir  $x_{500} \dots x_{1000}$  gegen die Ordinate aufgetragen haben (Abbildung 1). Das Bifurkationsdiagramm lässt sich in mehrere Bereiche unterteilen. Bis  $r = 3$  laufen die  $x_{500} \dots x_{1000}$  auf den gleichen Fixpunkt zu. An  $r = 3$  gabelt sich das Diagramm in zwei Äste auf (Periodenverdoppelung). An  $r = 3.449$  gibt es eine weitere Perditionenverdopplung und es ist eine Selbstähnlichkeit mit dem Bereich um  $r = 3$  zu erkennen (fraktale Strukturen). Ab  $r = 3.569$  entsteht ein chaotischer Bereich in welchem sich aber noch Strukturen feststellen lassen (Bögen, Punkte auf Geraden, freie Bereiche).



**Abbildung 1:** Bifurkationsdiagramm der logistischen Abbildung im Bereich  $r \in [2.6, 4]$ . sourcecode: prak/birukation-logistisch-no-opt.py

## 2.1 Fixpunkte / Stabilitätsbedingung

Bildet die Funktion einen Punkt idempotent ab, so handelt es sich um einen Fixpunkt, es gilt:  $f^n(x) = x \forall n \iff x$  ist Fixpunkt. Stabilitätsbedingung:

$$|f'(x)| < 1 \iff \text{Fixpunkt stabil} \quad (1)$$

$$|f'(x)| > 1 \iff \text{Fixpunkt instabil} \quad (2)$$

Mit  $f(x) = rx(1-x) = x$  folgen die Fixpunkte  $x_0 = 0$  und  $x_1 = 1 - \frac{1}{r}$ . Mit der Stabilitätsbedingung (wir betrachten  $f'(x) = r - 2rx$ ) folgt für  $x_0$ :

$$-1 < r < 1 : x_0 \text{ stabil}$$

$$r > 1 : x_0 \text{ instabil}$$

Für  $x_1 = 1 - \frac{1}{r}$  folgt:

$$1 < r < 3 : x_1 \text{ stabil}$$

$$r < 1 \wedge r > 3 : x_1 \text{ instabil}$$

Abbildung 2 zeigt wie sich die Iteration bei unterschiedlichen Parametern verhalten. Für  $r > 3$  ist  $x_1$  nicht mehr stabil. So lässt sich die Aufspaltung im Bifurkationsdiagramm (Abbildung 1) für  $r > 3$  verstehen. Ebenfalls wurde das Konvergenzverhalten der logistischen Abbildung gegen  $x_1$  geprüft indem von den letzten beiden Werten jeweils der Mittelwert gebildet wurde. Bereits nach 30 Iterationen konnten alle  $x_1$  für  $1 < r \leq 3$  numerisch bestimmt werden. Ab  $r > 3$  funktioniert diese Methode nicht mehr, da die beiden Äste im Bifurkationsdiagramm nicht Punktsymmetrisch an  $x_1$  spiegeln (Abbildung 3). In Abbildung 2 ist für den Fall  $r = 3.113$  ebenfalls zu sehen, dass das Konvergenzverhalten asymmetrisch im Vergleich zum Fall  $r = 2.878$  ist. Im folgenden untersuchen wir den Zweierzyklus  $f^2(x)$ .

$$x_{n+1} = f_r(x_n) = rx_n(1-x_n) \Rightarrow x_{n+2} = r^2x_n(1-x_n)(1-rx_n(1-x_n)) \quad (3)$$

Fixpunktgleichung (Zweierzyklus):

$$x = r^2x(1-x)(1-rx(1-x)) \Rightarrow x_{3,4} = \pm \frac{\sqrt{r^2 - 2r - 3} + r + 1}{2r} \quad (4)$$

Da  $x_{3,4} \in \mathbb{R}$ , muss  $r^2 - 2r - 3 \geq 0$

$$\Rightarrow r \leq -1 \wedge r \geq 3 \quad (5)$$

Für diesen Bereich gibt es folglich 2 weitere Fixpunkte  $x_{3,4}$ . Es gilt:

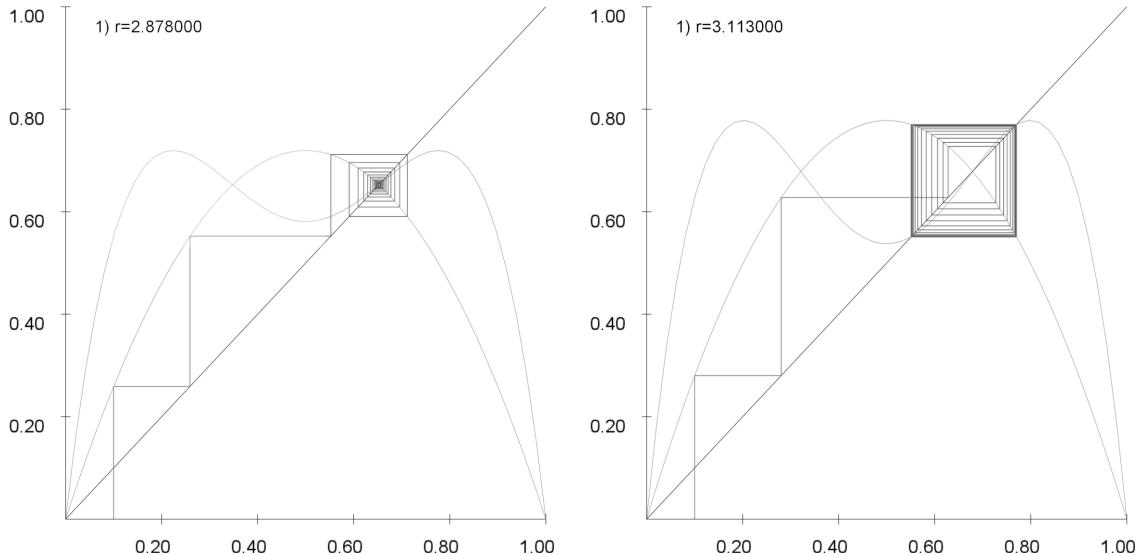
$$\frac{d}{dx} f^2(x) = -r^2(2x-1)(2r(x-1)x+1) \quad (6)$$

Lösungen der Gleichung  $|\frac{d}{dx} f^2(x_{3,4})| = 1$  sind:  $r_0 = -1$ ,  $r_1 = 3$ ,  $r_2 = 1 - \sqrt{6} \approx -1.45$ ,  $r_3 = 1 + \sqrt{6} \approx 3.45$ . Da die Fixpunkte  $x_{3,4}$  nur für  $r > 3$  und  $r < -1$  existieren folgt.

$$1 - \sqrt{6} < r \leq -1 \wedge 3 \geq r > 1 + \sqrt{6} : x_{3,4} \text{ stabil} \quad (7)$$

$$r < 1 - \sqrt{6} \wedge r > 1 + \sqrt{6} : x_{3,4} \text{ instabil} \quad (8)$$

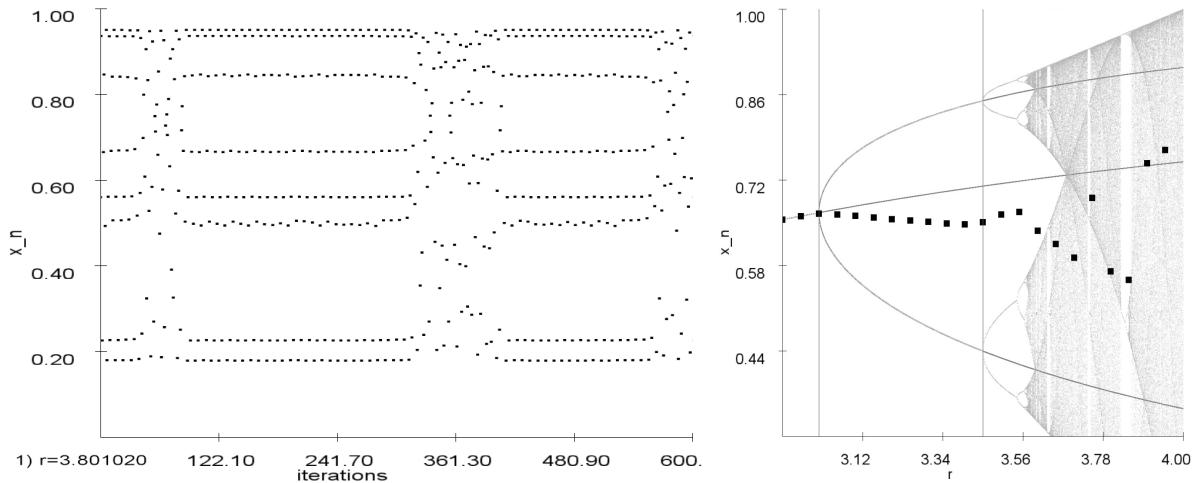
Die logistische Funktion zeigte bei einigen Parametern r Intermittenz (Abbildung 3). Intermittenz ist ein Grund für die Verdichtungen welche im chaotischen Bereich des Bifurkationsdiagrammes zu sehen sind.



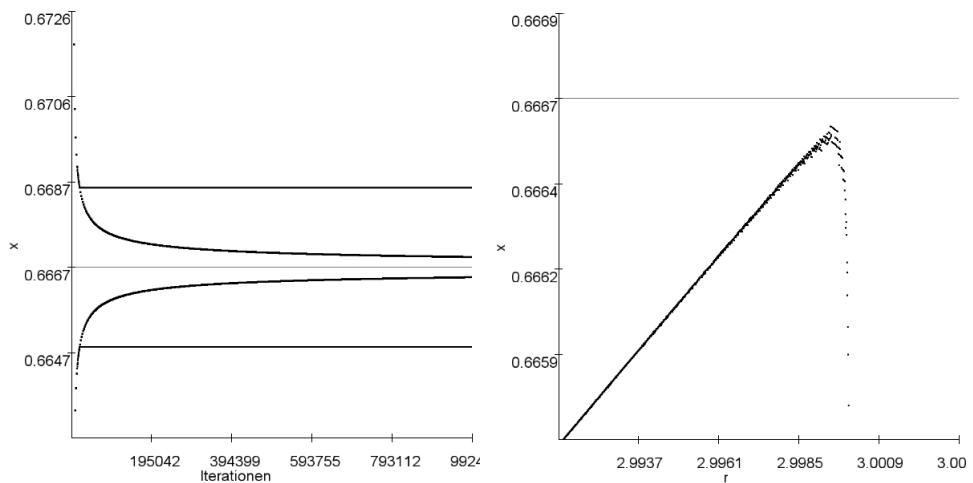
**Abbildung 2:** Verlauf der Iterationen bei festem Parameter r. Links: Konvergiert gegen Fixpunkt  $x_1$  bei  $r=2.878$ . Rechts: Bei  $r=3.113$  ist  $x_1$  kein stabiler Fixpunkt mehr. Der Verlauf der logistischen Funktion  $f(x)$ ,  $f^2(x)$  sowie die Einheitsgerade  $y = x$  sind aufgetragen. Als Linie Verbunden geplottet sind die Folgen  $(x_n, 0.0), (x_n, x_{n+1}), (x_{n+1}, x_{n+1}), (x_{n+1}, x_{n+2}), (x_{n+2}, x_{n+2}), (x_{n+2}, x_{n+3}), \dots$  Sourcecode: prak/logisitsch-no-opt-behavior.py

## 2.2 Bifurkationspunkt

Im weiteren betrachten wir das Konvergenzverhalten der logistischen Abbildung bei  $r = 3$  und Umgebung. Für  $r < 3$  kennen wir bereits den Fixpunkt  $x_1 = 1 - \frac{1}{r}$ . Abbildung 4 (links) trägt den x Wert gegen die Anzahl der Iterationen bei  $r = 3$  auf. Dabei sei zunächst angemerkt, dass wir bei der Berechnung über einen OpenCL Kernel mit double precision, also float64 anscheinend sehr schnell auf ein Rundungsproblem stoßen, während wir bei der Berechnung in Python anscheinend wesentlich höhere Genauigkeit erreichen. So haben wir nach  $10^7$  Iterationen ein  $\epsilon = 7.67 \cdot 10^{-5}$ , wobei  $\epsilon = |\frac{2}{3} - x|$ . [!htbp] Der Bereich um den Bifurkationspunkt ist in Abbildung 4 (rechts) dargestellt. Dabei wurde zunächst OpenCL für die Berechnung verwendet, was zu einem ungenauem Ergebnis führt. Es lässt sich nur Grob ablesen, dass bei ca.  $r = 3$  der Fixpunkt  $x_1$  instabil ist.



**Abbildung 3:** Logistische Funktion. Links: Intermittenz bei  $r = 3.80102$ . Rechts: Die stabile Lösung des Einerzyklus  $x^*$  und die beiden Lösungen des Zweierzyklus  $x_{3,4}$ . Im Hintergrund das Bifurkationsdiagramm. Die Punkte sind die das Ergebniss der numerischen Berechnung der Fixpunkte  $x_1$ . Die vertikalen Linien sind an  $r = 3.0$  und  $r = 3.44$  und markieren die Stellen wo  $|f'(x)| = 1$ ,  $|\frac{d}{dx} f^2(x)| = 1$  sind. Sourcecode: prak/intermittenz.py, prak/logis-zyklen.py



**Abbildung 4:** Links: Konvergenzverhalten der logistischen Abbildung bei  $r = 3$  untersucht mit zwei Methoden. OpenCL Berechnung: Nach ca.  $1.6 \cdot 10^4$  konstant bei  $x_n = 0.6685$ , bzw.  $x_{n+1} = 6648$ . Python Berechnung:  $x$  konvergiert kontinuierlich gegen  $x = \frac{2}{3}$ . Rechts: Vergrößerung des Bifurkationsdiagrammes bei  $r \approx 3$ .  $x$  nach  $10^5$  Iterationen aufgetragen.

## 2.3 Betrachtung von Extremwerten

Das Bifurkationsdiagramm scheint ab  $r > 4$  zu divergieren. Im folgenden haben wir diesen Bereich näher analysiert. Wegen  $x^2 < x \forall x \in (0, 1)$  ist schnell klar, dass sobald  $f(I) = [0, 1]$  auch  $f^n(I) = [0, 1]$ . Falls aber gilt  $x - x^2 < 0$  dann stellt man fest:

$$f^n(x) < \dots < f^2(x) < f(x) < 0 \forall n \in \mathbb{N} \Rightarrow \lim_{n \rightarrow \infty} f^n(x) = -\infty \quad (9)$$

Damit dieses Verhalten irgendwann einsetzt muss gelten:  $\exists m : f^m(x) > 1$  denn dann ist  $f^{m+1}(x) < 0$ . (wegen  $x^2 > x \forall x > 1$ ).

$$f(x_0) > 1 \iff x_0 - x_0^2 > \frac{1}{r} \iff x_0 = \pm \sqrt{\frac{1}{4} - \frac{1}{r}} + \frac{1}{2} \quad (10)$$

Also  $\exists x_0 \in \mathbb{R} \forall r > 4 : f(x_0) > 1 \Rightarrow f^2(x_0) < 0$  ist. Es ist also zu erwarten, dass

$$\exists m : f^m(x_0) > 1 \Rightarrow \lim_{n \rightarrow \infty} f^n(x_0) = -\infty \forall x_0 \forall r > 4 \quad (11)$$

In prak.iteration-log-extreme.py haben wir nun zu den Anfangsbedingungen  $x_0 = 0.4, r \in [3.99999, 4.999989], \Delta r = 0.000001$  überprüft wie viele Punkte r nach n Iterationen gegen  $-\infty$  divergieren. Dabei haben  $n = 1, n = 10, n = 100, n = 1000, n = 10000$  Iterationen durchgeführt mit folgendem Ergebniss.

---

```
python -m prak.iteration-log-extreme
nothing at N_ITER=1
nothing at N_ITER=10
N_ITER=100 found 680 intervals. Last interval [4.072665, 4.999989]
N_ITER=1000 found 17 intervals. Last interval [4.000251, 4.999989]
N_ITER=10000 found 1 intervals. Last interval [4.000001, 4.999989]
```

---

Je mehr Iterationen durchgeführt werden, desto mehr Punkte divergieren ab  $r \geq 4.000001$  gegen  $-\infty$ . Dieses Verhalten war nach Gleichung (11) zu erwarten.

## 2.4 Lyapunov Exponent

Der Lyapunov Exponent beschreibt mit welcher Geschwindigkeit sich zwei naheliegende Punkte voneinander entfernen. Es gibt drei Wege den Lyapunov Exponenten zu implementieren:

$$Definition : \lambda(x_0) = \lim_{N \rightarrow \infty} \lim_{\epsilon \rightarrow 0} \frac{1}{N} \log \left| \frac{f^N(x_0 + \epsilon) - f^N(x_0)}{\epsilon} \right| \quad (12)$$

$$Analytisch : \lambda(x_0) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=0}^{N-1} \log f'(x) \quad (13)$$

$$Renormiert \quad (14)$$

In Abbildung 9 ist der Lyapunov Exponent der analytischen Implementation gezeigt. Es zeigte sich, dass die Gleichung (12) sehr schlechte Ergebnisse im Vergleich zu den beiden letzten Methoden (13), (14) ergibt. Dies liegt daran, dass schon bei kleinen  $\epsilon$  die Funktionswerte sehr schnell divergieren und somit die Definition des Differenzenquotienten in (12) keinen Sinn ergibt. Die Renomierung (14) hält diesen Abstand in jedem Iterationsschritt klein, weshalb sich der Lyapunov Exponent trotzdem ausrechnen lässt. Der Lyapunov Exponent hat  $\lambda(x_0)$  seine Nullstellen dort wo die Abbildung ihre periodenverdoppelnden Stellen hat. Umgekehrt divergiert  $\lambda(x_0)$  an den superattraktiven Stellen gegen  $-\infty$ . Man erhält also Information über das Verhalten der Abbildung für bestimmte  $x_0$ . Tatsächlich kann man den Lyapunov-Exponenten über den mittleren Informationsverlust ausdrücken  $\lambda(x_0) = -\log(2) * \delta I$  (QUELLE skript todo). Im folgenden ist der OpenGL Quellcode der renomierten Formel des Lyapunov Exponenten gezeigt:

---

```

float g(float r, float x) {
    return r * x * (1-x);
}
vec4 f(vec4 x) {
    float x0 = 0.4;
    float eps = 0.0001;
    float n = 10000;
    float summe = x0;
    for (int i=1; i < n; i++) {
        x0 = g(x.x, x0);
        summe += log(abs(g(x.x, x0+eps)-g(x.x, x0))/eps);
    }
    return vec4(x.x, summe/n, 0, 0.5);
}

```

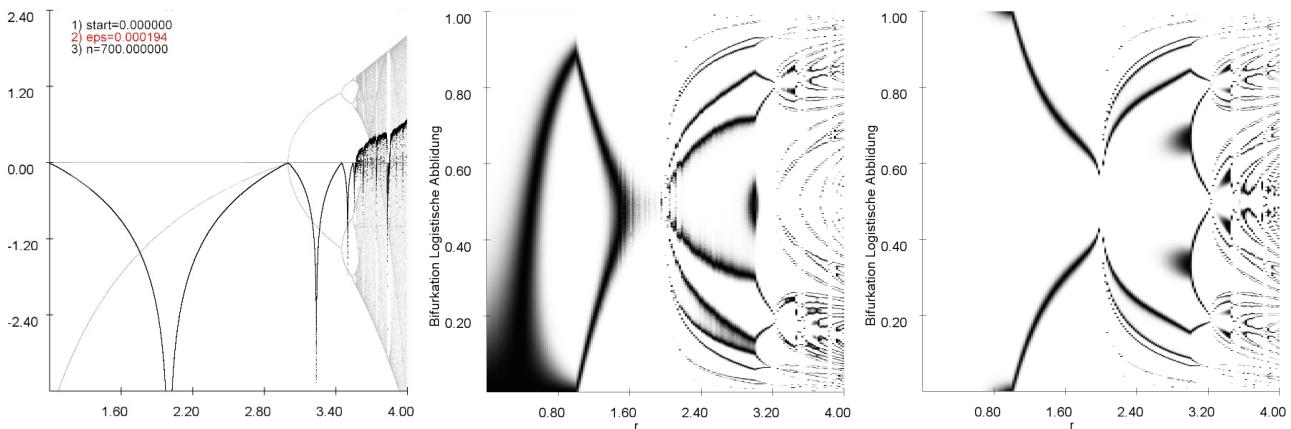
---

#### 2.4.1 Konvergenzverhalten

Es kann passieren, dass man einen Anfangswert wählt, für welchen der Lyapunov-Exponent kein sinnvolles Ergebniss liefert. Setzt man  $x_0 = 0.5$  so divergiert die analytische Formel, denn  $\log(f'(x_0)) = -\infty$ . Für die renormierte und analytische Implementation haben wir untersucht wie schnell bestimmte Anfangswerte  $x_0$  gegen den Grenzwert  $\lambda(x_0)$  streben. Dafür wurde untersucht wie nah  $\lambda_{N=5}(r, x_0)$  an  $\lambda_{N=1000}(r, x_0)$  ist:

$$\gamma = \exp^{-500*(\lambda_{N=5}(r, x_0) - \lambda_{N=1000}(r, x_0))} \quad (15)$$

In Abbildung 5 ist das Ergebniss zu sehen: Je dunkler ein Punkt  $(r, x_0)$  ist, desto besser konvergiert  $\lambda(r, x_0)$  ( $\gamma \approx 1$ ). Es gibt also  $x_0$  für welche der Lyapunov Exponent abhängig von der Implementation besser konvergiert. Der oben besprochene Startwert  $x_0 = 0.5$  ist in der rechten Abbildung gut bestätigt, denn hier konvergiert  $\lambda(r, x_0)$  nicht.



**Abbildung 5:** Analyse des Lyapunov Exponenten. Links: renormiert TODO. Mitte: Kontrollparameter  $r$  aufgetragen gegen Definitionsbereich  $x_0$ . Je dunkler ein Punkt  $(r, x_0)$  desto besser konvergiert  $\lambda(r, x_0)$ . Rechts: Analog zur Mittleren Abbildung für die analytische Implementation

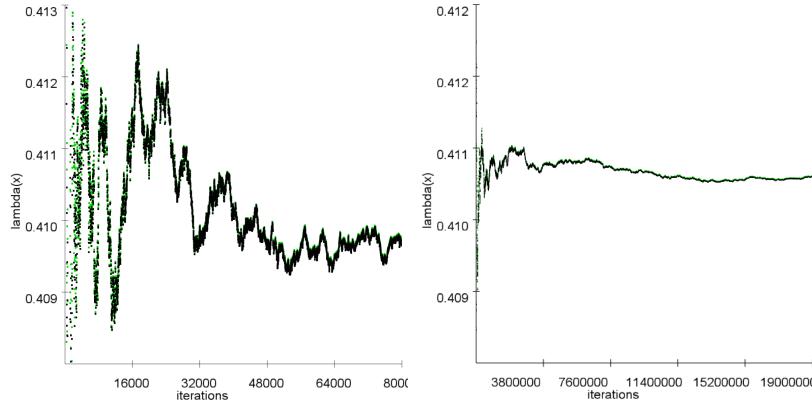
#### 2.4.2 Betrachtung im chaotischen Bereich.

In Abbildung 5 ist zu sehen wie der Lyapunov Exponent ab  $r = TODOPRÄZISE$  nicht mehr stetig konvergiert. Der Verdacht liegt nahe, dass nicht genügend Iterationsschritte zur Berechnung ausgeführt wurden also erhöhten wir die Anzahl an Iterationsschritten für die Berechnung von  $\lambda(x)$ . Es zeigte sich allerdings kaum eine Veränderung. Die Frage nach der Konvergenz von  $\lambda(x)$  im chaotischen Bereich galt es zu untersuchen: Abbildung 6 zeigt das Langzeitverhalten von  $\lambda(x)$  über  $19 * 10^6$  Iterationen.

$\lambda(x)$  erinnert die ersten  $10^7$  Iterationen eher an einen Börsenkurs als ein Objekt welches gegen einen Grenzwert konvergiert. Erst nach ca.  $10^7$  Iterationen deutet sich Konvergenzverhalten an. Dieses ist aber nicht präzise: Falls

$$\lim_{N \rightarrow \infty} \lambda_N(x) = L \quad (16)$$

so kann man nach  $2 * 10^7$  Iterationen höchstens feststellen, dass der Graph in einen Epsilon Schlauch von  $\epsilon = 0.002$  passt. Dies ist nicht sehr befriedigend angesichts der massiven Iterationslänge.



**Abbildung 6:** Analytischer(schwarz) und renormierter(grün) Lyapunov Exponent im chaotischen Bereich bei  $r = 3.78$ . Links: Zoom im Bereich 0 bis  $80 * 10^3$  Iterationen. Rechts: 0 bis  $19 * 10^6$  Iterationen.

### 2.4.3 Vergleich der analytischen und renomierten Implementation

Wir haben nun die analytische und die renormierte Implementation des Lyapunovexponenten weiter untersucht. Dabei stellten wir fest, dass bei Stellen mit Periodenverdoppelung die renormierte Formel etwas schneller gegen 0 konvergiert als die analytische Formel. An einer weiteren Stelle ( $r = 3.05$ ) ist zu erkennen wie beide Implementationen jeweils von oben (renormiert) und von unten (analytisch) gegen einen gemeinsamen Wert streben (Abbildung 7). Die Vermutung liegt nahe, dass man mit dem Mittelwert aus beiden Implementationen an solchen Stellen wesentlich schneller den Grenzwert bestimmen kann. Es zeigt sich aber, dass dieses Verhalten nicht regelmäßig auftritt weshalb wir es nicht weiter untersucht haben. Nach weiteren Stichproben scheint die analytische Implementation bis zum 500ten Iterationsschritt etwas schneller als die renormierte Implementation zu konvergieren. Beide Versionen zeigten in allen Stichproben  $r < 3$ , dass sie stets gegen den gleichen Grenzwert strebten.

## 2.5 Feigenbaumkonstante

Die Feigenbaumkonstante ist eine universelle Größe. Sie tritt in chaotischen nicht linearen System auf und ist definiert:

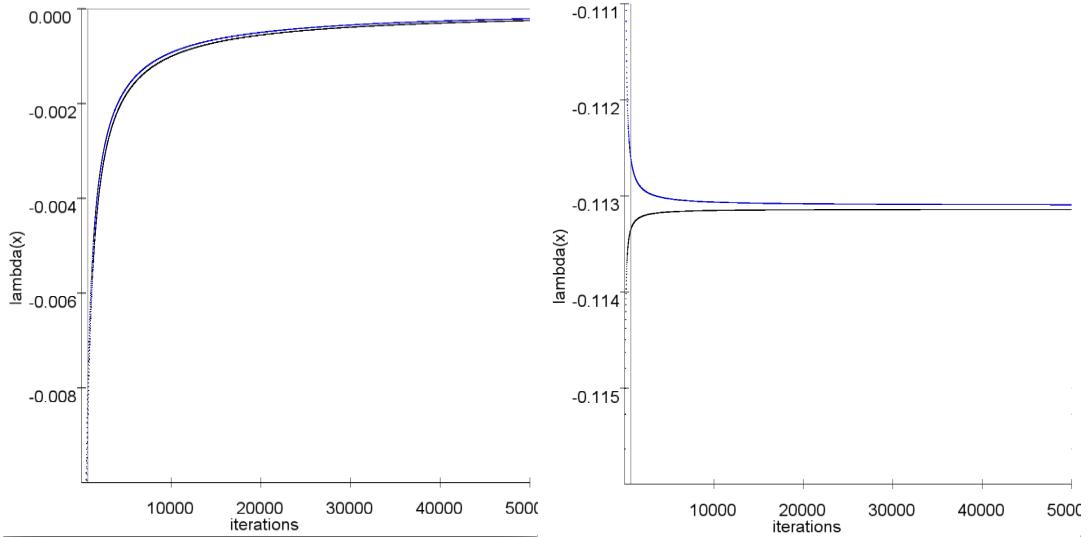
$$\delta_i = \frac{b_i - b_{i+1}}{b_{i+1} - b_{i+2}} = \frac{s_i - s_{i+1}}{s_{i+1} - s_{i+2}} \quad (17)$$

$$\delta = \lim_{i \rightarrow \infty} \delta_i = 4.669201609102991 \quad (18)$$

(TODO QUELLE <https://oeis.org/A006890>) wobei  $s_i, b_i$  die Folgen der superattraktiven und periodenverdoppelten Stellen sind. Also lässt sich die Feigenbaumkonstante mit dem Lyapunov Exponenten bestimmen. Als erstes haben wir versucht die Nullstellen des Lyapunov Exponenten zu bestimmen. Wir wendeten dabei das folgende Kriterium für Nullstellen an:

$$\lambda(x - \epsilon) < \lambda(x) \wedge \lambda(x + \epsilon) < \lambda(x) \wedge |\lambda(x)| < \epsilon \quad (19)$$

Bei genauer Betrachtung des numerisch bestimmten Lyapunov Exponenten zeigten sich große Schwankungen, weshalb dieses Kriterium nicht mit unseren Verfügbareren Rechenleistungen praktikabel war (Abbildung 8). Aufgrund dieser Probleme haben wir uns dazu entschieden die superattraktiven Stellen anstelle



**Abbildung 7:** Vergleich des analytischen(Schwarz) und renormierten(Blau) Iterationsverhalten des Lyapunov Exponenten. Vertikale Linie bei  $N=700$ . Linkes Bild zeigt Iterationsverhalten bei  $r = 3.0$ . Rechtes Bild zeigt Iterationsverhalten bei  $r = 3.05$

der Bifurkationspunkte numerisch zu bestimmen. Unser Algorithmus startet im Suchmodus bei gegebenen Startwert  $x_{start}$  und geht in kleinen Schritten  $\Delta x$  die x-Achse ab. In jedem Schritt wird  $l_1 = \lambda(x_0)$  berechnet. Ist

$$l_2 - l_1 \leq 0 \quad (20)$$

wandert der Iterationsschritt zum superattraktiven Fall. Dies wird so lange fortgesetzt bis die Bedingung (20) nicht mehr hält. Es wird nun um  $\Delta x$  zurückgegangen und anschließend die Schrittweite  $\Delta x \mapsto \frac{\Delta x}{10}$  verkleinert. Nun wird erneut so lange iteriert, bis (20) nicht mehr hält. Der Vorgang wiederholt sich 8 mal. Anschließend wird  $(2 * x_0 + \Delta x)/2$  als Ergebniss gespeichert. Als nächstes befindet sich der Algorithmus im Anfangspunkt-Modus. Es wird so lange die x-Achse abgetestet bis die Bedingung

$$\lambda(x_0) < \lambda(x_0 + \Delta x) < \lambda(x_0 + 2 * \Delta x) \quad (21)$$

nicht mehr erfüllt ist und somit ein neues  $x_{start}$  gefunden wurde. Der Suchmodus wird aktiviert (Quellcode: prak/feigenbaum.py). Im folgenden ist die Terminal Ausgabe des Algorithmus beispielhaft:

---

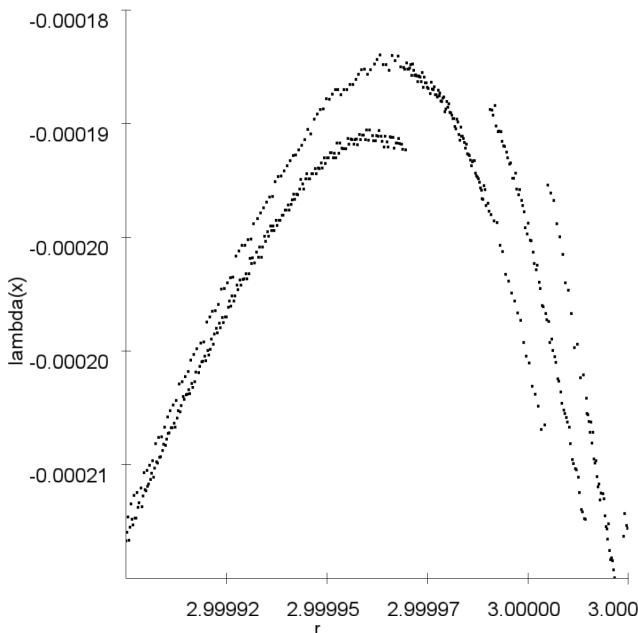
```

searching from 1.9
looking for next start_r from 2.00000000002
searching from 2.99950000003
looking for next start_r from 3.23606797751
searching from 3.44927797752
looking for next start_r from 3.49856169934
searching from 3.54400769935
looking for next start_r from 3.55464086278
searching from 3.56439786279
looking for next start_r from 3.56666737986
found values [2.0000000000249916, 3.236067977509959, 3.498561699344952,
               3.554640862779951, 3.5666673798649517]
delta_0=4.70894301336
delta_1=4.68077099865
delta_2=4.66295961155

```

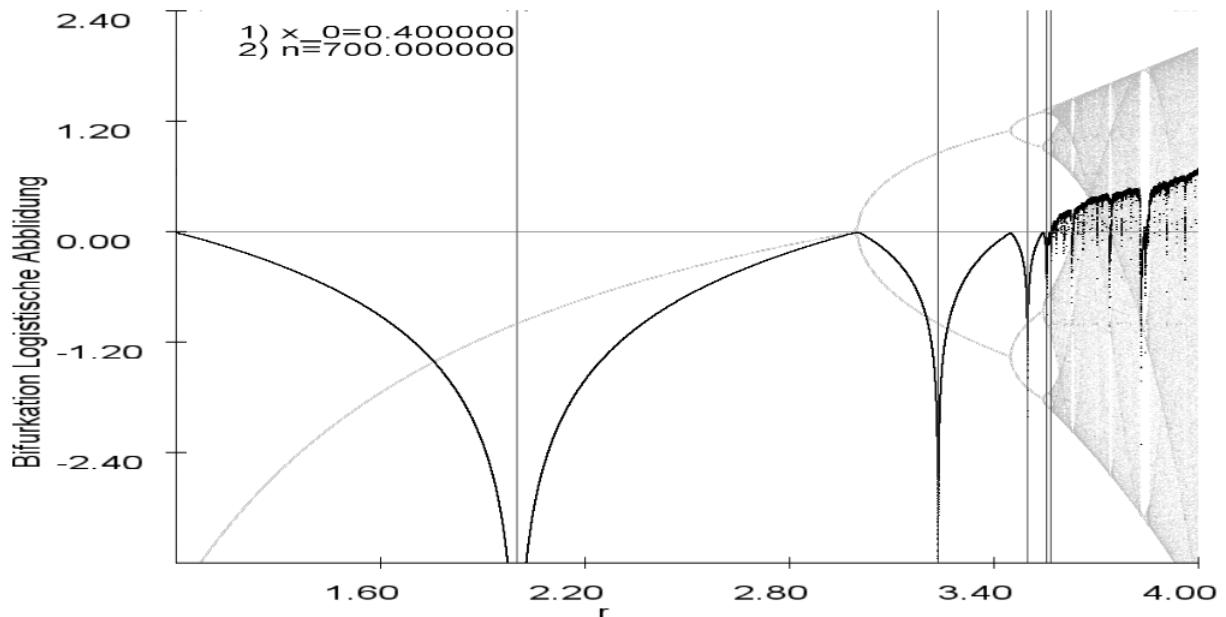
---

Somit konnten wir numerisch für  $i = 2$  eine Feigenbaumkonstante von  $\delta_2 = 4.66295961155$  berechnen. Dieser Wert weicht um 0.12% vom tatsächlichen Wert ab. Die Grenzen des Algorithmus sind bereits nach 5 gefunden superattraktiven Stellen erreicht. Ab  $r > 3.57$  fängt die numerische Implementation des Lyapunov Exponenten an zu "rauschen" (Abbildung 9). Der Algorithmus kann daher nicht mehr



**Abbildung 8:** Zoom in den Lyapunovexponenten der logistischen Abbildung bei  $r = 3$  mit 80000 Iterationen. Eine präzise Konvergenz ist nicht zu erkennen wodurch die numerische Bestimmung der Spitzen an den Bifurkationspunkten erschwert wird. Sourcecode: [prak/lyapunov-superattraktiv.py](#)

präzise seinen Suchmodus ausführen. Ebenfalls liegen die nächsten superattraktiven Punkte noch dichter zusammen als es für  $s_4, s_5$  der Fall war, was ebenfalls vom Algorithmus nicht mehr detektiert wird.



**Abbildung 9:** Analyse des Bifurkationsdiagrammes der logistischen Abbildung. Eingezeichnet sind die  $y=0$  Achse, sowie die ersten 5 Superattraktiven Stellen. Das Bifurkationsdiagramm wurde so translatiert und skaliert, so dass es hinter dem Lyapunov Exponenten erscheint.

### 3 Sinus Abbildung

Die Sinusabbildung ist gegeben durch

$$f(x_n) = x_{n+1} = r * \sin(x)$$

Wir haben die bereits implementierten Programme nun auf die Sinus Funktion angewendet. Abbildung 10 zeigt das Bifurkationsdiagramm der Sinusabbildung. Es fällt sofort auf, dass sich die Bildpunkte

innerhalb einer Einhüllenden befinden. Betrachtet man die Funktion wird wegen des Bildbereiches des Sinus ( $[-1, 1]$ ) sofort klar, dass  $\sup_{n \in \mathbb{N}} |f^n(r)| \leq r$  gilt. Bei den Bifurkationsdiagrammen der logistischen Abbildung hatten wir ein festen Startwert  $x_0$  gewählt. Für die Sinusabbildung wählten wir als Startwerte die Folge  $x_{0,n} = (-1)^n * x_0$  da sonst im Bereich  $r \in [0, \pi]$  das Bild des Graphen stets positiv wäre. Auch dieses Phänomen lässt sich mit einfachen Überlegungen erklären:

$$\sin([0, \pi]) = [0, 1] \Rightarrow f([0, \pi]) = [0, \pi] \Rightarrow f^2([0, \pi]) = [0, \pi] \forall r \in [0, \pi]$$

Das Anwenden unseres Algorithmus zur bestimmung der Feigenbaumkonstante lieferte im Vergleich zur logistischen Abbildung keine guten Ergebnisse. Die ersten superattraktiven Stellen  $s_i$  ließen sich ohne weiteres bestimmen. Die Abstände  $\Delta s$  werden schon nach den ersten 4 Folgegliedern sehr klein und der Lyapunov Exponent beginnt zu "rauschen". Wir entwickelten einen weiteren Algorithmus, welcher nur marginal bessere Ergebnisse nach  $s_4$  lieferte. Die Idee des Algorithmus war es, zunächst den Lyapunov Exponenten Grob für  $r \in [0, 3.2]$  zu berechnen. Die so entstandene Folge  $l_i = (x_i, l(x_i))$  wird nun iteriert. Sobald  $l_{i,1} < M < 0$  gilt wird  $A = l_{i,0}$  gesetzt. Es wird weiter iteriert bis  $l_{i,1} > 0$  und schließlich wird das Intervall  $[A, l_{i,1}]$  gespeichert. Dies wird bis zum Ende der Folge  $l_i$  ausgeführt. Nun wird der Algorithmus für eine tiefere Schranke auf alle ermittelten Intervalle erneut angewendet. Wir fanden so mehrere Hinweise für die nächsten superattraktiven Stellen. Schließlich wurde  $M \leq 4.0$  und plötzlich fielen keine Punkte außer  $s_1, s_2, s_3, s_4$  in das Raster. Beim betrachten des Lyapunovexponenten fiel auf, dass trotz einer Iterationstiefe von 10000 Iterationen stets galt  $\lambda(x) > -4.0$ . Wir haben dafür nur zwei plausible Erklärungen: (1.)  $s_5 - s_4$  ist sehr klein und wir erwarten, dass  $s_6 - s_5$  noch kleiner ist (untere Abbildung 10): Eventuell reicht unsere numerische Genauigkeit nicht aus. (2.) Der Sinus ist ungenau implementiert und verschmiert so den Grenzwert. Die folgenden superattraktiven Stellen haben wir ermittelt:  $s_1 = 0.0$ ,  $s_2 = 1.570728749999546$   $s_3 = 2.4432446071429674$ ,  $s_4 = 2.6580499999998904$ ,  $s_5 = 2.70599999999986$ ,  $s_6 = 2.71600010303040$  und daraus die ersten 4 Glieder in der Feigenbaum Folge bestimmen:

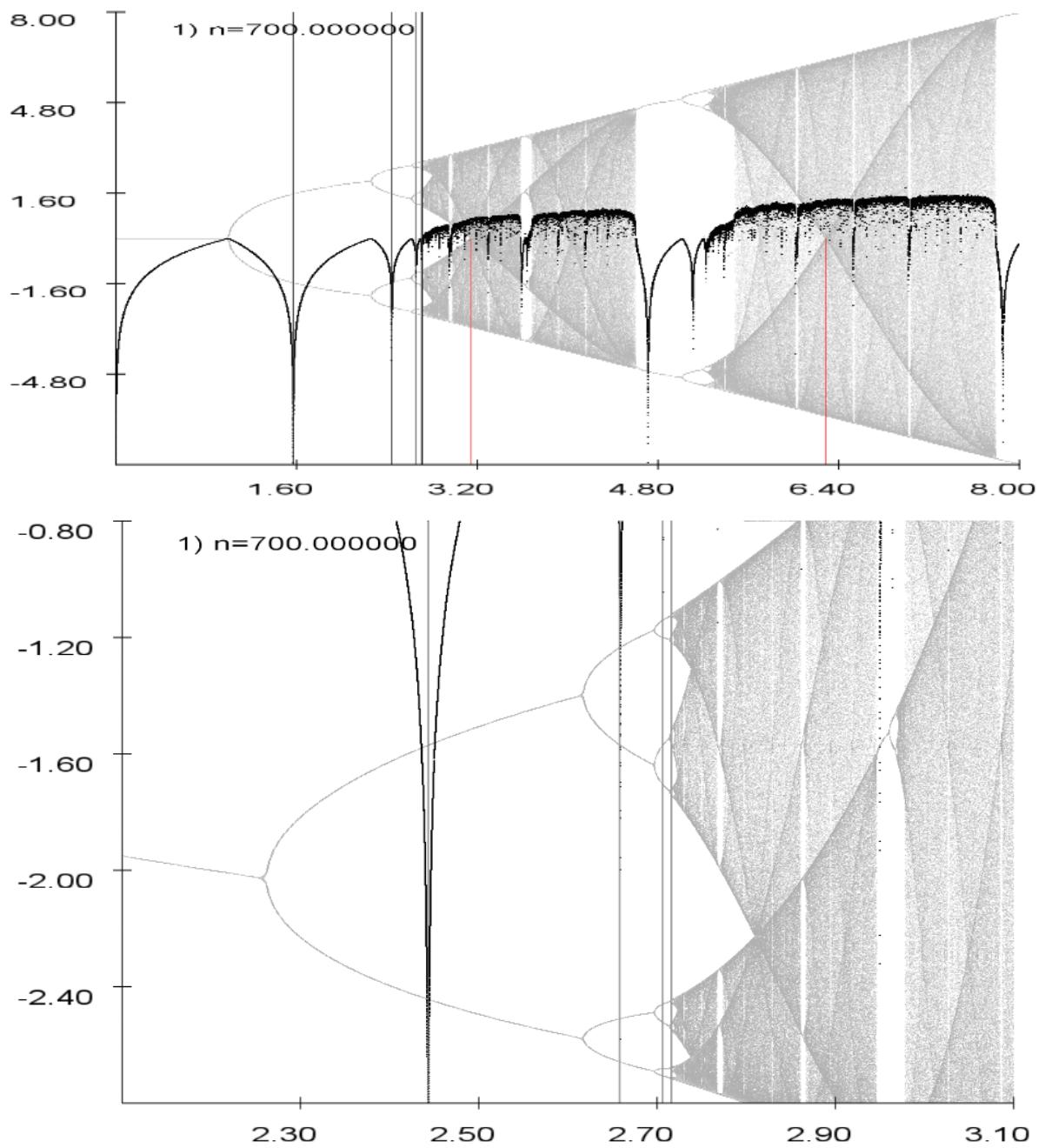
$$\delta_1 = 1.8002294596$$

$$\delta_2 = 4.06188990667$$

$$\delta_3 = 4.47977878743$$

$$\delta_4 = 4.79495059736$$

Anders als die logistische Abbildung bricht das Bifurkationsdiagramm der Sinusfunktion nicht ab. Auch dies lässt sich auf den Kompakten Bildbereich des Sinus zurückführen. Der Lyapunov Exponent weist ab  $r > 8$  eine Perdiodizität von  $\pi$  auf weshalb weitere Betrachtungen für  $r > 8$  nicht zielführend erscheinen.



**Abbildung 10:** Oben: Analyse des Bifurkationsdiagrammes der sinus Abbildung. Eingezeichnet sind die ersten 6 Superattraktiven Stellen (Erste Superattraktive Stelle bei  $r = 0$ ). Ebenfalls wurden  $\pi$  und  $2\pi$  eingezeichnet. Unten: Vergrößerung des Ausschnittes  $r \in [2.1, 3.1]$ ,  $y \in [-2.8, -0.8]$

## 4 Duffing-Gleichung

Wir betrachten nun einen angetrieben und gedämpften Oszillatator. Als Unterschied zum gewöhnlichen Harmonischen Oszillatator wird der Term mit der Federkonstante kubisch. Dies ist vergleichbar mit der Schwingung eines Metallstabes.

$$\ddot{x} + \lambda \dot{x} + \beta x^3 = \epsilon \cos \Omega t \quad (22)$$

Diese DGL lässt sich nun nicht mehr analytisch berechnen. Im Folgenden lösen wir die Gleichung mit der Euler-Methode [2] als auch mit dem Runge-Kutta Verfahren [3]. Um dies zu ermöglichen lässt sich (22) als folgendes DGL System umformen.

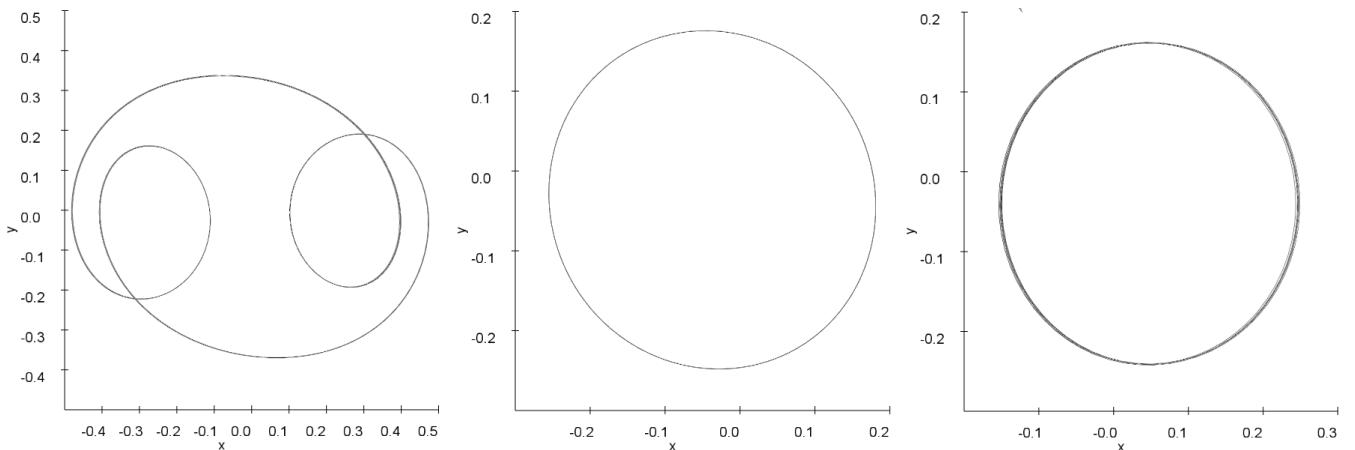
$$\frac{dy}{dt} = \epsilon \cos \theta - \lambda y - \beta x^3 \quad (23)$$

$$\frac{dx}{dt} = y \quad (24)$$

$$\frac{d\theta}{dt} = \Omega \quad (25)$$

### 4.1 Phasenraumdiagramm

Im folgenden verwenden wir als Parameter  $\epsilon = 0,2$ ,  $\lambda = 0,08$ ,  $\beta = 1$  und  $\Omega = 1$  und untersuchen die Unterschiede der verwendeten numerischen Verfahren bei gleich bleibenden Anfangswerten. Dazu wählen wir  $x_0 = 0.21$  und  $y_0 = 0.02$  und betrachten das Phasendiagramm bei unterschiedlichen Schrittgrößen  $h$ . Der zu erwartende Attraktor ist in Abbildung 11 (mitte und rechts) dargestellt. Ist die Schrittgröße zu groß so entsteht bei dem Euler-Cauchy Verfahren ein anderer Attraktor (Abbildung 11 links).

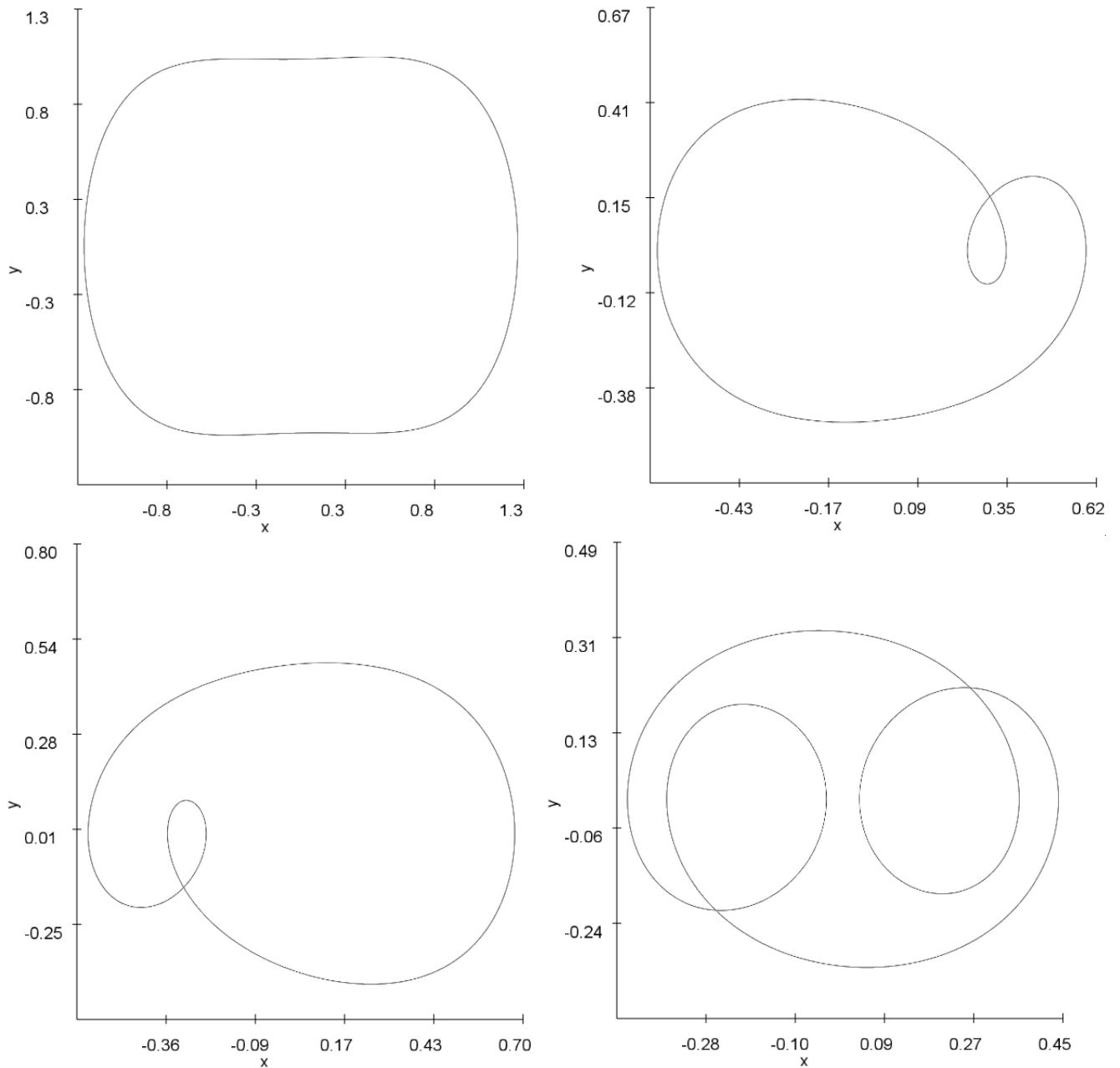


**Abbildung 11:** Phasenraumdiagramm mit Euler-Cauchy. Links: Schrittweite  $h = 0.1$ . Mitte: Schrittweite  $h = 0.08$ . Rechts: Runge-Kutta mit Schrittweite  $h = 0.1$

Im weiteren sind alle folgenden Berechnungen in diesem Kapitel mit dem Runge-Kutta-Verfahren 4ter Ordnung gemacht.

Ganz bedeutend ändert sich der Attraktor, bei Variation der Anfangsbedingungen. Variieren wir  $x_0$  so bleibt der Attraktor identisch bei  $x_0 = 0.18$ , während wir bei  $x_0 = 0.17$  ein Phasenraumdiagramm ähnlich zu Abbildung 11 (links) erhalten.

Abbildung 12 zeigt Attraktoren bei unterschiedlichen Anfangsbedingungen.



**Abbildung 12:** Phasenraumdiagramm mit Runge-Kutta. Links oben:  $x_0 = 1.05$ ,  $y_0 = 0.77$ . Rechts oben:  $x_0 = -0.67$ ,  $y_0 = 0.02$ . Links unten:  $x_0 = -0.46$ ,  $y_0 = 0.30$ . Rechts unten:  $x_0 = -0.43$ ,  $y_0 = 0.12$

## 4.2 Poincareschnitt

Die gezeigten Phasenraumportraits sind Projektionen des dreidimensionalen Phasenraums  $(x, y, \theta)$  auf die  $(x, y)$  Ebene. Der Poincaree Schnitt ist eine Abbildung aller  $(x, y)$  welche eine bestimmte Ebene im Phasenraum schneiden. In Abbildung 13 ist ein Poincareschnitt des Duffing-Oszillators mit  $\theta = 0$  gezeigt. Zur Implementation des Poincaree Schnittes wählten wir  $\theta = 0$  um die Praktikumsanleitung als Test unserer Software nutzen zu können. Dabei nutzten wir  $\sin(\theta_1) * \sin(\theta_2) \leq 0.0 \iff$  Ebenenschnitt:

---

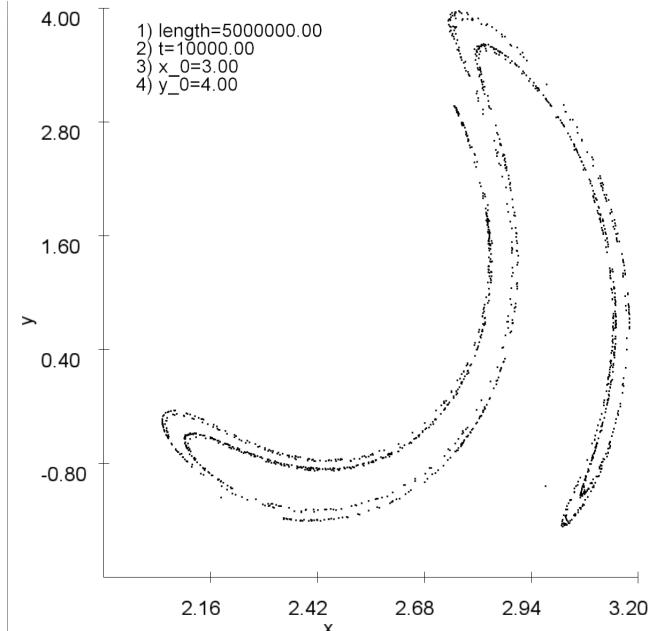
```

pos = sin(theta);
if (last_pos*pos <= 0.0f) {
    result[k*2] = last_x;
    result[k*2+1] = last_y;
    k++;
}
last_pos = pos;

```

---

Dieses Verfahren zeigt bei genauerer Betrachtung aber leichter Ungenauigkeiten. So wird nicht exakt das  $(x, y)$  duplet abgebildet bei welchen die Ebene geschnitten wurden, stattdessen wird das  $(x, y)$  Duplet bei  $\theta_2$  angezeigt. Eine Möglichkeit dies zu Optimieren wäre den Mittelwert  $(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2})$  als Schnittpunkt zu identifizieren.



**Abbildung 13:** Poincaré-Schnitt des Duffing Oszillators für  $\epsilon = 7.72$ ,  $x = 3.0, y = 4.0$ ,  $\lambda = 0.2$ ,  $\beta = 1$ ,  $\theta = 1$ . als Referenz aus dem VORBEREITUNGSSHEFT-LITERATUR-S38

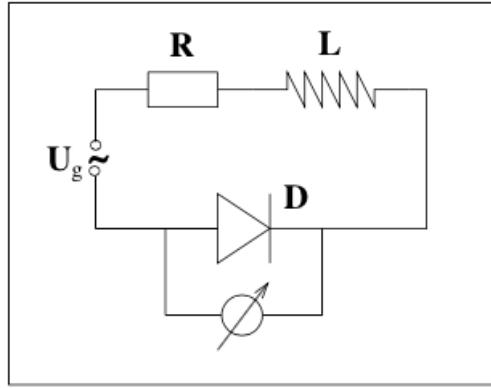
## 5 LDR-Oszillator

Im folgenden untersuchen wir einen realen nichtlinearen Schwingkreis. Dieser wurde realisiert indem bei einem LCR-Schwingkreis der Kondensator durch eine Diode ausgetauscht wurde (Abbildung 14)

### 5.1 Theorie

Zunächst lässt sich ein linearer Schwingkreis der aus einem Widerstand, einem Kondensator und einer Spule besteht über die Spannungen an den einzelnen Bauteilen beschreiben

$$V_g = V_c + V_l + V_r \text{ (Kirschoff'sches Gesetz)}$$



**Abbildung 14:** Schaltkizze für nichtlinearen Schwingkreis. Quelle: Versuchsanleitung, S. 43, Uni Hamburg

Daraus folgt mit  $V_g = V_s \cos \omega t$  die Differentialgleichung

$$\ddot{Q}L + \dot{Q}R + \frac{Q}{C} = V_s \cos \omega t$$

welche mit dem angetriebenen, gedämpften Oszillator vergleichbar ist. Dabei ist  $Q$  die Ladung,  $L$  die Induktivität der Spule,  $R$  der Widerstand,  $C$  die Kapazität des Kondensators  $V_s$ , die angelegte Spannung und  $\omega$  die Frequenz der angelegten Wechselspannung.

Wird nun der Kondensator durch eine Diode ausgetauscht erhalten wir, wie bei dem Duffing-Oszillator eine Gleichung die nicht mehr analytisch lösbar ist. Während beim linearen Schwingkreis  $I = \frac{dQ}{dt}$  gilt lässt sich die Diode als parallel geschalteter Kondensator und Widerstand mit

$$I = I_f(1 - \exp(-\frac{V_d}{V_t})) + \frac{dQ}{dt} \quad (26)$$

beschreiben, wobei  $I_f$  und  $V_t$  Konstanten sind. Dabei beschreibt der erste Summand den Strom durch den Widerstand und der zweite die Änderung der Ladung aufgrund der Diodenkapazität. Dies führt bei einer Kapazität in Durchlassrichtung von  $C = C_f \exp(-\frac{V_d}{V_t})$  zu

$$\frac{dV_d}{dt} = \frac{I - I_f(1 - \exp(-\frac{V_d}{V_t}))}{C_f \exp(-\frac{V_d}{V_t})} \quad (27)$$

und bei einer Kapazität in Sperrrichtung von  $C = C_r(1 + \frac{V_d}{\phi})^{-\gamma}$  zu

$$\frac{dV_d}{dt} = \frac{I - I_f(1 - \exp(-\frac{V_d}{V_t}))}{C_r(1 + \frac{V_d}{\phi})^{\gamma}} \quad (28)$$

wobei  $\phi$  und  $\gamma$  Konstanten sind. Außerdem gilt

$$\frac{dI}{dt} = \frac{V_s \cos \theta - V_d - RI}{L} \quad (29)$$

$$\frac{d\theta}{dt} = \omega \quad (30)$$

Diese Gleichungen lassen sich numerisch lösen.

## 5.2 Numerische Berechnungen

Unter Verwendung der Parameter:

$R$	$100\Omega$	Widerstand
$L$	$2367 \cdot 10^{-6}H$	Spule
$C_r$	$82pF$	Konstante proportional zur Kapazität der Diode in Sperrrichtung
$C_f$	$56 \cdot 10^{-6}pF$	Konstante proportional zur Kapazität der Diode in Durchlassrichtung
$I_f$	$2,8pA$	Konstante proportional zum Strom
$\gamma$	$0,44$	Konstante
$\phi$	$0,6V$	Konstante
$V_t$	$34mV$	Konstante

haben wir für unterschiedliche Anregungsspannungen  $V_s$  numerisch die Gleichungen aus 5.1 gelöst. Dabei sind wir davon ausgegangen, dass für  $V_d > -0.6V$  die Diode sperrt.

Um sinnvolle Lösungen zu erhalten muss für die Schrittweite  $h \geq 10^{-8}$  gelten. Diese oszillierenden Lösungen für den Strom und die Spannung über der Diode sind in Abbildung 15 dargestellt. Strom und Spannung oszillieren mit gleicher Phase und Pendeln sich nach gewisser Zeit unabhängig von den Anfangsbedingungen ein.

### 5.2.1 Phasenraumdiagramme

Bei einer Anregungsspannung  $V_s < 0.001V$  zeigt sich das Verhalten im Grenzfall des gedämpften Oszillators (siehe Abbildung 16). Eindeutige Attraktoren erhalten wir bei Spannungen ab  $V_s = 1V$  (Abbildung 17). Dabei ist bei  $V_s = 1V$  (links) die Trajektorie wohl definiert, während bei  $V_s = 1.2V$  (mitte) sie sich langsam aufspaltet und bei  $V_s = 1.42V$  (rechts) auf 2 Pfaden durch den Phasenraum verläuft. Je höher  $V_s$  wird, desto stärker gehen die Trajektorien auseinander bis sie bei ca.  $V_s = 20V$  ein Maximum erreichen (Abbildung 18 links) und anschließend wieder zusammen gehen und chaotisches Verhalten zeigen (Abbildung 18 mitte und rechts).

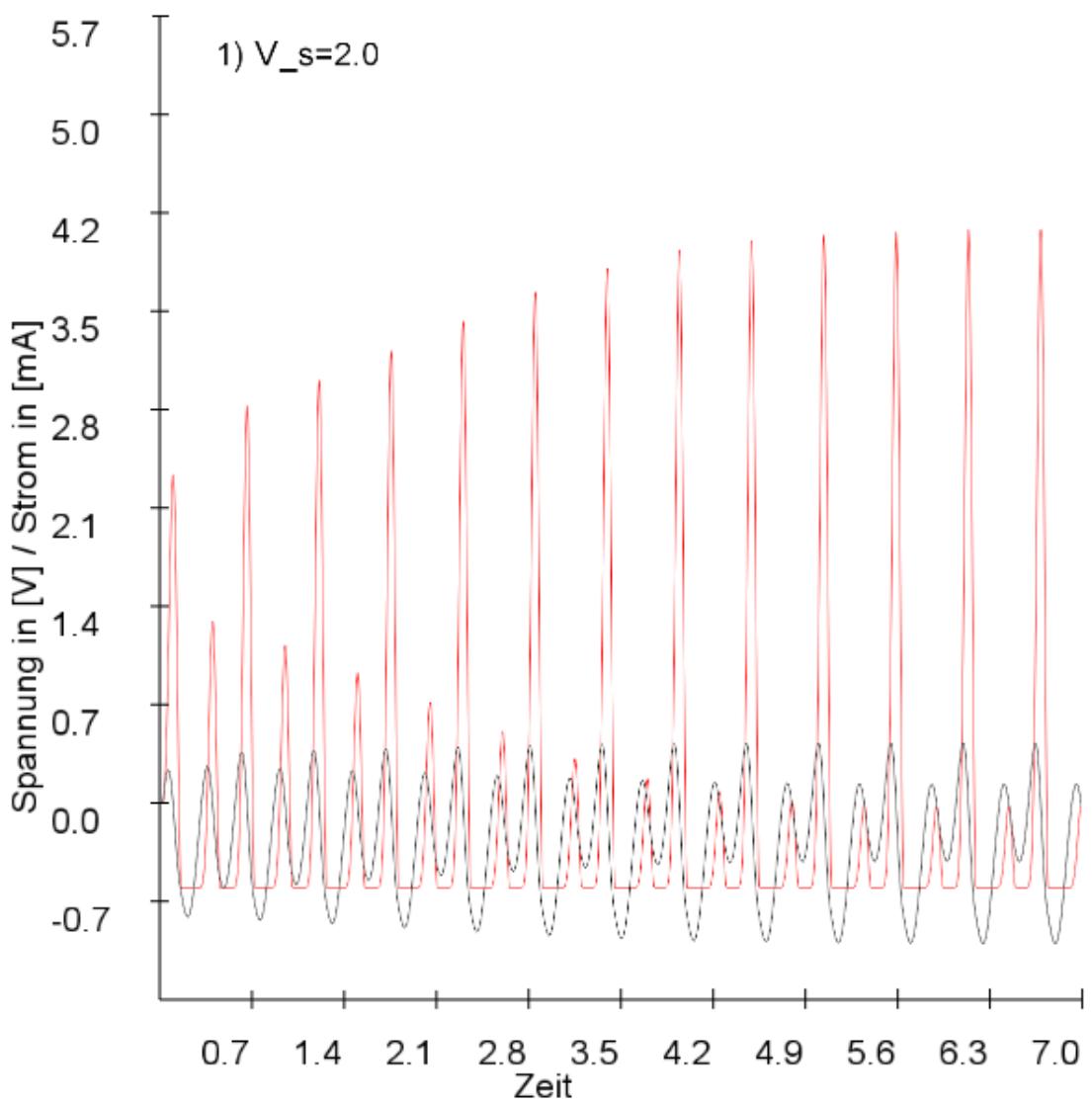
Nach Änderung des Zeitschrittes  $h$  auf  $h = 10^{-8}$  haben wir sehr unterschiedliche Beobachtungen gemacht. Beispielsweise erhalten wir bei  $V_s = 9.1300024986267072V$  einen eindeutigen Attraktor (Abbildung 19 links) und Chaos bei einem Wert  $10^{-15}V$  größer (rechts). Unter Verwendung des Euler-Cauchy-Verfahrens beobachten wir teilweise ein anderes Verhalten. So können wir im Bereich von  $V_s = 3V$  bis  $V_s = 4.7V$  mehrere Periodenverdopplungen beobachten (Abbildung 20 und 21) bis hin zum Chaos.

### 5.2.2 Poincaré-Schnitt

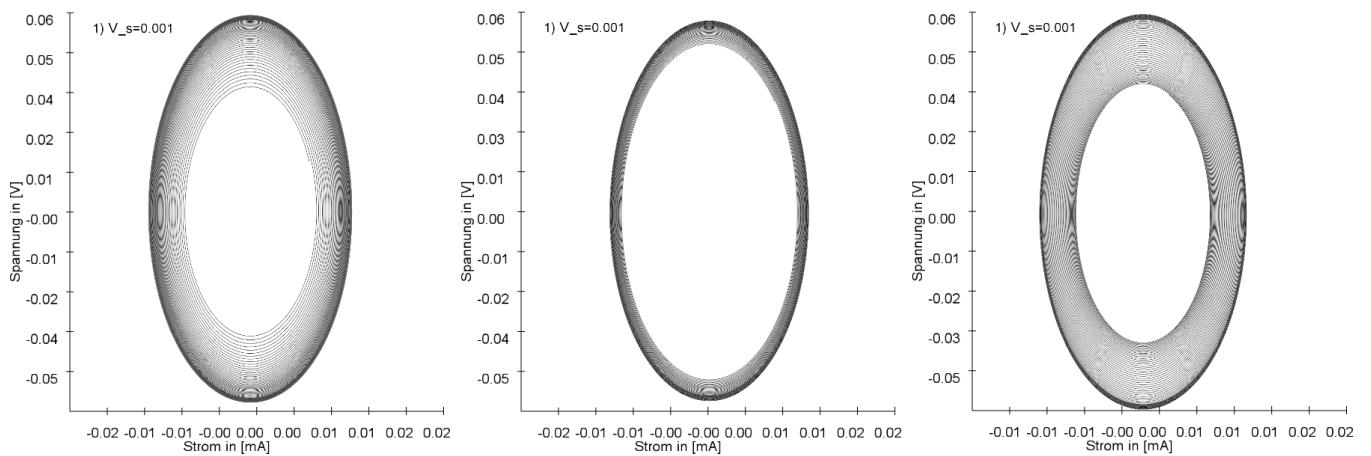
Bei der Betrachtung des Poincaré-Schnittes fällt auf, dass bei wohldefinierten Attraktoren die Phasenraumtrajektorie immer bei ähnlichen Spannungen durch die  $\theta = 0$  Ebene verläuft (vgl. Abbildung 22). Im chaotischen Fall weißt der Poincaré-Schnitt weitere Punkte bei unterschiedlichen Spannungen auf (vgl. Abbildung 23). Anmerkung: Bei der numerischen Berechnung kommt es immer wieder vor, dass bei ganz bestimmten  $V_s$  der Open-GL Buffer auf der Grafikkarte überläuft.

### 5.2.3 Bifurkationsdiagramm-Diagramm

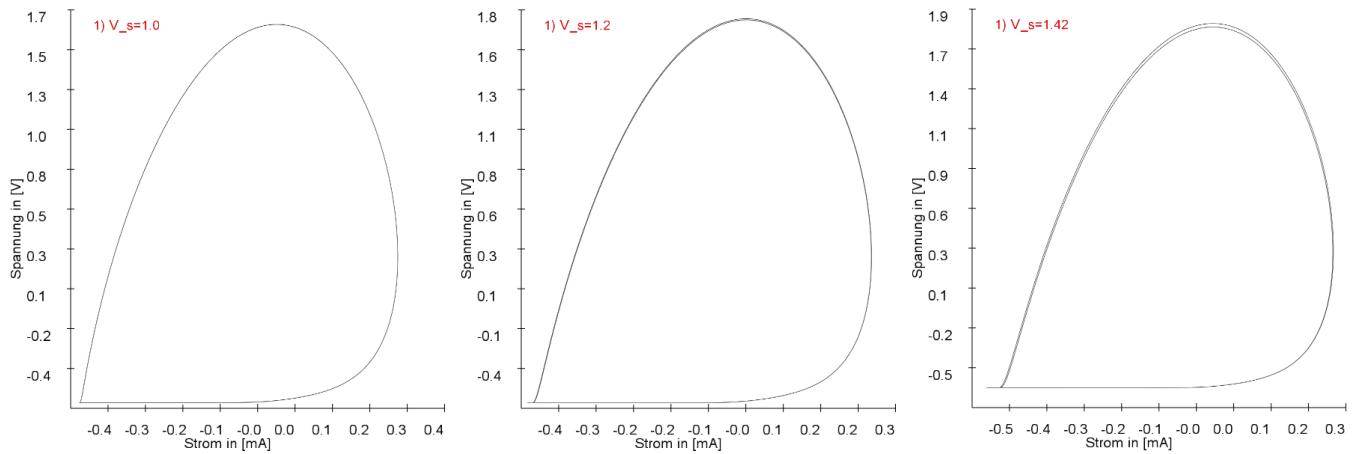
Um den Bereich von Chaos und Ordnung zu visualisieren, haben wir im folgenden separat für Strom und Spannung das Bifurkationsdiagramm geplottet. Dabei ist der variierende Parameter die Eingangsspannung  $V_s$ . Bei  $V_s \approx 1.4$  beobachten wir sowohl beim Strom als auch bei der Spannung eine Periodenverdopplung. Dies passt mit unserer Beobachtung des Attraktors bei  $V_s = 1.42$  (Abbildung 17 rechts) zusammen. Auch das Chaotische Verhalten bei  $V_s = 7$  (Abbildung fig:ldr-poin2 links) finden wir im Bifurkationsdiagramm wieder.



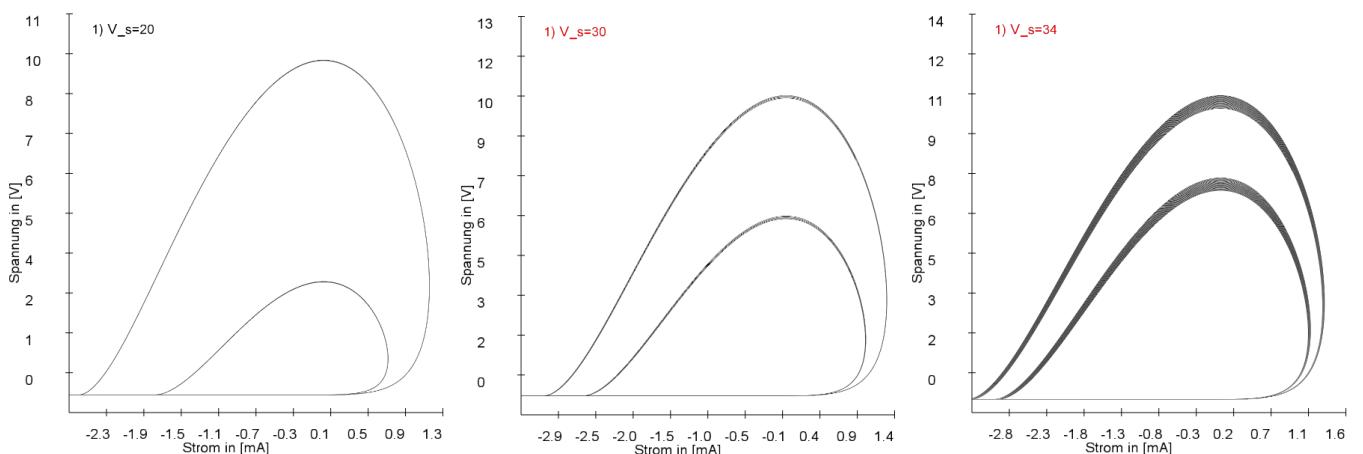
**Abbildung 15:** Oszillation von Strom (schwarz) und Spannung (rot) bei  $V_s = 2V$  mit Runge-Kutta-Verfahren 4ter Ordnung



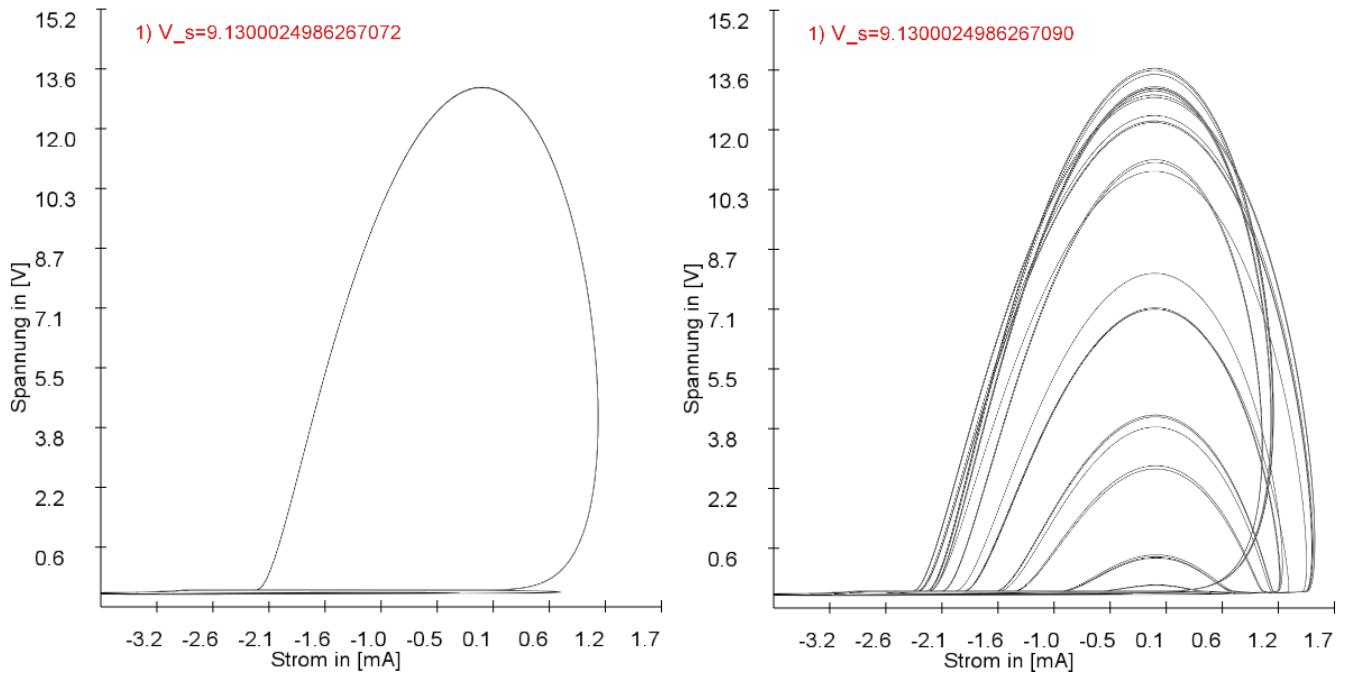
**Abbildung 16:** Phasenraumdiagramm mit Runge-Kutta 4ter Ordnung für  $V_s = 0.001V$  mit  $2 \cdot 10^5$  Zeitschritte bei einer Schrittweite von  $h = 10^{-9}$ . Links: Nach  $5 \cdot 10^4$  Zeitschritten Einschwingzeit, Mitte: Nach  $10^2$  Zeitschritten Einschwingzeit. Rechts: Nach  $2 \cdot 10^5$  Zeitschritten Einschwingzeit



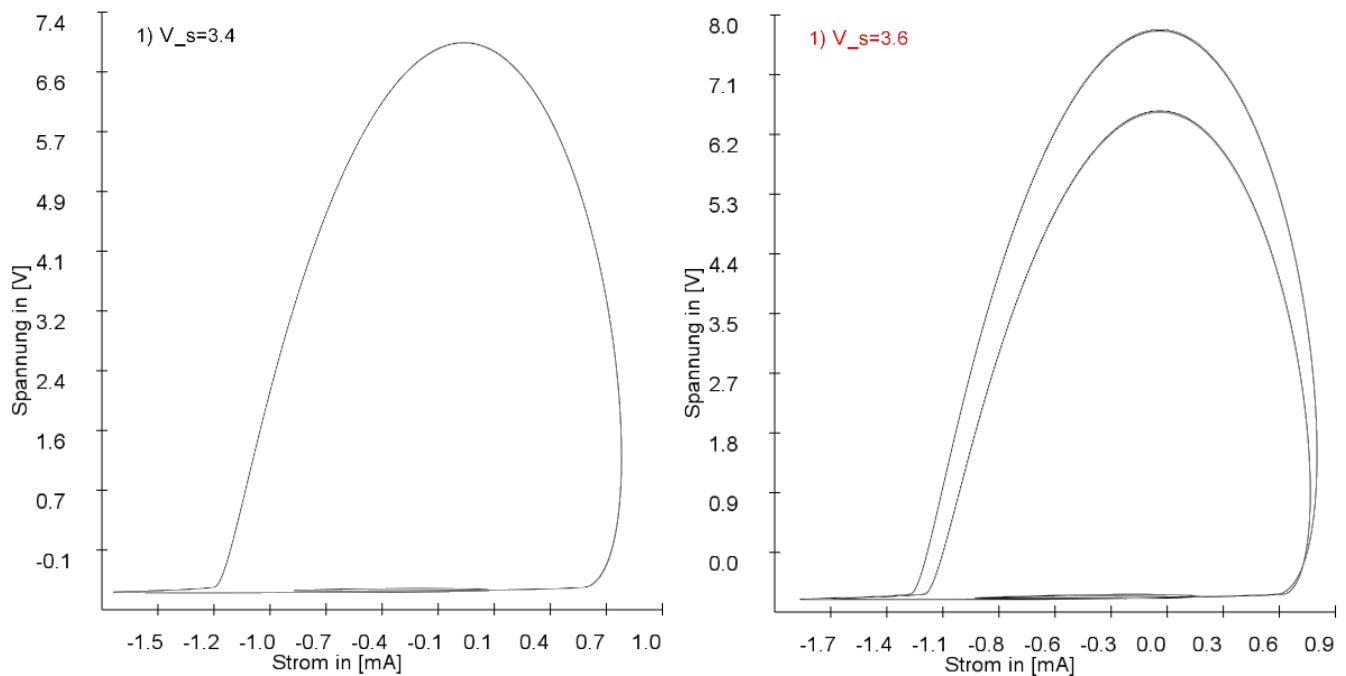
**Abbildung 17:** Phasenraumdiagramm mit Runge-Kutta 4ter Ordnung für  $2 \cdot 10^4$  Zeitschritte nach einer Einschwingzeit von  $3 \cdot 10^5$  Zeitschritten und einer Schrittweite von  $h = 10^{-9}$ . Links:  $V_s = 1V$ , Mitte:  $V_s = 1.2V$ . Rechts:  $V_s = 1.42V$



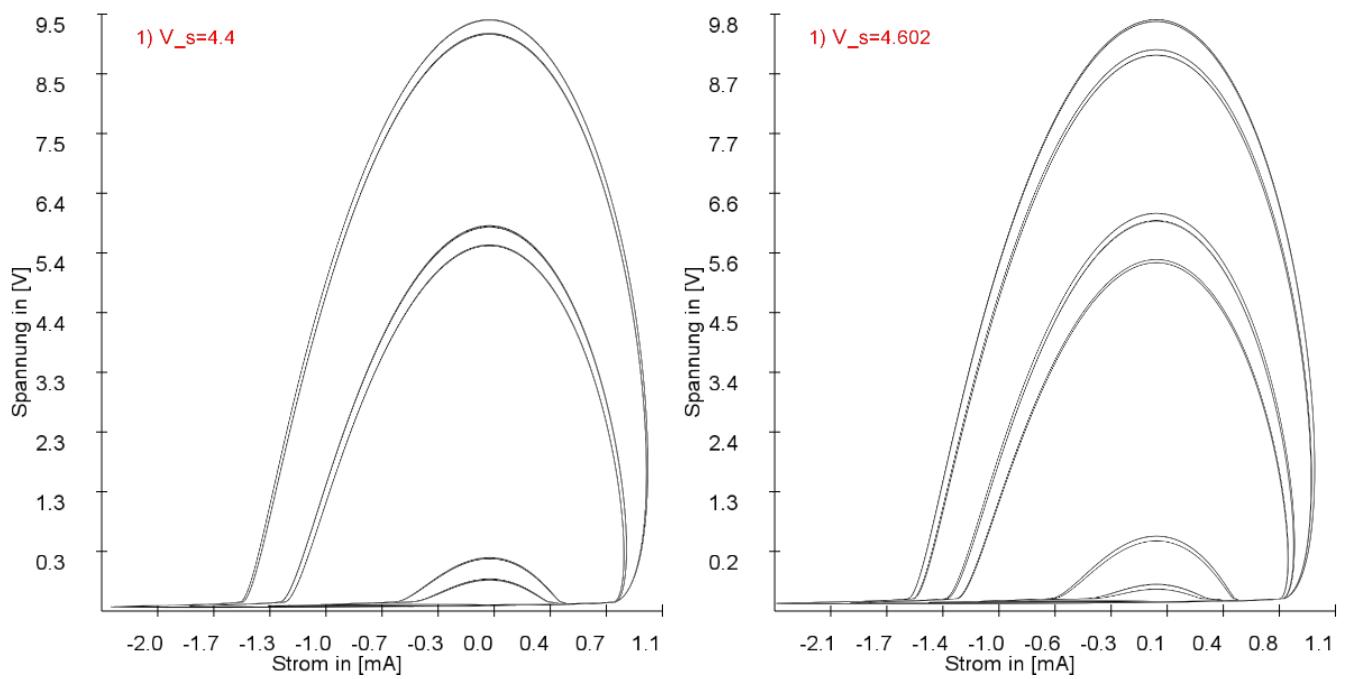
**Abbildung 18:** Phasenraumdiagramm mit Runge-Kutta 4ter Ordnung nach einer Einschwingzeit von  $3 \cdot 10^5$  Zeitschritten und einer Schrittweite von  $h = 10^{-9}$ . Links:  $V_s = 20V$  für  $2 \cdot 10^4$  Zeitschritte, Mitte:  $V_s = 30V$  für  $2 \cdot 10^4$  Zeitschritte. Rechts:  $V_s = 34V$  für  $1.2 \cdot 10^5$  Zeitschritte



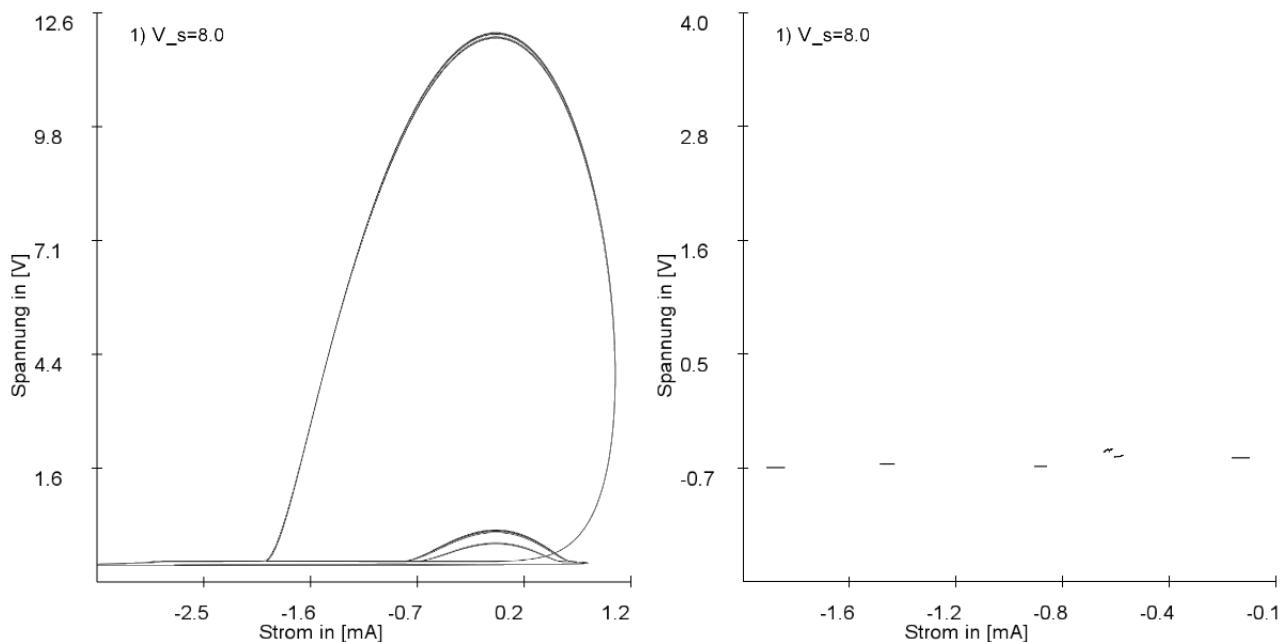
**Abbildung 19:** Phasenraumdiagramm mit Runge-Kutta 4ter Ordnung nach einer Einschwingzeit von  $3 \cdot 10^5$  Zeitschritten für weitere  $5 \cdot 10^4$  Zeitschritte bei einer Schrittweite von  $h = 10^{-8}$ . Links: Wohldefinierter Attraktor. Rechts: Chaos



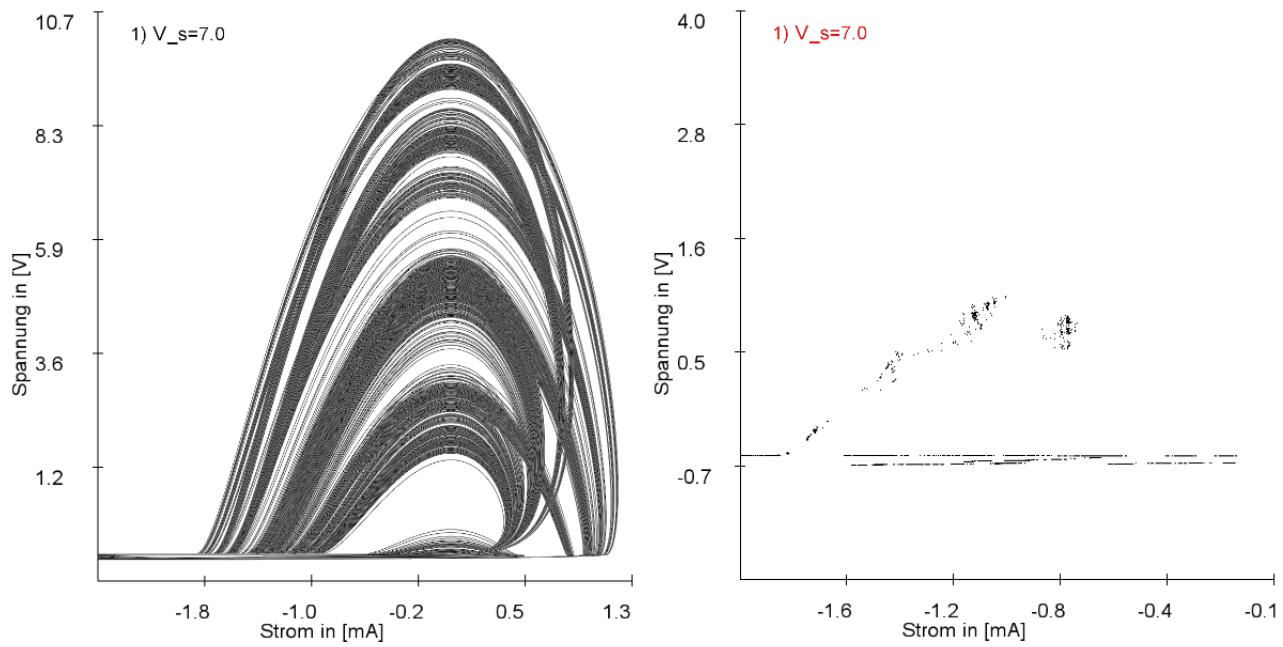
**Abbildung 20:** Phasenraumdiagramm mit Euler-Cauchy-Verfahren nach Einschwingzeit von  $2.5 \cdot 10^6$  Zeitschritten für weitere  $10^5$  Zeitschritte. Links:  $V_s = 3.4$  V, Rechts:  $V_s = 3.6$  V



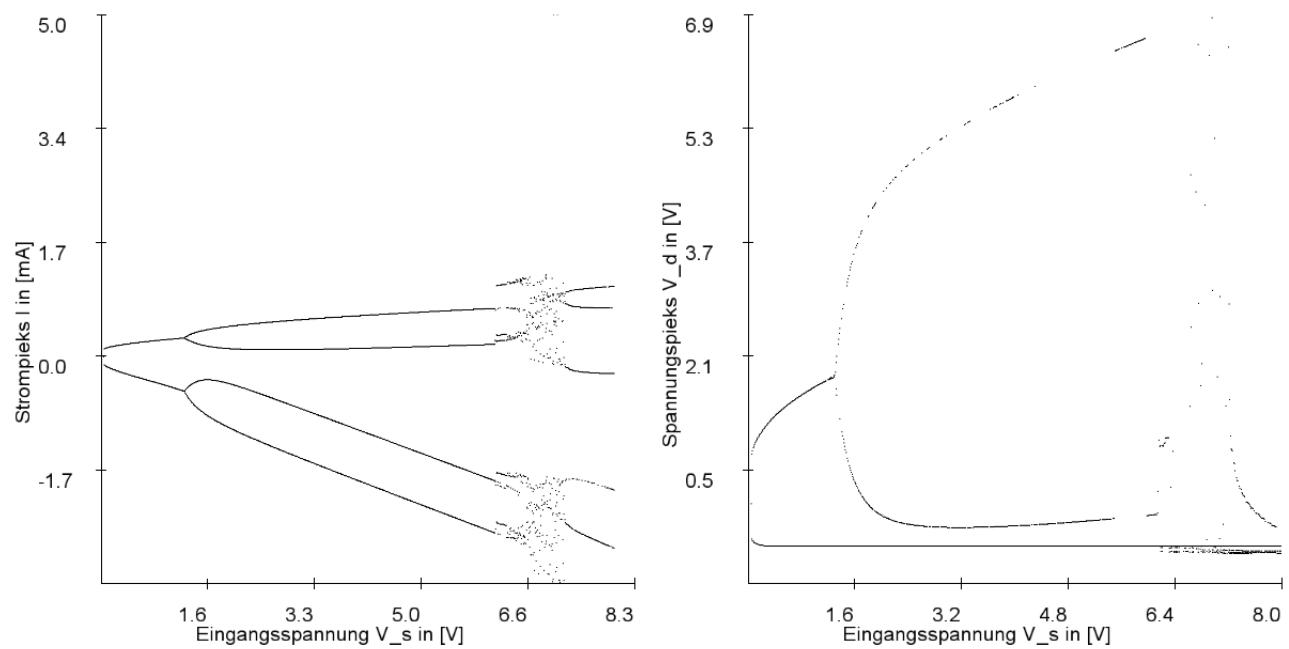
**Abbildung 21:** Phasenraumdiagramm mit Euler-Cauchy-Verfahren nach Einschwingzeit von  $2.5 \cdot 10^6$  Zeitschritte für weitere  $10^5$  Zeitschritte. Links:  $V_s = 4.4V$ , Rechts:  $V_s = 4.602V$



**Abbildung 22:**  $5 \cdot 10^5$  Zeitschritte nach einer Einschwingzeit von  $\cdot 10^4$  Zeitschritten bei  $V_s = 8V$ . Links: Phasenraumdiagramm. Rechts: Poincaré-Schnitt durch  $\theta = 0$  Ebene



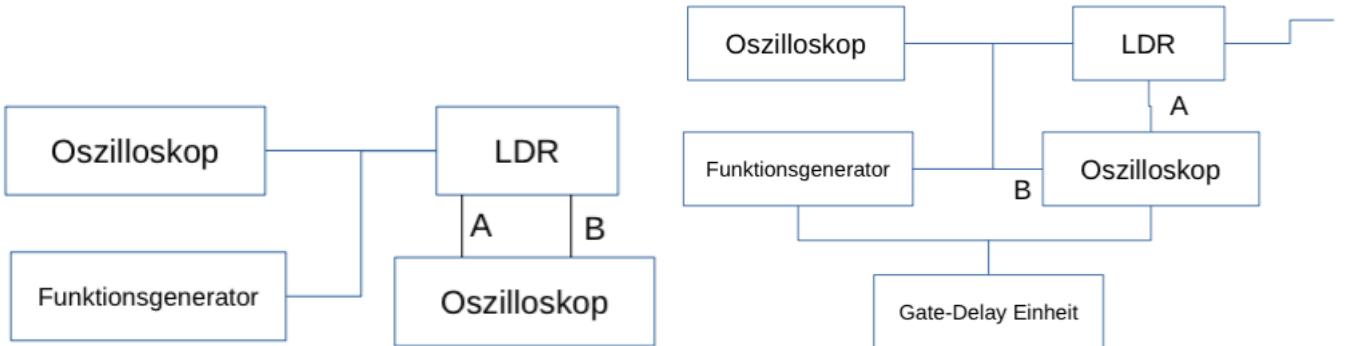
**Abbildung 23:**  $5 \cdot 10^5$  Zeitschritte nach einer Einschwingzeit von  $\cdot 10^4$  Zeitschritten bei  $V_s = 7V$ . Links: Phasenraumdiagramm. Rechts: Poincaré-Schnitt durch  $\theta = 0$  Ebene



**Abbildung 24:** Bifurkationsdiagramm mit  $4 \cdot 10^5$  Iterationen pro  $V_s$ . Von  $V_s = 0$  bis  $V_s = 8$  in 0.01 Schritten. Links: Strom. Rechts: Spannung

### 5.3 Versuchsaufbau

Für die Messung an einem realen LDR-Schwingkreis haben wir zunächst einen Messaufbau wie in Abbildung 5.5 (links) verwendet. Über die Eingänge A und B des Oszilloskop konnten wir Strom und Spannung über der Diode einzeln und als Phasendiagramm darstellen. Ferner haben wir einen Gate-Delay Generator verwendet der 5.5 (rechts) zusehen ist. TODO: Abbildungen noch nicht korrekt. Die Parameter des



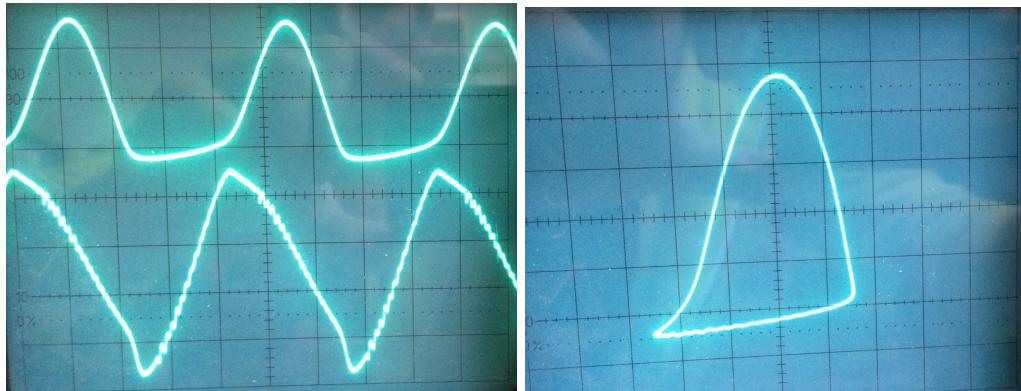
**Abbildung 25:** Messaufbau zur Visualisierung des LDR-Schwingkreises. Oszilloskop links zur Überprüfung der Eingangsspannung und Frequenz. Links: Strom und Spannung über der Diode werden dargestellt auf Oszilloskop über Eingänge A und B. Rechts: Gate-Delay Einheit verzögert Signal in Abhängigkeit der jetzt variierenden Eingangsspannung  $V_s$  zur Darstellung des Stroms bzw. der Spannung in Abhängigkeit von  $V_s$

Schwingkreises unterscheiden sich von den Paramtern, die bei der numerischen Berechnung verwendet wurden. So hatten wir eine gesamt Induktivität von  $L \approx 370 \cdot 10^{-6} H$  was bei einer Resonanzfrequenz von  $w \approx 800 kHz$  zu  $C_r \approx 4.2 \cdot 10^{-9} F$  führt.

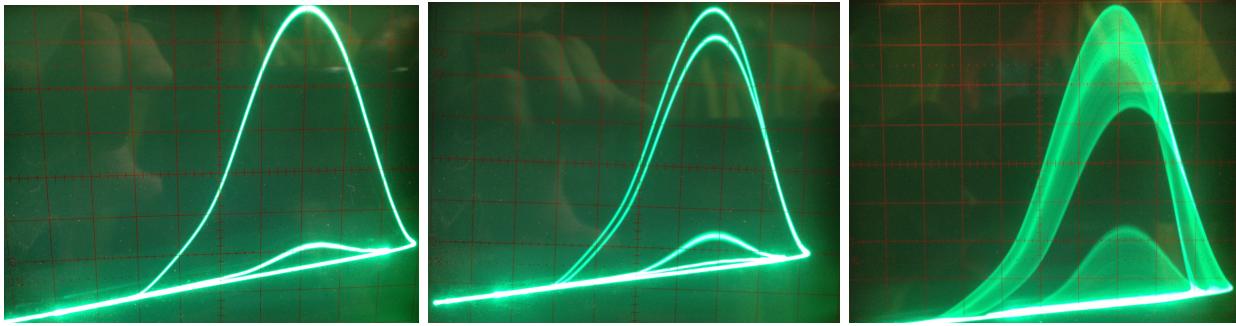
### 5.4 Versuchsdurchführung

#### 5.4.1 Oszillation und Phasenraumdiagramm

Zunächst haben wir bei einer Eingangsspannung von  $V_s = 0.128V$  die Resonanzfrequenz über die Betrachtung des Phasenraumdiagramms auf  $\omega = 800kHz$  bestimmt. Der aufgebaute Schaltkreis hatte nicht die selbe Konfiguration wie unsere numerischen Modelle, dennoch erhalten wir einen sehr ähnlichen Verlauf. In Abbildung 26 sind die Oszillationen von Strom und Spannung (links) und das entsprechende Phasenraumdiagramm dargestellt (rechts) welches dem numerischen Ergebniss in 20 (links) sehr ähnlich sieht.



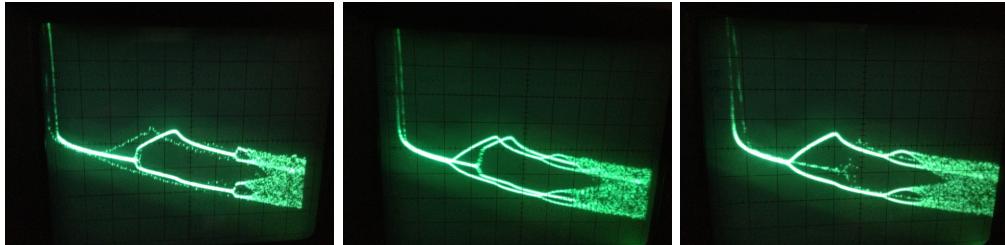
**Abbildung 26:** Eingangsspannung  $V_s = 5V$  und Resonanzfrequenz von  $\omega = 800kHz$ . Links: Zeitlicher Verlauf von Spannung (oben) und Strom (unten). Rechts: Phasenraumdiagramm



**Abbildung 27:** Eingangsspannung  $V_s = 5V$  und Resonanzfrequenz von  $\omega = 800kHz$ . Links: Zeitlicher Verlauf von Spannung (oben) und Strom (unten). Rechts: Phasenraumdiagramm

## 5.5 Bifurkationsdiagramm

Beim erzeugen des Bifurkationsdiagrammes haben wir einen anderen Schaltkreis verwendet, dessen Resonanzfrequenz wir auf  $\omega = 325kHz$  bestimmt haben. Der Versuchsaufbau wurde nun mit einer Gate-Delay Einheit aufgebaut (Abbildung ). Wir modellierten über den Aux-Out des Funktionsgenerator eine Sägezahnspannung welche später den Definitionsbereich des Kontrollparameters  $V_s$  darstellt. Zunächst ist dies nicht gelungen. Mit Hilfe des digitalen Oszilloskopes konnten wir dann aber den Verlauf so modellieren, dass  $V_s = 0V \dots V_s = V_{Ausgang}$  linear periodisch durchlaufen wird. Wir stellten fest, dass das Bifurkationsdiagramm am besten am Oszilloskop zu sehen war, wenn wir den Definitionsbereich von oben nach unten ( $V_s = V_{Ausgang} \dots V_s = 0V$ ) eingestellt haben (Abbildung 28). In dieser Einstellung gab es zwar noch Artefakte von der Ungenauigkeit des Funktionsgenerators (Abbildung 28 rechts) aber die Bifurkationspunkte waren nun präziser. Wir erzeugten das Bifurkationsdiagramm zusätzlich mit einer Dreiecksspannung. Wie zu erwarten ergab sich so eine Überlagerung aus den beiden Sägezahnspannungen.



**Abbildung 28:** Bifurkationsdiagramme mit verschiedenen Modulationen des Kontrollparameters  $V_s$ : Links Sägezahnspannung  $V_s = 0V \dots V_s = V_{Ausgang}$ , Mitte: Dreiecksspannung, rechts: Sägezahnspannung  $V_s = V_{Ausgang} \dots V_s = 0V$

Wir fragten uns, wieso es zu einer Spitze im oberen Ast nach der ersten Bifurkation kommt. Dabei zeigte sich, dass die Frequenz nicht genau auf  $\omega = 325kHz$  eingestellt war. Als wir dies korrigierten erhielten wir ein glatteres Bifurkationsdiagramm (Abbildung N)

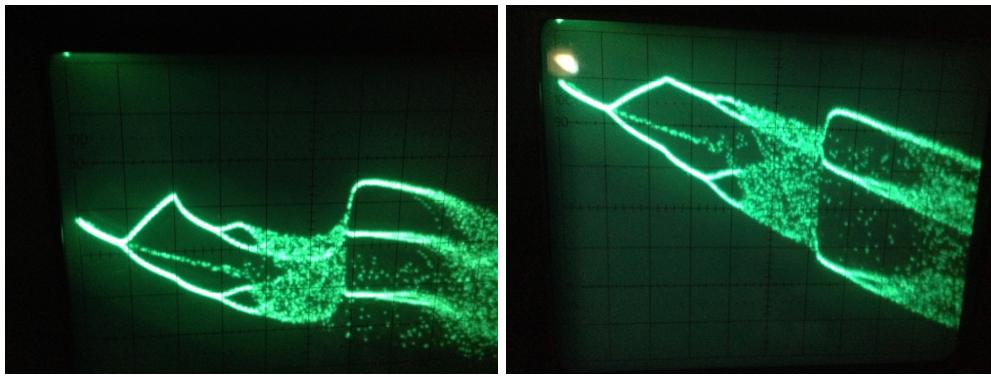
## 5.6 Vermessung der Bifurkationspunkte

Zuletzt haben wir noch die Bifurkationspunkte vermessen. Dazu haben wir die Amplitude der Sägezahnspannung minimiert, so dass  $V_s = V_{Ausgang}$  konstant war. Bei geringer Spannung wurde auf dem Oszilloskop nur ein Punkt abgebildet. Beim durchlaufen der Spannung entstanden so sprungartig zwei Punkte und schließlich 4 Punkte. Die jeweiligen  $V_{Bifurkation,i}$  konnten wir somit als Bifurkationspunkte identifizieren:

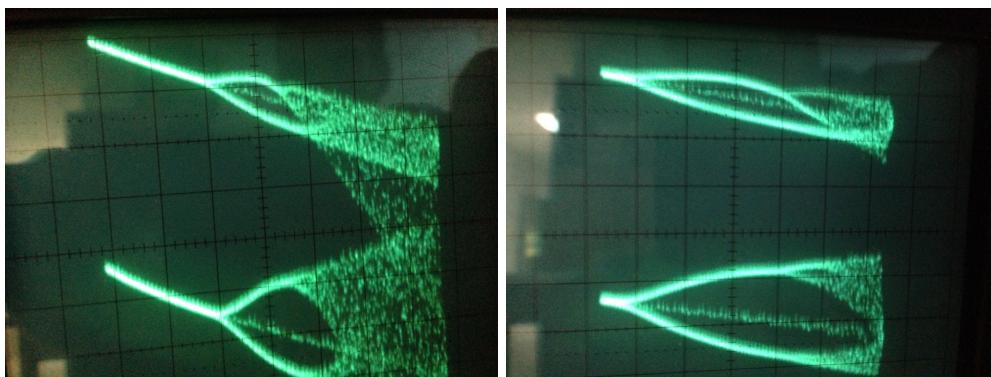
$$V_{Bifurkations,1} = 6.6V \pm 0.2V$$

$$V_{Bifurkations,2} = 13.6V \pm 0.2V$$

In Abbildung ?? haben wir noch ein mal mit der Sägezahnspannung die Bifurkationspunkte vergrößert aufgenommen. Es gelang uns nicht den Bifurkationspunkt  $Bifurkations,3$  zu bestimmen weshalb wir die Feigenbaumkonstante nicht aus dem Experiment ableiten konnten.



**Abbildung 29:** Eingangsspannung  $V_s = 5V$  und Resonanzfrequenz von  $\omega = 800kHz$ . Links: Zeitlicher Verlauf von Spannung (oben) und Strom (unten). Rechts: Phasenraumdiagramm



**Abbildung 30:** Vergrößerung der ersten beiden Bifurkationspunkte.

## 5.7 Zusammenfassung

## 6 Literatur

### Literatur

- [1] OESIS. *A006890 Decimal expansion of Feigenbaum bifurcation velocity.* @ONLINE. URL: <https://oeis.org/A006890>.
- [2] Explizites Euler-Verfahren Wikipedia. *Explizites Euler-Verfahren* @ONLINE. URL: [https://de.wikipedia.org/wiki/Explizites\\_Euler-Verfahren](https://de.wikipedia.org/wiki/Explizites_Euler-Verfahren).
- [3] Runge-Kutta-Verfahren Wikipedia. *Runge-Kutta-Verfahren* @ONLINE. URL: <https://de.wikipedia.org/wiki/Runge-Kutta-Verfahren>.
- Nichtlineare Dynamik und Chaos - Physikalisches Praktikum für Fortgeschrittene Universität Hamburg