

# Newton Fraktale

Yaroslav Nalivayko

**Betreuer:** M.Sc. Benjamin Maier

**Zusammenfassung** Newton Fraktale sind eine Teilmenge der mathematischen Fraktale, die durch Benutzung des Newton Verfahrens für Lösung von nichtlinearen Gleichungen auf Komplexe Ebene erscheinen.

## 1 Einleitung

Newton Fraktale stellen eine interessante Klasse der mathematischen Fraktalen dar. Im Rahmen dieser Arbeit werden Theoretische Grundlagen vorgestellt und ein Programm für die Visualisierung der Fraktalen entwickelt. In letzte Sektion findet Analyse mancher interessanten Funktionen statt.

## 2 Theoretische Grundlagen

### 2.1 Numerische Mathematik

Die numerische Mathematik beschäftigt sich als Teilgebiet der Mathematik mit der Konstruktion und Analyse von Algorithmen für kontinuierliche mathematische Probleme. [Tre92] Oft wird benutzt, um approximative Lösungen mit Hilfe von Computer zu finden.

### 2.2 Newton Verfahren

Newton Verfahren ist das iterative numerische Verfahren, das eine Wurzel gegebener Funktion findet. Die Methode ist nach Sir Isaac Newton benannt.

Wir interessieren uns in stetig differenzierbaren Funktionen mit nur einer Variable.

$$f(x) = 0$$

Man soll manuell den Startwert  $x_0$  wählen und dann die iterative Methode benutzen, bis akzeptable Lösung gefunden wird.

$$x_{n+1} = x_n + \frac{f(x_n)}{f'(x_n)}$$

Gewöhnlich wählt man eine zulässige Abweichung  $\varepsilon$  und eine maximale Anzahl der Schritte  $N$ . Nach jedem Schritt der iterative Methode prüft man. Falls  $f(x_n) < \varepsilon$ , dann ist die Lösung gefunden. Und falls  $n > N$ , dann ist die Lösung unauffindbar in akzeptable Anzahl der Schritte. Iterative Prozess der Annäherung von  $x_n$  zu  $x_n + 1$  heißt die Konvergenz, und falls  $x_0$  zu  $x_n$  kommt, heißt das, dass  $x_0$  gegen  $x_n$  konvergiert.

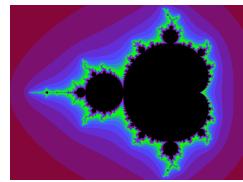
## 2.3 Fraktale

Fraktal (lateinisch *fractus* - gebrochen) ist ein von Mathematiker Benoît Mandelbrot geprägter Begriff, der bestimmte natürliche oder künstliche Gebilde oder Muster bezeichnet. Diese Gebilde oder Muster weisen einen hohen Grad von Skaleninvarianz bzw. Selbstähnlichkeit. Das ist beispielsweise der Fall, wenn ein Objekt aus mehreren verkleinerten Kopien seiner selbst besteht. [fra]

Figure 1a stellt ein Beispiel für ein künstliches Fraktal, und Figure 1b für ein mathematisches Fraktal.



(a) Romanesco. Ein natürliches Fraktal. [Mak]



(b) Mandelbrot. Ein künstliches Fraktal. [Bey]

## 2.4 Newton Fraktale

Diese Fraktale erscheinen sich, wenn man das Newton Verfahren für Auffinden der Wurzeln der nichtlinearen Gleichungen auf Komplexe Ebene benutzt. Genauer gesagt, soll man die Wurzel für jeden Punkt des gesuchten Bildes mit Newton Verfahren finden.

Zum Beispiel nehmen wir die Funktion  $f(x) = x^3 - 1$ . Diese Funktion hat drei Lösungen auf Komplexe Ebene:  $1, -\sqrt[3]{-1}$  und  $(-1)^{2/3}$ . Näherungswerte in Koordinatenform sind  $(1, 0)$ ,  $(-0.5, 0.866)$  und  $(-0.5, -0.866)$ . Die Punkte des Bildes, die sich durch das Newton Verfahren zu entsprechenden Wurzeln annähern, werden entsprechend mit rot, blau und grün gefärbt. Das Fraktal auf dem Bild 2 repräsentiert die gewählte Umgebung. Die Ursachen zu dieser komischen Mischung werden in der Sektion 4 erläutert.

## 3 Visualisierung

In diesem Abschnitt wird ein Programm entwickelt, das Bilder der Newton Fraktale generiert. Außerdem es wird ein Beispiel vorgestellt, wie man Animationen erzeugen kann.

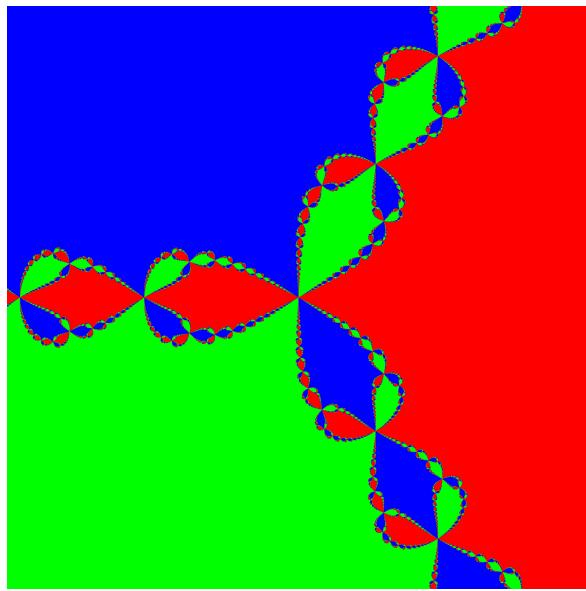


Abbildung 2: Newton Fraktal für  $f(x) = x^3 - 1$

### 3.1 Bildgenerierung

Als Programmiersprache wurde Java gewählt.

**Bibliotheken** Zusätzlich zu allgemeine Java-Bibliothek wurde noch das *javax.imageio.ImageIO* verwendet. Das ist öffentliche Bibliothek für die Arbeit mit Bildern, das unter anderem für das *.png* Format geeignet ist. Das PNG (Portable Network Graphics) Format passt für unsere Ziele ideal, wegen kleines Gewicht und keinen Qualitätsverlust.

**Bild** Mit Hilfe der *javax.imageio.ImageIO* kann man sehr einfach die *.png* Dateien erzeugen. Ein Beispiel in der Auflistung 1.

Listing 1: Ein Beispiel für PNG Generierung

```
BufferedImage bi = new BufferedImage(w, h, TYPE_INT_ARGB) ;
ImageIO.write(bi, "png", outputfilename);
```

Um den Bildinhalt zu generieren, benutzen wir den *bi.setRGB(X, Z, Color)*; Befehl. Natürlich soll man bestimmten Bereich wählen, um dann zu zeichnen. Bei unseres Programm wählt man den Startpunkt durch *X* und *Y* Koordinatenwerte, die Bildschirmgröße: *Size* und die Auflösung: *resolution*.

**Komplexe Zahlen** Für die Bearbeitung der komplexen Zahlen wurden die entsprechende Klasse implementiert, die die Arbeit mit komplexen Zahlen erleichtert. Man kann komplexe Zahlen addieren, abrechnen, multiplizieren, dividieren und potenzieren.

**Newton Verfahren** Um Newton Verfahren zu benutzen, soll man manuell die Ableitung berechnen, die Newton iterative Funktion ausfinden und im Programmcode umwandeln. Als Beispiel nehmen wir die Funktion  $f(z) = z^3 - 2z + 2$ . Entsprechende Ableitung hat die Form  $f'(z) = 3z^2 - 2$  und die iterative Funktion  $z_{n+1} = \frac{2z_n^3 - 2}{3z_n^2 - 2}$ . Und hier steht der Programmcode für diese iterative Funktion.

```
devide(mult(hoch(wert, 3), 2).add(-2), mult(hoch(wert, 2), 3).add(-2));
```

Natürlich nach jedem Schritt soll man Prüfen, ob die maximale Anzahl der Schritte *deep* nicht überschreitet wird und ob der Punkt schon genug nah zu Nullstelle liegt. Das heißt, dass es für jeden gefundenen Punkt  $z_n$  soll geprüft werden, ob  $n > \text{deep}$  ist und  $|f(z_n)| < \text{toleranz}$ . Falls einer der beiden Fälle eintritt, bricht man die iterative Berechnung ab. Im ersten Fall bezeichnet man das Punkt mit weiße Farbe, im zweiten - mit Farbe assoziierte mit gefundener Lösung.

**Lösungen** Da man manuell Farben für unterschiedliche Lösungen wählen will, war es entschieden, alle Lösungen mit assoziierten Farben im Programmcode eintragen. Ein Beispiel für die Funktion  $z^3 - 1$  findet man in der Auflistung 2

Listing 2: Lösungen mit Farben für  $z^3 - 1$

```
points.add(new point(1, 0, 255, 0, 0));
points.add(new point(-0.5, 0.866, 0, 255, 0));
points.add(new point(-0.5, -0.866, 0, 0, 255));
```

**Schatten** Für bessere Informationswiedergabe kann man die Schatten einschalten. Dann je mehr Schritten des Newtonverfahren nötig sind, desto dunkler das Pixel wird.

**Programmcode** Vollständige Programmcode kann man unter [Nal] finden.

### 3.2 Animation

Einer der einfachsten Wegen für die Animationserzeugung wurde gewählt. Das Programm wurde so geändert, dass *size* und *outputfilename* variabel werden. Dann im Zyklus wird eine Reihenfolge der Bildern generiert, so dass neues Bild die Annäherung des Alten wird. Dann aus dieser Reihenfolge mit Hilfe des [ani] wurde die Animation generiert.

## 4 Analyse

Es werden manche Newton Fraktale vorgestellt und analysiert.

## 4.1 $z^3 - 1 = 0$

Wie oben schon gesagt war, hat diese Gleichung drei Lösungen auf Komplexe Ebene:  $+1$ ,  $-\sqrt[3]{-1}$  und  $(-1)^{2/3}$  und war auf dem Bild 2 visualisiert. Wenn man versucht die Grenze zwischen zwei Zonen zu annähern, sieht man, dass dort immer dritte Zone vorkommt. Das wiederholt sich rekursiv bei Annäherung. In anderen Worten, wenn man einen Anfangspunkts  $z_0$  wählt, der gegen  $+1$  konvergiert und einen Punkt  $z_1$ , der gegen  $-\sqrt[3]{-1}$  konvergiert, dann existiert es immer einen dritten Punkt  $z_2$ , der noch näher zu  $z_0$  als  $z_1$  liegt und zu  $(-1)^{2/3}$  konvergiert.

Wichtigste Frage ist: warum sehen die Zonen nicht wie die einfachen 120-Grad-Sektoren aus? Anfang letztes Jahrhunderts gelang es zwei französischen Mathematiker Gaston Julia und Pierre Fatou zu zeigen, dass die Grenzpunkten eines Einzugsgebiets die Grenzpunkte aller Einzugsgebiete sind. Folglich können Iterationen mit mehr als zwei Einzugsgebieten keine einfach zusammenhängenden Liniensegmente als Gebietsgrenzen in 2D haben. Solche Grenzen müssen zwangsläufig fraktalen Natur sein, bestehend aus vollständig separaten Punktmengen - sozusagen eine unendlich feine Staubwolke aus nichtabzählbar vielen Staubpartikelchen. [Sch94] Machen wir uns damit ein bisschen vertraut, indem wir die Nullstelle (der Punkt  $0 : 0$ ) anschauen. Logischerweise soll die Nullstelle zwischen allen Zonen liegen und Null selbst konvergiert überhaupt nicht, da bei Newton Verfahren für diese Funktion,  $z = 0$  zu Division durch Null folgt. Aber diese Nullstelle ist nicht allein, außerdem existieren noch unendlich viele Punkte, die gegen die Null konvergieren und für die auch gilt, dass daneben sich alle Zonen befinden. (Das folgt aus die lineare Natur des Newton Verfahrens.) Diese Punkten, die gegen die Null konvergieren, bilden diese komische Grenze Zwischen Zonen. Zum Beispiel betrachten wir ein paar Punkte. Konvergenz des Punktes  $(1, 1)$  wird auf dem Bild 3 vorgestellt. Der Punkt liegt weit genug von der Grenze und wird eindeutig konvergiert, trotzdem sieht man so ein Sprung, die sehr oft bei Newton Verfahren eintreten werden.

Jetzt betrachten das Punkt  $(0.9, 1)$  auf dem Bild 4. Die Konvergenz des Punktes zeigt eindeutig, wie die iterative Schritte des Newton Verfahrens funktionieren und wie man zu solchen Bildern kommt. Erste drei Punkte sind gleich zu die Konvergenz des Punktes  $(1, 1)$ , was die lineare Natur des Newton Verfahrens bekräftigt, aber dann, neben der Nullstelle, wird die Konvergenz nach grünen Gebiet geworfen. Die Punkten neben Grenzen bewegen sich zuerst in die Richtung der Nullstelle und dann rasant in die passende Zone springen.

Zusätzliches Beispiel wird auf Bild 5 vorgestellt, wo helle Bereiche am schnellsten zu Lösungen konvergierende Punkte bezeichnen.

## 4.2 $z^5 - 1 = 0$

Das Bild 6 illustriert das Newton Fraktal. Eisziegen Unterschied liegt in der Anzahl der Zonen u Lösungen, hier gibt es 5. Wie bei  $z^3 - 1 = 0$  sind die Grenzpunkten eines Einzugsgebiets die Grenzpunkte aller Einzugsgebiete.

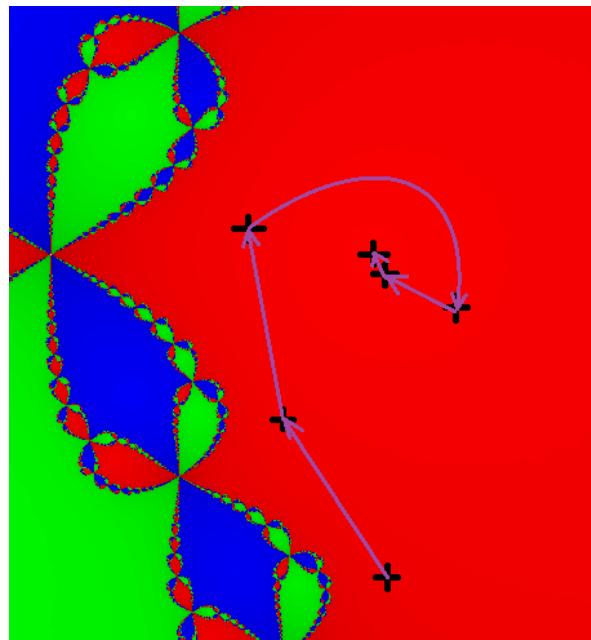


Abbildung 3: Konvergenz des Punktes  $(1, 1)$  für  $f(x) = x^3 - 1$

## 5 Zusammenfassung

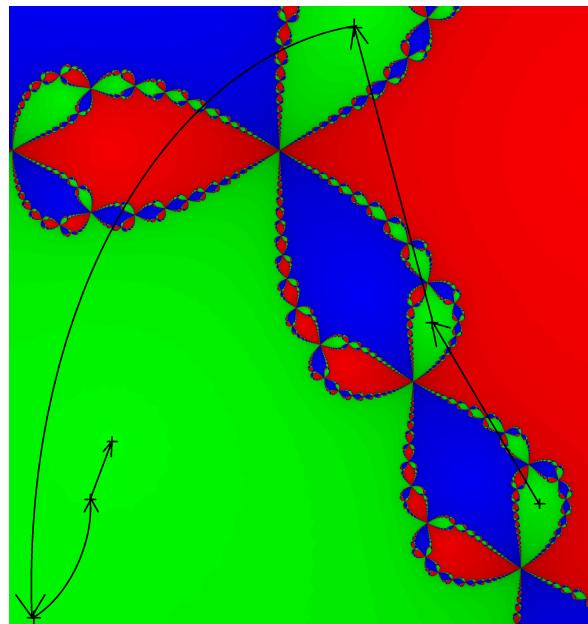


Abbildung 4: Die Konvergenz des Punktes  $(0.9, 1)$  für  $f(x) = x^3 - 1$

## Literatur

- ani. <https://toolson.net/GifAnimation/Create>
- Bey. [https://de.wikipedia.org/wiki/Datei:Mandelbrot\\_set\\_with\\_coloured\\_environment.png](https://de.wikipedia.org/wiki/Datei:Mandelbrot_set_with_coloured_environment.png)
- fra. *Fraktal.* <https://de.wikipedia.org/wiki/Fraktal.> – Online-Ressource
- Mak. <https://de.wikipedia.org/wiki/Datei:Romanesco.jpg>
- Nal. NALIVAYKO, Yaroslav: *Newton Fraktale Visualisierung.* [https://github.com/Jery77/Simulation\\_Prog/tree/master/src, .](https://github.com/Jery77/Simulation_Prog/tree/master/src, .) – Accessed Juni 4, 2017
- Sch94. SCHROEDER, Manfred ; SCHLIERF, Markus (Hrsg.): *Fraktale, Chaos und Selbstähnlichkeit.* Spektrum, 1994
- Tre92. TREFETHEN, Lloyd: The definition of numerical analysis. In: *SIAM News* (1992)

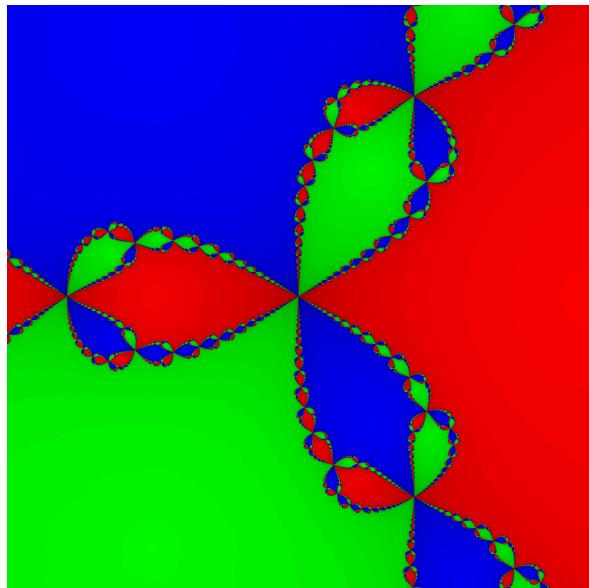


Abbildung 5: Die Konvergenzgeschwindigkeit durch die Helligkeit für  $f(x) = x^3 - 1$

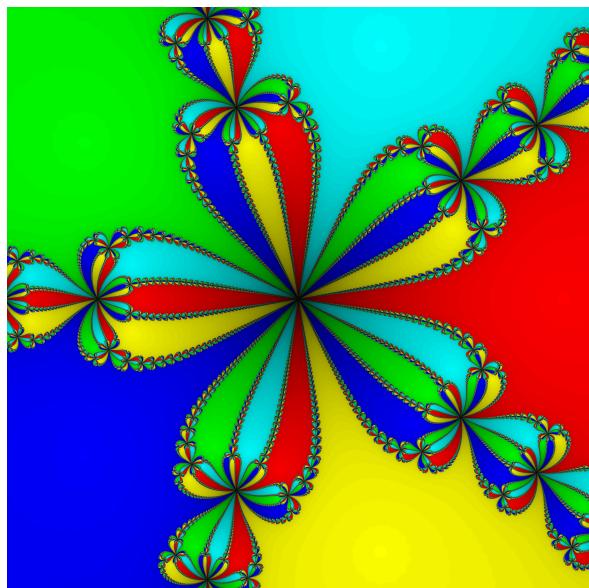


Abbildung 6: Das Newton Fraktal für  $f(x) = x^3 - 1$