Seminar Fraktale
MA Medienwissenschaft
Humboldt-Universität zu Berlin
Zeitbasierte Medien und zeitkritische Medienprozesse
Sommersemester 2017
Dr. Stefan Höltgen

Auf den Spuren des *Computer Literacy Project* und Helge von Kochs - ein Fraktal in BBC BASIC

Matthias Korte
Mariannenplatz 25
10997 Berlin
matthias.korte@hu-berlin.de
Matrikelnummer 576872

Inhaltsverzeichnis

1 Einleitung

In den 1970er-Jahren entstand in der Mathematik, und dann sogar in Teilen der Populärkultur ein Interesse an geometrischen Formen, die bis dahin wenig erforscht und noch unbenannt waren, den Fraktalen.[1] Das besondere war, dass die Erforschung und Faszination durch Computer ermöglicht wurde, die komplexe Abbildungen dieser geometrischen Gebilde berechnen und anzeigen konnten.[2] Etwa zeitgleich wurden die ersten Mikrocomputer, die kleiner und günstiger als die meisten bisherigen Computer waren, populär und ermöglichten die Nutzung des Computers durch Heimanwender.[3] Diese Arbeit behandelt den BBC Mikrocomputer und vor allem die dafür entwickelte Programmiersprache BBC BASIC. Allerdings wird nicht der materielle Computer selbst zur Betrachtung herangezogen sondern durch einen Emulator wird die Nutzung von dessen Programmiersprache simuliert. Die Arbeit soll durch die Benutzung der Sprache BBC BASIC, die Anfang der 1980er-Jahre für einen Computer entwickelt wurde, der hauptsächlich für den Bildungs- und Heimgebrauch gedacht war, einen Eindruck geben, wie einige Schüler und Heimanwender in Großbritannien erstmals das technische Medium Computer für eigene Programmierungen benutzten, Computergraphiken ausgeben ließen und zu einer Zeit, als die Nutzung des Werkzeugs Computer in der breiten Bevölkerung noch in den Kinderschuhen stand, vielleicht erstmals überhaupt mit dem Computer in Berührung kamen.

Nicht einmal ein Jahrzehnt vor der Veröffentlichung des BBC Mikrocomputers beschrieb und benannte der Mathematiker Benoît Mandelbrot geometrische Formen, die, wenn sie in Teile aufgeteilt wurden, eine kleine Kopie der ganzen Form wären und fand den Begriff Fraktal für dieses mathematische Phänomen, das bereits lange vor ihm entdeckt und beschrieben wurde. Mandelbrot aber nutzte das recht neue technische Medium Computer mit den für diese Forschung hilfreichen Möglichkeiten der Datenspeicherung und des Datenzugriffs, der Rechenleistung und der Möglichkeiten zur graphischen Darstellung zur tiefgreifenden Forschung an

---

[1] vgl. Josh Sanburn, *Dude, you have fractals on your wall*, in: Time, http://newsfeed.time.com/2010/10/18/dude-you-have-fractals-on-your-wall/ (Abrufdatum: 24.01.2018).

[2] vgl. Herbert Zeitler, *Fraktale und Chaos – Eine Einführung,* Darmstadt 1993, S. 3f.

[3] Joseph D Dumas, *Computer Architecture -Fundamentals and Principles of Computer Design*, Boca Raton, FL 2006, S.10f.

Fraktalen. Erst der Computer machte die Arbeit Mandelbots und somit das in den darauffolgenden Jahrzehnten populäre Phänomen der Fraktale in dieser Weise möglich.[4]

In dieser Arbeit wird die Koch-Kurve, ein vor über 100 Jahren erstmals beschriebenes Fraktal genauer beschrieben, analysiert und mithilfe des Computers dargestellt. Eine Codierung in BBC BASIC, erstellt in einem Emulator, soll zeigen, wie die Sprache des BBC Mikrocomputers genutzt werden kann, um Funktionen zu lösen, Rekursionen durchzuführen und dadurch dieses Fraktal darzustellen. Zuvor werden der Mikrocomputer und dessen vorinstallierte Programmiersprache vorgestellt und das Fraktal und ein besonderer Dimensionsbegriff zur Analyse dieser Gebilde erklärt. Im letzten Schritt wird der Autor kurz seine eigene Erfahrung mit dem Erlernen und der Anwendung der Sprache BBC BASIC und die Annäherung an die Codierung der Koch-Kurve wiedergeben.

2 Der BBC Micro und BASIC

Im Jahr 1978 sendete die British Broadcasting Corporation (BBC) die Sendung *Now the Chips are Down* als Teil ihrer Reihe *Horizon*. Die Folge beschäftigte sich ausführlich mit Mikrocomputern und spielte eine wichtige Rolle darin, die Öffentlichkeit und die Regierung Großbritanniens für dieses Thema zu sensibilisieren. Durch Innovationen im Bereich der Mikroprozessoren sanken die Kosten für entsprechende Computer in den 1970er-Jahren stetig. Nachdem andere Staaten wie die USA oder Japan Vorreiterrollen in diesem Bereich eingenommen hatten, diese Industrie förderten und Heimcomputer wie der APPLE II erschienen waren, entschied die 1979 neu gewählte Regierung Großbritanniens diesen Sektor verstärkt zu unterstützen und führte erstmals das ‚Ministry of State for Industry and Information Technology' ein.[5]

[4] vgl. Herbert Zeitler, *Fraktale und Chaos – Eine Einführung,* Darmstadt 1993, S. 3f.
[5] vgl. Carl Gardner u. Robert Young, *Scrience on TV: A Critique*, in: *Popular television and film: a reader,* London 1981: S. 173; u. Alison Gazzard, *Now the Chips Are Down: The BBC Micro,* Cambridge 2016, S. 5f.; ein direkter Zusammenhang zwischen einer Erhöhung des Etas im Bereich der Informationstechnologie durch die Regierung Thatcher und dem Aufkommen des *Computer Literacy Project* wird zwar nicht explizit gesehen, wohl aber das veränderte politische Klima, das zu höherer

Nach der erfolgreichen Dokumentation *The Mighty Micro* (1979) wuchs bei der öffentlich-rechtlichen Fernsehanstalt BBC das Interesse an der Mikrochiptechnologie und die Möglichkeiten der Verwendung für die allgemeine Bevölkerung. Durch ein großes Programm sollte die Sensibilität zur Informationstechnologie erhöht werden und im Januar 1982 wurde das *Computer Literacy Project* initiiert.[6] In einer Mitteilung erklärt die BBC das Ziel zu diesem entsprechenden Projekt folgendermaßen:

> „The aim of the project is to introduce interested adults to the world of computers and computing, and to provide the opportunity for viewers to learn through direct experience how to program and use a microcomputer.“[7]

Zu dem lang angelegten Projekt sollte die zehnteilige Fernsehsendung *The Computer Programme* sowohl regulär im Fernsehen als auch speziell für Schulen und andere Bildungseinrichtungen ausgestrahlt werden.[8]

Abb. 1: Acorn BBC Micro Model A[9]

Akzeptanz des und Interesse am Heimcomputer in der Bevölkerung gesorgt haben soll.

[6] vgl. Alison Gazzard, *Now the Chips Are Down: The BBC Micro,* Cambridge 2016, S. 5ff.

[7] aus der Mitteilung der British Broadcasting Corporation: *BBC Continuing education Television: Computer Literacy Project*, 1981, S. 1: http://www.computinghistory.org.uk/det/7182/BBC-Computer-Literacy-Project/ (Abrufdatum: 23.01.2018).

[8] vgl. British Broadcasting Corporation, *BBC Continuing education Television: Computer Literacy,* 1981, S. 1-4.

[9] Centre For Computing History: http://www.computinghistory.org.uk/det/3041/acorn-bbc-micro-model-a/ (Abrufdatum: 23.01.2018).

Zu dieser Fernsehsendung wurde ein eigener Computer angeboten, der BBC Microcomputer, auch BBC Micro genannt. Durch den frei verkäuflichen und teilweise auch in Schulen eingesetzten Computer konnte das in der Sendung Vorgestellte direkt umgesetzt werden. Ein solches Gerät sollte aber nicht Voraussetzung für das Verstehen des Programms sein, sondern auch generell das Thema Computer aus einem theoretischen und einem praktischen Blickwinkel beleuchten werden. Die auf dem BBC Micro verwendete und in der Sendung angewandte Programmiersprache war eine Abwandlung von BASIC und bot sich an, da sie für Anfänger als leicht zugänglich galt. Zu dieser Zeit gab es eine Menge aktueller Literatur zu dieser Sprache und genügend allgemeine Beispielprogramme konnten auf verschiedene Systeme portiert werden. Die Hardware und Software der Maschine waren speziell auf das *Computer Literacy Project* angepasst. Für die gemeinsame Entwicklung und Herstellung des Computers wählte die BBC den britischen Hersteller Acorn Computers, der Ende der 1970er-Jahre den Mikrocomputer Acorn Atom veröffentlichte, der bereits für den Verbrauchermarkt entwickelt worden war. Zur Einführung gab es zwei Modelle dieses Computers. Die einfache Version wurde ab Dezember 1981 für 235£ angeboten und basiert auf einem 2 MHz Mikroprozessor mit einem 16 KB großen Arbeitsspeicher, einem Betriebssystem und BBC BASIC auf einem 32 KB großen RAM-Speicher. Der Computer bot eine Grafikauflösung von 320 × 256 Pixel und konnte acht Farben ausgeben. Das leistungsstärkere Modell für 335£ bot eine doppelte Auflösung und doppelten RAM-Speicher. Die gesamte Technik war inklusive Tastatur in einem kompakten weißen Gehäuse untergebracht, was den BBC Micro insbesondere für den Heimgebrauch nützlich machte. Diese Bauart war in den 1980er-Jahren bei Heimcomputern sehr verbreitet. Neben den Schnittstellen zur Videoausgabe waren die möglichen Erweiterungen durch zusätzliche Module wie eine Erhöhung des RAMs oder einen zweiten Prozessor eine Besonderheit. [10]

Schon der Name der Programmiersprache BASIC gibt einen deutlichen Verweis auf Zweck und Art dieser Sprache. Als Beginner's All-purpose Symbolic Instruction

[10] vgl. British Broadcasting Corporation, *BBC Continuing education Television: Computer Literacy,* 1981, S. 1-4; u. Alison Gazzard, *Now the Chips Are Down: The BBC Micro,* Cambridge 2016, S. 9-11 u. S. 24-28.

Code, also eine auf Symbolen basierende Allzweck-Programmiersprache für Anfänger, war sie von Beginn an als Sprache gedacht, die den Zugang zur Programmierung erleichtern soll. John G. Kemeny und Thomas E. Kurtz entwickelten BASIC am Dartmouth College in Hanover, New Hampshire, um Studenten einen im Vergleich zu anderen Sprachen besonders einfachen Umgang mit Computern und Grundlagen der Programmierung zu ermöglichen. Nach dem erstmaligen Einsatz 1964 wurde die Sprache nicht nur von Studierenden genutzt und weiterentwickelt, sondern fand auch Anwendung in Schulen, denen sie kostenlos zur Verfügung gestellt wurde. Über den Bildungsbereich hinaus war BASIC nützlich, so ermöglichte die Sprache etwa auch Ingenieuren oder Statistikern einfache mathematische Probleme zu lösen.[11] Bei der BBC wurde der BASIC-Experte und Spezialist im Bereich Mikrocomputer Richard Russel vom *BBC Engineering Designs Department* mit dem Projekt betreut. Die Sendeanstalt wollte für das Computerprojekt eine Sprache verwenden, die übersichtlich strukturiert war, um für eine praxistaugliche Anwendung im Heimbereich nützlich zu sein, gleichzeitig aber mächtig genug, um auch weiterentwickelt zu werden und zukunftssicher zu sein. Russel definierte die genauen Spezifikationen und passte BASIC den Anforderungen der BBC an und gilt als Erfinder dieses Dialekts.[12]

Ursprünglich war das *Computer Literacy Project* weniger auf Schulen als auf Erwachsene ausgerichtet, um eine erste Annäherung an den Computer oder eine Weiterbildung in der Programmierung zu bieten. Die BBC kooperierte für das Projekt eng mit der gemeinnützigen Organisation Broadcasting support service (Bss), bei der sich am *Computer Literacy Project* interessierte Organisationen registrieren konnten. Über 1000 Erwachsenenzentren, Universitäten und Computerclubs zeigten Interesse an diesem Projekt. Dies führte zu einem weit reichenden Angebot und einer großen Vielfalt an Bildungsmöglichkeiten auch unabhängiger Kursanbieter im Zusammenhang mit dem *Computer Literacy Project.* Darüber hinaus wurde der BBC Micro auch in einer Vielzahl an weiterführenden Schulen eingesetzt. Das *30 Hour*

[11] vgl. Peter Sanderson, *Interactive Computing in BASIC - An introduction to interactive computing and a practical course in the BASIC language*, Bath 1973, S. 43f.; u. Thomas Kurtz, *BASIC Session,* in: *History of programming languages*, New York 1981, S. 518-520.
[12] Tilly Blyth, *The Legacy oft he BBC Micro – effecting change in the UK's Culture of Computing*, London 2012, S. 12 u. 70.

BASIC Kursbuch etwa verkaufte sich 150.000 bis 250.000 mal und die Anzahl der angebotenen Computerkurse in Großbritannien vervielfachte sich zwischen 1981 und 1984. Trotz allgemeiner, staatlicher Förderung hing der Einsatz von Mikrocomputern in britischen Schulen maßgeblich von den lokalen Bildungsbehörden und der Bereitschaft der Schulen, Computerkurse und Weiterbildungen des Lehrpersonals zu unterstützen, ab.[13]

3 Fraktale

Als Fraktal oder fraktale Strukturen werden natürliche oder künstlich erstellte Figuren oder Muster bezeichnet, die ein hohes Maß an Selbstähnlichkeit aufzeigen und sich häufig durch die vielfache Wiederholung des eben gleichen Motivs in unterschiedlichem Maßstab auszeichnen. Ein fraktaler Aufbau ist etwa in der Natur zu finden, als Beispiel dient der Romanesco, eine Variante des Blumenkohls, dessen Erscheinung von Selbstähnlichkeit geprägt ist. Jede Knospe ähnelt dem gesamten Kopf und besteht aus weiteren kleineren Knospen. Dabei bilden die verzweigten Meristeme eine Spirale und nähern sich so in ihrem Aufbau dem Fraktal. Dieses Muster setzt sich weiter fort, endet aber bei ausreichend kleiner Struktur, wodurch es sich von der klassischen geometrischen Form eines Fraktals unterscheidet.[14]

Erst 1975 wurde der Begriff Fraktal geprägt, in *Les objets fractals, forme, hasard et dimension* beschreibt der Französisch-US-Amerikanische Mathematiker Benoît Mandelbrot das Phänomen der Fraktale und benennt sie nach dem lateinischen Adjektiv *fractus* und dem abgeleiteten Verb *frangere*, übersetzt gebrochen bzw. zerbrechen. Mandelbrot war nicht der erste Wissenschaftler, der sich mit diesem Phänomen auseinandersetzte. Er selbst nennt etwa den Mathematiker Georg Cantor, der bereits zu Ende des 19. Jahrhunderts eine bestimmte Teilmenge der reellen Zahlen beschrieb, die heute als Cantor-Menge oder auch Cantor-Staub bekannt und aufgrund ihrer Selbstähnlichkeit und einer ungeraden Hausdorff-Dimension ein Fraktal ist

[13] Tilly Blyth, *The Legacy oft he BBC Micro – effecting change in the UK's Culture of Computing*, London 2012, S. 26-30.
[14] vgl. David Feldman, *Chaos and Fractals – An Elementary Introduction*, Oxford 2012, S, 4f.; u. Hartmut Jürgens, Heinz-Otto Peitgen, Dietmar Saupe, *Chaos and Fractals – new frontiers of science*, New York 2004, S. 138.

(siehe Abb. 2). Weitere Mathematiker erforschten fraktale Geometrie oder ihre besondere Dimension, etwa der Mathematiker Wacław Sierpiński, der mit dem Sierpiński-Dreieck bereits 1915 das vielleicht klassischste Beispiel eines Fraktals beschrieb (siehe Abb. 2) oder Felix Hausdorff, dessen Dimensionsbegriff zur Analyse und Erforschung von Fraktalen aufgegriffen und genutzt wird.[15]

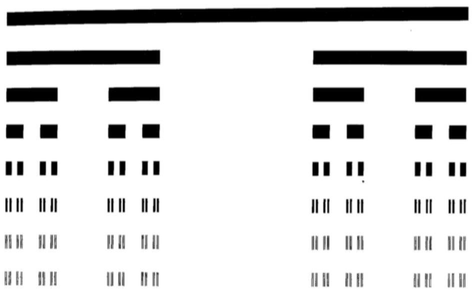
(a)  (b) 

Abb. 2: Cantor-Menge (a) und Sierpiński-Dreieck (b)[16]

Das in Abb. 2(b) gezeigte Dreieck wird im nächsten Schritt um den Skalierungsfaktor 0,5 verkleinert und drei Kopien werden ähnlich der Ursprungsform zusammengesetzt. Das vierte Dreieck zeigt dabei die dritte Iteration der ursprünglichen Figur und wird daher dritte Iterationsstufe genannt. Diese Iteration kann unendlich fortgesetzt werden. Wird ein Teilbereich der Figur daraufhin vergrößert zeigt sich stets das gleiche Bild. Das Sierpiński-Dreieck ist genau selbstähnlich, da es immer aus verkleinerten Kopien seiner selbst besteht.

Durch das Aufkommen des Computers boten sich in der zweiten Hälfte des 20. Jahrhunderts ganz neue Möglichkeiten, Fraktale zu berechnen und darzustellen. Mandelbrot nutzte dieses Vorwissen und die Möglichkeiten neuer Technologie und erforschte neben anderen Wissenschaftlern die fraktale Geometrie. Hervorzuheben ist hierbei die nach ihm benannte Mandelbrot-Menge. Die Wurzeln gehen auf die Theorien der französischen Mathematiker Gaston Maurice Julia und Pierre Fatou zurück. Die nach Ersterem benannte Julia-Menge zeigt eine Menge komplexer

[15] vgl. Benoît B. Mandelbrot, *The Fractal Geometry of Nature,* New York 1982, S. 4f. u. S.15; u. Wolfgang Neidhardt, Herbert Zeitler, *Fraktale und Chaos – Eine Einführung,* Darmstadt 1993, S. 3f.
[16] Benoît B. Mandelbrot, *The Fractal Geometry of Nature,* New York 1982, S. 80 u. 142.

Zahlen, die nicht an eine Grenze konvergieren, wenn eine gegebene Zuordnung wiederholt auf sie angewendet wird. In einigen Fällen ist das Ergebnis hierbei ein verbundener Fraktalsatz. Die Arbeiten der beidem Mathematiker gerieten weitgehend in Vergessenheit bis Benoît Mandelbrot in den 1970er-Jahren auf Grundlage der Julia-Menge die Mandelbrot-Menge definierte. Durch den Computer als Darstellungsmittel konnte er sie bis ins kleinste Detail darstellen und so faszinierende Bilder erstellen. Aufgrund ihrer sonderbaren, an einen Apfel erinnernden Form wird die graphische Darstellung auch Apfelmännchen genannt (siehe Abb. 3).[17]

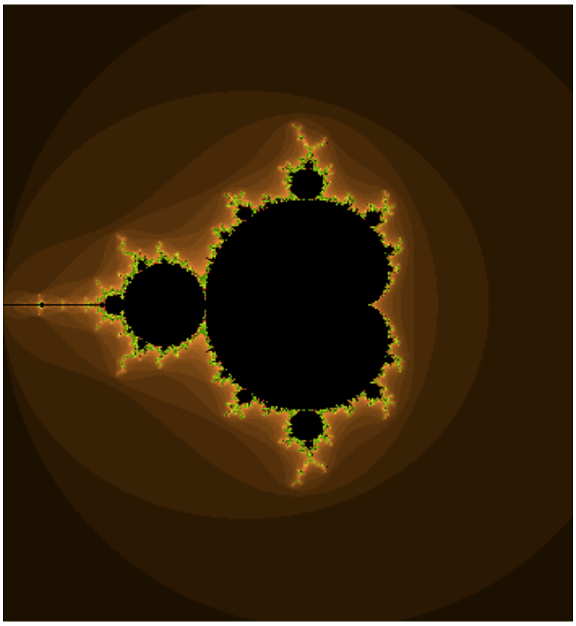


Abb. 3: Die Mandelbrot-Menge[18]

4 Die fraktale Dimension

Eine zentrale Eigenschaft der Fraktale ist ihre besondere Dimension. Unterschiedliche Definitionen dieser Dimension gehen auf die Arbeit des Mathematikers und Schriftstellers Felix Hausdorff aus dem Jahr 1919 zurück, der sich sowohl mit

[17] vgl. Hartmut Jürgens, Heinz-Otto Peitgen, Dietmar Saupe, *Fraktale - eine neue Sprache für komplexe Strukturen,* in: *Chaos und Fraktale*, Heidelberg 1989, S. 113 f.

[18] Mandelbrot-Menge erstellt in BBC BASIC mit einer Codierung der Website Rosettacode: http://rosettacode.org/wiki/Mandelbrot_set#BBC_BASIC (Abrufdatum: 23.01.2018).
Die einzelnen farbigen Abstufungen entstehen durch den Einsatz eines Zählers. Die Anzahl der Iterationen, die das Programm für jeden Punkt durchführen muss, um herauszufinden, dass er nicht innerhalb der Menge liegt, definiert die Farbe.

metrischen Räumen beschäftigte, als auch Aphorismen und Gedichte schrieb.[19] Mandelbrot nutzte eine vereinfachte Form der Hausdorff-Dimension, die er gemeinsam mit dem Begriff Fraktale als fraktale Dimension einführte.[20]

Allgemein bekannt sind die ganzzahligen Dimensionen einfacher Objekte im Vektorraum, also die eindimensionale Strecke, das Quadrat mit der Dimension 2 und der Würfel als dreidimensionales Objekt. Werden diese drei Objekte jeweils zu gleichen Teilen unterteilt, kann eine Beziehung zwischen der Anzahl der Teile (a), der Skalierung (s) und der Dimension (D) gesehen werden: Wird die Strecke mit der Dimension 1 in drei gleichgroße Teile zerlegt, so ist die gesamte Strecke dreimal so lang wie jedes Teil. Wenn ein Quadrat in gleichgroße Teile mit einer Seitenlänge von 1/3 zerlegt wird, entstehen 9 gleich große Quadrate (Anzahl $a=3^2$). Wird dies ebenfalls bei einem dreidimensionalen Würfel angewandt, sind es am Ende sogar 27 (Anzahl $a=3^3$) gleich große Teile. Für jede dieser Skalierungen wurde der Skalierungsfaktor s = 3 angewandt, woraus sich das Gesetz $a=s^D$ ergibt. Wird diese Gleichung nach der Dimension D aufgelöst ergibt sich $D = \frac{\log a}{\log s}$. Die Dimension kann also durch den Quotient der Logarithmen der Skalierung (s) und der Anzahl der Objekte (a) gebildet werden. Neben der allgemein bekannten, ein-, zwei- und dreidimensionalen euklidischen Geometrie lässt sich dies auch auf selbstähnliche Objekte anwenden. Die in Abb. 2 gezeigte Cantor-Menge etwa besteht in jeder Iterationsstufe aus zwei selbstähnlichen, um den Faktor 3 skalierten Teilen, wodurch sich eine nicht-ganzzahlige Hausdorff-Dimension errechnen lässt:

$$\frac{log2}{log3} \approx 0{,}631$$

In der gleichen Abbildung 2 ist das Sierpiński-Dreieck, das sich aus drei Teilstücken zusammensetzt und insgesamt doppelt so groß ist wie jedes einzelne Dreieck, die Skalierung (s) beträgt also 2 und die Anzahl (a) 3, was hier eine Hausdorff-Dimension zwischen 1 und 2 ergibt:

$$\frac{log3}{log2} \approx 1{,}585$$

[19] vgl. Hartmut Jürgens, Heinz-Otto Peitgen, Dietmar Saupe, *Fraktale - eine neue Sprache für komplexe Strukturen,* in: *Chaos und Fraktale*, Heidelberg 1989, S. 117.
[20] Benoît B. Mandelbrot, *The Fractal Geometry of Nature,* New York 1982, S. 14f.

Eine solche Berechnung der Dimension ist nur möglich, da die jeweiligen Objekte genau selbstähnlich sind, ein einzelnes Dreieck des Sierpiński-Dreiecks kann bei Anwendung der richtigen Skalierung das gesamte Objekt bilden. Diese Definition der Dimension auf ein Fraktal angewandt wird daher auch Selbstähnlichkeits-Dimension genannt.[21] Um eine nicht-ganzzahlige Dimension im Vergleich zur ganzzahligen Dimension in der euklidischen Geometrie verständlich zu machen, kann die Cantor-Menge als etwas verstanden werden, das weder eine Linie ist und damit keine Dimension von 1 hat, noch ein Punkt ist und damit nicht dimensionslos ist, stattdessen ähnelt sie in mancherlei Hinsicht einer Linie, in anderer Hinsicht ähnelt sie einem Punkt. Die Bildung dieses Fraktals beginnt mit einer zweidimensionalen Linie, von der nach vielen Iterationen jedoch so viel entfernt wurde, dass die weiteren Iterationsschritte wie eine Ansammlung von Punkten wirken, auch Cantor-Staub genannt.[22] Fraktale können allerdings auch eine ganzzahlige Dimension aufweisen, jedoch ist die nicht-ganzzahlige Dimension eine häufig und gerade bei populären Fraktalen auftretende Besonderheit.[23]

5 Funktionsweise BBC BASIC

Das in Kapitel 2 bereits vorgestellte BASIC liegt für diese Arbeit in einer mit dem Betriebssystem Mac OS X nutzbaren Version der vom BBC Microcomputer verwendeten Sprache BBC BASIC vor. Als die Weiterentwicklung des BBC Micro und des Basic-Dialekts seitens Acorn und der BBC eingestellt wurde beschäftigte sich der englische Entwickler Richard Russell, selbst als Design Engineer bei der BBC maßgeblich an der Entwicklung von BBC BASIC beteiligt, privat weiter mit diesem Thema und veröffentlichte 2001 erstmals eine Version für Microsoft Windows, welche 2016 zu einer Version für Apple Mac OS X implementiert wurde. Hierbei wurde versucht, möglichst genau das BBC BASIC des Mikrocomputers zu emitieren,

---

[21] vgl. Paul S. Addison, *Fractals and Chaos – An illustrated course*, London 1997, S. 14ff.; u. Hartmut Jürgens, Heinz-Otto Peitgen, Dietmar Saupe, *Fraktale - eine neue Sprache für komplexe Strukturen,* in: *Chaos und Fraktale*, Heidelberg 1989, S. 117f.

[22] vgl. David P. Feldman, *Chaos and Fractals – An Elementary Introduction*, Oxford 2012, S. 167f.

[23] Beispiele für Fraktale mit einer ganzzahligen Hausdorff-Dimension sind etwa die Drachenkurve und der Pythagoras-Baum.

selbst die Töne des Computers wurden hierzu emuliert.[24] Die zu diesem Zweck gegründete R.T. Russell Company betreibt die Website bbcbasic.co.uk und bietet dort neben der erwähnten Mac OS X-Version *BBC BASIC for SDL 2.0* weitere BBC BASIC Programme zum Download an und zeichnet sich durch eine große Menge ausgesprochen detaillierter Anleitungen zur Anwendung dieser Sprache aus, auf die auch maßgeblich für die Programmierung in dieser Arbeit zurückgegriffen wurde.

Leicht nachvollziehbare Befehle sorgen bei BASIC für eine einfache Anwendung und schnelle Erlernbarkeit. In der imperativen Programmiersprache ist ein Programm als eine Folge einzelner Anweisungen aufgebaut, die vorgeben, was von der Software ausgeführt wird und in welcher Reihenfolge dies geschehen soll. Ein einzelnes Programm besteht aus einer Menge an Befehlen und Anleitungen, die in einzelne Reihen untereinander geschrieben werden, wobei eine Besonderheit von BBC BASIC ist, dass es auch funktionsfähige, einzeilige Programme gibt, die nur aus wenigen Zeichen bestehen. Es bietet sich an, die einzelnen Reihen in Zehnerschritten zu nummerieren, sodass zu einem späteren Zeitpunkt weitere Reihen zwischen bestehende eingefügt werden können, ohne die gesamte Nummerierung erneuern zu müssen. Es ist auch möglich mehrere Befehle in eine Reihe zu setzen, aufgrund der leidenden Übersichtlichkeit empfiehlt sich dieses Vorgehen allerdings nicht. Wenn es wegen einer fehlerhaften Programmierung zu Komplikationen bei der Durchführung eines Programms kommt teilt BBC BASIC dies dem Nutzer mit dem Hinweis auf die Art des Fehlers mit.[25]

Die einzelnen Befehle in BASIC orientieren sich an der englischen Sprache, wodurch sie in der Regel für den Nutzer direkt nachvollziehbar sind. Ein einfaches PRINT vor einer zu berechnenden Gleichung sorgt bei Durchführung des Programms direkt zur Anzeige des Ergebnisses in einem neuen Fenster. Mit den vergleichsweise

[24] Die Arbeit Russels bei der BBC und darüber hinaus ist auf der Seite *Recollections of BBC engineering* nachzulesen: http://www.bbceng.info/Designs/designs_reminiscences/Richard_Russell/rtrdd.html Genaueres zur Historie der Sprache BBC BASIC auf der Seite der R.T. Russel Company: http://www.rtrussell.co.uk/products/bbcbasic/history.html; Die Software ist auf dieser Unterseite zu finden: http://www.bbcbasic.co.uk/bbcwin/bbcwin.html (Abrufdatum jeweils: 23.01.2018).

[25] vgl. Andrew C. Bray, Adrian C. Dickens, Mark A. Holmes BA, *The Advanced User Guide for the BBC Microcomputer*, Cambridge 1983, S. 7ff.

verständlichen Befehlen lassen sich auch komplexe Programme einfach berechnen und durchführen. Neben Ziffern kann auch Text ausgegeben werden und in der Durchführung und Ausgabe des Programms sogar aufgefordert werden, eine Eingabe in das Programm zu geben, die dann vom Programm weiterverarbeitet wird. In diesem Beispiel lassen sich einige besondere Funktionen erkennen:

```
10 REPEAT
20   INPUT "Ich nenne den Flaecheninhalt deines Kreises, wie ist der Radius?: " Radius
30   F = PI*Radius^2
40   PRINT "Der Flaecheninhalt betraegt ";F
50   INPUT "Noch einmal? (J oder N) " Antwort$
60 UNTIL Antwort$="N" OR Antwort$="n"
70 PRINT "Danke und bis dann!"
80 END
```

Das Programm berechnet den Flächeninhalt eines Kreises, wenn der Anwender den Radius eingibt. Bis der Nutzer eine weitere Berechnung ablehnt wird durch den Befehl REPEAT eine Schleife ausgelöst, die zu jeder neuen Eingabe den entsprechenden Flächeninhalt angibt. Dies hätte auch durch Bedingungen ausgelöst werden können wie ‚IF Antwort$="J" OR Antwort$="j" THEN: GOTO 20'. Eine Funktion wie IF ist etwa nützlich, wenn ein Teil der Programmierung nur unter bestimmten Bedingungen überhaupt ausgeführt werden soll. Die Kompaktheit des Programms in acht Zeilen ist ein Vorteil der Programmiersprache, da sie auch auf kleinen, speicherplatzarmen Rechnern wie ersten Microcomputern genutzt werden kann.[26]

Neben der Ausgabe von Zeichen ist BBC BASIC in der Lage Graphiken darzustellen, um etwa geometrische Figuren oder Berechnungen auszugeben, von Beginn an war der BBC Micro in der Lage, farbige Bilder auszugeben (siehe Abb. 4).

[26] vgl. Andrew C. Bray, Adrian C. Dickens, Mark A. Holmes BA, *The Advanced User Guide for the BBC Microcomputer*, Cambridge 1983, S. 7ff.

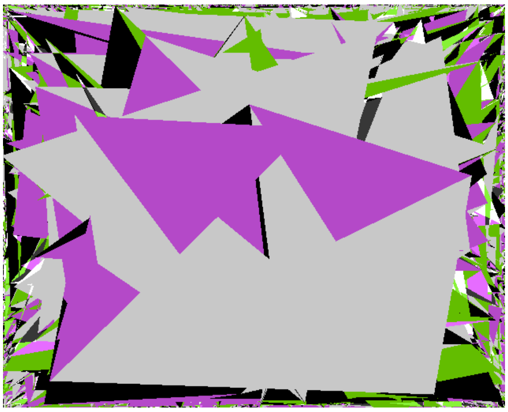


Abb. 4: kaleidoskopisches Muster mit BBC BASIC[27]

Während beim ersten BBC Microcomputer durch die Definition eines von sieben Graphikmodi mit dem Befehl MODE eine Graphikauflösung bis 640×256 Pixel mit 8 Farben möglich war können modernere BBC BASIC-Programme heute auch höhere Auflösungen anzeigen.[28] Einfache Befehle wie DRAW, MOVE oder CIRCLE zeigen in Kombination mit X und Y-Koordinaten bzw. dem Radius einfache geometrische Formen. Komplexere Darstellungen sind durch PLOT möglich, ein Befehl, der in über 200 unterschiedlichen Modi die Form und auch die Darstellung genauer definiert, oder direkt über den Befehl VDU, der vielfältige wichtige Aktionen zu Bildschirmdarstellung und Grafiken ausführt und ebenfalls die MODE-Displaymodi definieren kann.[29] Zwar ist die vergleichsweise einfache Handhabe eine Eigenschaft, die BBC BASIC für den Einstieg in die Programmierung etwa zur Anwendung im Bildungsbereich prädestiniert, allerdings eignet sich die Sprache aufgrund des Funktionsumfangs, der weit über das hier erwähnte hinausragt, auch zur Lösung komplizierter Problemstellungen und komplexer Programmierung.

---

[27] Der BBC Microcomputer wurde mit dem Buch *Welcome: to the BBC Computer Literacy Project* ausgeliefert, das auch die graphischen Möglichkeiten des Geräts zeigen sollte. Die Abbildung ist ein Screenshot der Ausführung eines fünfzeiligen Codes aus diesem Buch, der in einer ewigen Schleife durch den Code RND scheinbar zufällig und in hoher Taktrate neue, bunte Grafiken kaleidoskopartig darstellt. Vgl. Alison Gazzard, *Now the Chips Are Down: The BBC Micro*, Cambridge 2016, S. 31f.

[28] Alison Gazzard, *Now the Chips Are Down: The BBC Micro,* Cambridge 2016, S. 26.

[29] John D. Ferguson et al, *Programming the BBC Micro*, Kent 1983, S. 34ff.; Eine Auflistung der Unterschiedlichen VDU-Befehle auf S. 44.

6 Die Koch-Kurve

Bereits vor Benoît Mandelbrots Geburt veröffentlichte der schwedische Mathematiker Helge von Koch in zwei Artikeln 1904 und 1906 eine geometrische Figur, die heute als Koch-Kurve bekannt ist. Diese Kurve zeichnet sich nicht durch glatte Rundungen aus sondern ist ein höchst komplexes Gebilde, das durch die vielen Falten an eine Küstenlinie oder mit vier entsprechend rotierten Kopien ihrer selbst ein Schneeflockenähnliches Gebilde schafft, die Koch-Schneeflocke (siehe dazu Abb. 5).

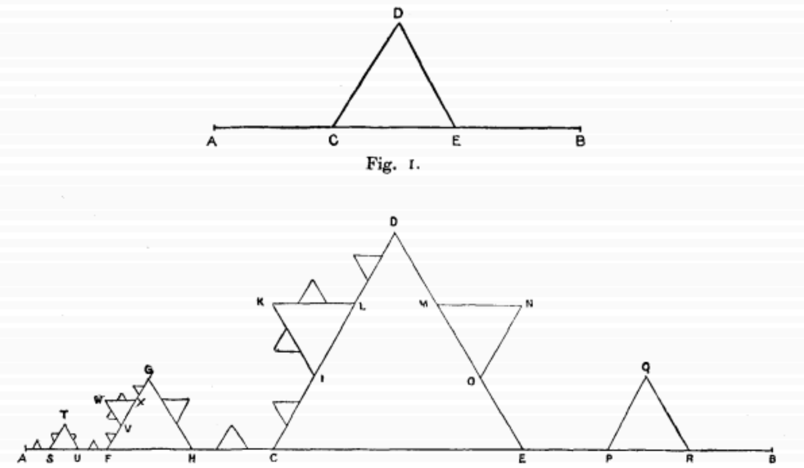


Fig. 1.

Fig. 2.

2. Partons maintenant d'une ligne droite déterminée AB, le sens de A vers B étant considéré comme positif (fig. 2). Par l'opération Ω, AB est remplacée par la ligne brisée $ACDEB$, les segments AC, CD, DE, EB étant égaux entre eux et leur sens positif étant respectivement celui de A vers C, de C vers D, de D vers E, de E vers B.

Effectuons l'opération Ω sur chacun de ces segments; la ligne $ACDEB$ sera remplacée par la ligne brisée $AFGHCIKLDMNOEPQRB$ composée de 16 segments égaux AF, FG etc.

Abb. 5: Ausschnitt aus Kochs Artikel von 1906[30]

Die Koch-Kurve entsteht durch eine stete Iteration, der sogenannte Initiator, also die erste geometrische Form, auf der das Fraktal aufbaut ist hierbei eine einfache Linie. Als erster Schritt wird diese Linie in drei gleichgroße Teile gegliedert. Das mittlere Drittel wird daraufhin durch ein gleichseitiges Dreieck ersetzt und die untere Seite dieses Dreiecks entfernt. Die entstandene Figur besteht nun aus vier geraden Strecken,

[30] Helge Von Koch, *Une méthode géométrique élémentaire pour l'étude de certaines questions de la théorie des courbes planes*, in: *Acta Mathematica 30*, Institut Mittag-Leffler, Djursholm 1906, S. 149; erstmals stellt von Koch die Kurve im Artikel *Sur une courbe continue sans tangente, obtenue par une construction géométrique élémentaire* vor. In: *Arkiv för matematik 1*, Norstedt & Soner, Stockholm 1904, S. 681-704.

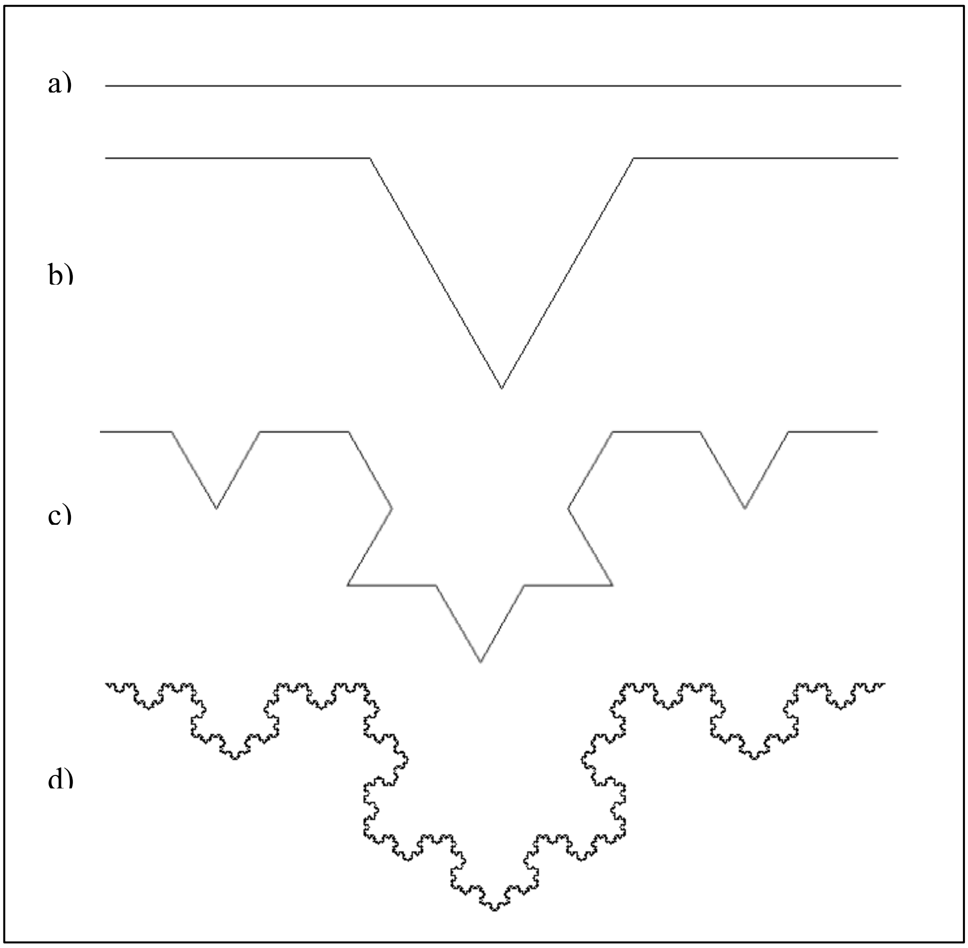


Abb. 6: Die Koch-Kurve: a) Initiator n=0, b) Generator n=1,
c) zweite Iterationsstufe n=2, d) sechste Iterationsstufe n=6[31]

die jeweils ein drittel so lang sind wie die ursprüngliche Linie. Dieser erste Schritt wird Initiator genannt. Nun wird das Fraktal in jeder Iterationsstufe aus verkleinerten Kopien dieser Figur gebildet. Der beschriebene Vorgang wird dann auf jedes einzelne Liniensegment angewandt, also jede Linie wieder in drei gleichgroße Stücke aufgeteilt und das mittlere Stück durch ein Dreieck ersetzt. In jeder weiteren Iterationsstufe wird dieser Prozess rekursiv wiederholt (siehe Abb. 6).

Die Selbstähnlichkeit ist hierbei offensichtlich. Jedes Segment ist eine exakte Kopie der originalen Kurve, so zeigt sich beim Vergrößern des ersten Viertels der zweiten Iterationsstufe in Abb. 6 wieder die vorherige erste Iterationsstufe. Es wird klar, dass sich diese Iteration unendlich wiederholen lässt und bei unendlich vielen Iterationen auch eine unendlich lange Kurve mit unendlich vielen Eckpunkten bilden wird. Die

[31] Die einzelnen Stufen wurden durch den in Kapitel 8 beschriebenen BBC BASIC-Code erstellt.

Länge dieser Kurve strebt also gegen Unendlich. Zumindest bei einzelnen Iterationsstufen und bei Gleichseitigkeit der Dreiecke lässt sich die Länge aber genau berechnen, da die Kurve in einer Iterationsstufe stets viermal so lang ist wie ein drittel der vorherigen Stufe. Grundsätzlich zeigen sich beim Generator vier Linien gleicher Länge. Im weiteren Schritt sind es 4x4 Linien, in der nächsten Stufe sind es 4x4x4 und so weiter. Wenn der Initiator, also die ursprüngliche Linie die Länge L misst, dann hat ein Segment im nächsten Schritt die Länge 1/3 x L und im darauffolgenden Schritt $1/3^2$ x L. Die Kurve setzt sich jeweils aus vier dieser Linien zusammen, weshalb der Generator eine Länge von 4 x 1/3 x L misst, in der zweiten Iterationsstufe misst die Länge dann 4^2 x $1/3^2$ x L, dies erhöht sich bei jeder weiteren Iteration. Es lässt sich daraus also ableiten:

$$L_n = L * 4^n/3^n$$

In jeder Iterationsstufe gibt es eine bestimmte Anzahl an Linien, die auch durch eine simple Formel berechnet werden kann. Der Generator besteht aus vier Linien, die zusammen eine Kochkurve ergeben, in der darauf folgenden Iterationsstufe wird jede einzelne dieser Linien wieder durch vier weitere ersetzt, wodurch die Anzahl um das vierfache steigt. Da dies in jeder weiteren Stufe wiederholt wird lässt sich ableiten:

$$A_n = 4^n$$

Der Initiator wurde im ersten Schritt in vier identische Segmente geteilt, deren Länge ein Drittel der Ursprungslinie beträgt. In jedem weiteren Iterationsschritt werden dann wieder die Segmente der vorherigen Stufe gedrittelt. Da die einzelnen Elemente jeder Iterationsstufe identischer Länge sind, lässt sich eine Formel zur Berechnung der Länge dieser Segmente bilden:

$$l_n = 3^{-n}$$

Aufgrund sonderbarer Eigenschaften wie einer unendlichen Länge werden Kurven dieser Art auch „Monsterkurven" genannt, die lange vor Mandelbrots Fraktalbegriff entdeckt wurden, von Mathematikern allerdings lange wenig beachtet wurden. Aufgrund nie endender Iterationen und einer Länge entgegen Unendlichkeit lässt sich dieses Objekt nicht aufzeichnen, sondern nur durch das Abbild einer bestimmten Iterationsstufe annähern. Erst neue Dimensionsbegriffe wie der von Hausdorff ließen diese Figuren verständlicher werden.[32]

[32] vgl. Hartmut Jürgens, Heinz-Otto Peitgen, Dietmar Saupe, *Fraktale - eine neue Sprache für komplexe Strukturen,* in: *Chaos und Fraktale*, Heidelberg 1989, S. 87-90;

7 Die Hausdorff-Dimension der Koch-Kurve

Eine Besonderheit der Fraktale ist häufig eine nicht-ganzzahlige Dimension nach dem Hausdorffschen Dimensionsbegriff. Diese Dimension lässt sich nur korrekt berechnen, wenn das Objekt, auf das sie angewendet wird, exakt selbstähnlich ist. Bei der Koch-Kurve ist dies der Fall, jeder einzelne Teilbereich ist eine genaue Kopie der gesamten Kurve. In jeder Iterationsstufe wird das Objekt in vier gleich große Teile aufgeteilt, weshalb die Anzahl (a) = 4 beträgt. Jedes Segment einer Iterationsstufe ist 1/3 so groß wie das des vorherigen Schritts, der Skalierungsfaktor beträgt also s = 3.[33]

Um die Hausdorff-Dimension dieses Fraktals zu berechnen wird nun die in Kapitel 4 vorgestellte Formel genutzt:

$$D = \frac{\log a}{\log s} \qquad D = \frac{\log 4}{\log 3} = 1{,}2618595071429149$$

Die Koch-Kurve hat eine Hausdorff-Dimension von ~ 1,262 und somit zwischen der Dimension einer Linie und der einer Fläche. Eine gebrochene Dimension zwischen eins und zwei lässt sich im Falle der Koch-Kurve durch ihre Form erklären, denn im Metrischen Sinne ist eine Kurve von unendlicher Länge nicht mehr eindimensional, aber auch keine Fläche. Die euklidische Dimension (D_E) dieser Kurve wiederum ist 2, da zur Bestimmung der Punkte dieser Linie zwei Koordinaten benötigt werden.

8 Die Koch-Kurve in BBC BASIC

Es gibt unterschiedliche Herangehensweisen, ein rekursives Objekt wie ein Fraktal in BBC BASIC berechnen und darstellen zu lassen. Durch stete Wiederholungen soll mithilfe des Programms eine Koch-Kurve dargestellt werden. Die letztlich entstehende Kurve hat eine definierte Länge und strebt nicht entgegen Unendlichkeit. Neben einem genau genannten Anfangs- und Endpunkt kann anders als bei einem wirklichen Fraktal durch die Multiplikation der Anzahl der einzelnen Linien mit der jeweiligen Länge die Gesamtlänge der Koch-Kurve berechnet werden, da bei Durchführung des Programms die Iterationsstufe angegeben werden muss und eine

---

u. Paul S. Addison, *Fractals and Chaos – An illustrated course*, London 1997, S. 16f.; u. Manfred Schroeder, *Fraktale, Chaos und Selbstähnlichkeit: Notizen aus dem Paradies der Unendlichkeit*, Heidelberg 1994, S. 8ff.

[33] vgl. dies mit den einzelnen Iterationsstufen in der Abb. 6.

entsprechende Kurve ausgegeben wird. Ein wirkliches Fraktal ergäbe erst eine unendliche Rekursion, wodurch eine Kurve entstehen würde, deren Richtung sich an jedem Punkt in einer steten Veränderung befände.

Das Programm zur Koch-Kurve in BBC BASIC:

```
10 VDU 23,22,640;512;8,16,16,0
20 REPEAT
30   INPUT "Welche Iterationsstufe"; I
40   INPUT "Welche Aufloesung (zB 1000)"; A
50   S = 0.29
60   P =I+1
70   DIM XL(20), XR(20), YL(20), YR(20)
80   XL(P) = 0
90   XR(P) = A
100  YL(P) = 500
110  YR(P) = 500
120  GOSUB 180
130  INPUT "Schoen, oder? Noch einmal? (J oder N) " Antwort$
140  VDU 12
150 UNTIL Antwort$="N" OR Antwort$="n"
160 PRINT "Danke und bis dann!"
170 END
180 IF P > 1 GOTO 210
190 LINE XL(1),YL(1),XR(1),YR(1)
200 GOTO 430
210 P = P-1
220 XL(P) = XL(P+1)
230 YL(P) = YL(P+1)
240 XR(P) = (1/3)*XR(P+1) + (2/3)*XL(P+1)
250 YR(P) = (1/3)*YR(P+1) + (2/3)*YL(P+1)
260 GOSUB 180
270 XL(P) = XR(P)
280 YL(P) = YR(P)
290 XR(P) = (1/2)*XR(P+1) + (1/2)*XL(P+1) - S*(YL(P+1)-YR(P+1))
300 YR(P) = (1/2)*YR(P+1) + (1/2)*YL(P+1) + S*(XL(P+1)-XR(P+1))
310 GOSUB 180
320 XL(P) = XR(P)
330 YL(P) = YR(P)
340 XR(P) = (2/3)*XR(P+1) + (1/3)*XL(P+1)
350 YR(P) = (2/3)*YR(P+1) + (1/3)*YL(P+1)
360 GOSUB 180
370 XL(P) = XR(P)
380 YL(P) = YR(P)
390 XR(P) = XR(P+1)
400 YR(P) = YR(P+1)
410 GOSUB 180
420 P = P+1
430 RETURN
```

Der Code beginnt mit dem Displaybefehl VDU, was für Visual Display Unit steht. In diesem Modus führen die VDU-Codes einen Modus aus, der dem des BBC Micro so ähnlich wie möglich ist. Aufgrund des Unterschiedes in der Hardware des modernen Computers im Vergleich zum BBC Micro sorgt dies zwangsläufig zu Einschränkungen in der Ausgabe.

```
10  VDU 23,22,640;512;8,16,16,0
```

Die 10 ist hierbei die Zählung der Zeile dieser Programmierung. Der Modus 23,22 ist der benutzerdefinierte Modus, hier kann die Grafikausgabe in der Programmierung frei bestimmt werden. 640 und 512 bestimmen die Anzahl der Pixel in der Weite und der Höhe. Da ein Pixel je zwei Graphikeinheiten breit und hoch ist ergibt dies eine Ausgabe von 1280x1024 Graphikeinheiten. 8 und 16 legen die Breite und Höhe der Zeichen fest, wobei die Pixelanzahl ein ganzzahlig Vielfaches der Zeichen sein sollte. In diesem Fall könnten also 80 Zeilen mit je 32 Zeichen dargestellt werden. Die nächste 16 definiert die Zahl der verfügbaren Farben. Hier entspricht sie der des BBC Micro, auch wenn in der Programmierung nur zwei Farben genutzt werden. Die letzte Angabe gibt das charset an, wobei 0 für die Einstellung weißen Texts und weißer Graphiken auf schwarzem Untergrund steht. Diese Einstellung wird vorgenommen, da die Standardeinstellung eines BBC Micro und somit auch von BBC BASIC identisch ist. Wird die 0 etwa durch 128 ausgetauscht, werden schwarze Graphiken und Schriften auf weißem Untergrund ausgegeben, was etwa für die Abbildungen der Koch-Kurve in dieser Arbeit genutzt wurde.[34]

Grundsätzlich soll das Programm den Nutzer als erstes nach den genauen Eigenschaften der zu berechnenden Koch-Kurve fragen. Die Iterationsstufe (I) und die Auflösung (A) sollen genannt werden. Die Anzahl der Graphikeinheiten in der Weite soll durch die Frage nach der Auflösung angegeben werden, dies legt die Größe der Koch-Kurve auf dem Bildschirm fest. Das Programm empfiehlt hier 1000 Graphikeinheiten zu wählen, was 500 Pixel in der Breite bedeutet, da sich diese Größe und Auflösung für die Wiedergabe am Computer eignet. Darüber hinaus wird der Nutzer nach der Darstellung der gewünschten Kurve gefragt, ob das Programm

[34] vgl. Richard Russel, *BBC BASIC for Windows – VDU Emulation,* In: http://www.bbcbasic.co.uk/bbcwin/manual/bbcwin8.html#usermode (Abrufdatum: 23.01.2018).

erneut gestartet werden soll, auf diese Weise soll die Koch–Kurve sehr einfach und schnell in unterschiedlichen Iterationsstufen angezeigt werden können. Es wird mit dem Befehl REPEAT (Zeile 20) eine Schleife ausgelöst, die so oft durchgeführt wird, bis der Nutzer durch „N“ oder „n“ eine weitere Durchführung verneint (Zeile 150). Der Displaybefehl VDU 12 (Zeile 140) löscht nach jeder Durchführung das Ausgabefenster zur festgelegten Hintergrundfarbe.

Die Zeilen 50 und 60 definieren durch S=0.29 den Punkt, der die Spitzen der Kurve darstellt[35] und mit P=I+1 die Iterationsstufe. Hierbei wird das I aus der Eingabe des Nutzers genutzt und um 1 erhöht, die Iterationsstufe 0 ist im Programm als Stufe 1 definiert sodass mit positiven reellen Zahlen gearbeitet werden kann.

Für die Berechnung und den Aufbau der Koch-Kurve in der jeweiligen Iterationsstufe wird eine Rekursion genutzt. Hierbei wird anders als etwa beim menschlichen Aufzeichnen der Koch-Kurve nicht jede Stufe einzeln aufgezeigt, gedrittelt und das Dreieck hinzugefügt, sondern die einzelnen Linien werden durch die Eckpunkte und deren genauen Koordinaten berechnet. Um etwa Iterationsstufe 3 anzuzeigen werden nicht mit einem Mal alle Seiten aller vorherigen Stufen nacheinander berechnet und gespeichert, um dann die 64 einzelnen Linien der Koch-Kurve in dritter Iteration zu erhalten. Es wird mit der Berechnung der ersten Iterationsstufe begonnen. Sobald dabei das erste, linke Segment definiert ist, durchläuft es direkt das rekursive Verfahren, um hier die Linien der zweiten Stufe zu erhalten. Dieser Vorgang wird daraufhin direkt wiederholt: Ist das erste, linke Segment definiert, wird für diese Linie schon die nächste Iterationsstufe berechnet. Eine Wiederholung findet so häufig statt, bis die vom Nutzer gewünschte Iterationsstufe erreicht ist und die ersten vier Segmente angezeigt werden. Nun geht das Programm zurück, nimmt die nächste Seite vorheriger Stufen und berechnet anhand derer die vier weiteren Segmente der Koch-Kurve. Dieser Vorgang wird wiederholt bis der Endpunkt des Initiators und somit der finalen Kurve erreicht ist.

[35] Durch eine Veränderung dieses Wertes lassen sich interessante Abweichungen der Koch-Kurve bilden, etwa indem die Spitzen sehr flach gesetzt werden (Beispielsweise bei 0.1) oder eine sehr spitze und dadurch bizarr wirkende Ausformung gewählt wird (bei 0.9).

Da das Programm jeweils Seite für Seite die jeweiligen Stufen durchläuft, bis in der letzten Iterationsstufe der erste Teil der Kochkurve berechnet wird, muss es sich auch stets nur ein Segment aus jeder Iterationsstufe gleichzeitig merken, was vor allem für speicherarme Systeme wie den BBC Micro vorteilhaft ist. Schleifen (GOSUB und GOTO) sind im Programm für die Rekursion zuständig und speichern bei jedem Durchlauf die Punktkoordinaten der Linie. Die Speicherung erfolgt hierbei durch Arrays, deren Funktionsweise einer Datenbank ähnelt. Durch den Befehl DIM wird dem Programm mitgeteilt, dass es genügend Speicherbereich für eine definierte Anzahl an Variablen reservieren soll.[36] In diesem Fall handelt es sich bei den Variablen um die einzelnen Punktkoordinaten der Segmente in den jeweiligen Iterationsstufen:

```
70 DIM XL(20), XR(20), YL(20), YR(20)
```

Die Linien werden durch die Angabe der jeweiligen Endpunkte beschrieben, also XL und YL als die X- und Y-Koordinaten des linken Endpunkts und XR und YR für die Koordinaten des rechten Punkts. Zusätzlich wird angegeben, wie viele Variablen höchstens gespeichert und daraus folgend wie viel Platz reserviert werden soll. In diesem Fall können 20 Iterationen gespeichert werden, also die Iterationsstufen 0 bis 19. Sollte eine höhere Iterationsstufe als INPUT gegeben werden, wird das Programm eine Fehlermeldung aussenden, da es nicht die entsprechenden Speicherkapazitäten freigehalten hat. Zu jeder Schleife, die das Programm durchlaufen wird, werden die entsprechenden Variablen dann eingespeichert und entsprechend indexiert, sodass bei einem Zugriff die jeweilige Iterationsstufe der Variable identifiziert werden kann.

Die Koordinaten zu den Endpunkten der Koch-Kurve sind in den Zeilen 80 bis 110 durch XL(P), XR(P), YL(P) und YR(P) definiert. Passend zur Bildschirmausgabe liegen die Y-Koordinaten bei 500, also etwa in mittlerer Höhe, die linken X-Koordinaten liegen bei 0, also am linken Bildschirmrand, während der zweite Endpunkt durch den Nutzer bestimmt wird. Mit GOSUB beginnt in Zeile 120 die Schleife, in der die Iterationen stattfinden, hierbei wird ein Unterprogramm in Zeile 180 eingeleitet. Durch den Verweis auf diese Unterprogramme kann immer wieder auf die gleiche Kodierung zurückgegriffen werden, um diese Gruppe von Befehlen nicht jedes Mal zu wiederholen, was auch bei vielen Iterationsschritten eine

[36] vgl. Richard Russel, *Chapter 14 - Grouping Data in Arrays,* in: http://www.bbcbasic.co.uk/bbcwin/tutorial/chapter14.html (Abrufdatum: 23.01.2018).

Kompaktheit des Programms gewährleistet. Sollte als Iterationsstufe 0 angegeben und somit P=1 sein, wird das Programm in Zeile 190 nur den Initiator, also eine Linie zwischen den beiden Endpunkten der Kochkurve zeichnen. Durch GOTO wird in diesem Fall direkt zur Zeile 430 verwiesen, die wiederum durch RETURN die GOSUB-Schleife beendet. Ansonsten beginnt nun bei jeder höheren Iterationsstufe in Zeile 210 die Schleife zu Bildung der Kochkurve. Anders als die Benennung der Iterationsstufen von 0 bis N beginnt das Programm mit dem Initiator als N und zählt die Iterationsstufen herunter, sodass 1 die letztlich abgebildete Stufe ist, Zeile 210 definiert somit die aktuell berechnete Ebene. Es folgt bis Zeile 410 die Teilung der Liniensegmente und deren Berechnung. Wenn die Eckpunkte des linken Segments feststehen, wird durch GOSUB 180 stets zur nächsten Rekursionsebene vorgerückt. Nach dem gleichen System folgen die weiteren drei Teile nacheinander, die jeweils berechnet und rekursiv unterteilt werden. Hierbei werden die Punkte XL(P), YL(P), XR(P) und YR(P) zu jeder Linie mithilfe der jeweiligen Punkte der vorherigen Ebene (also P+1) berechnet.

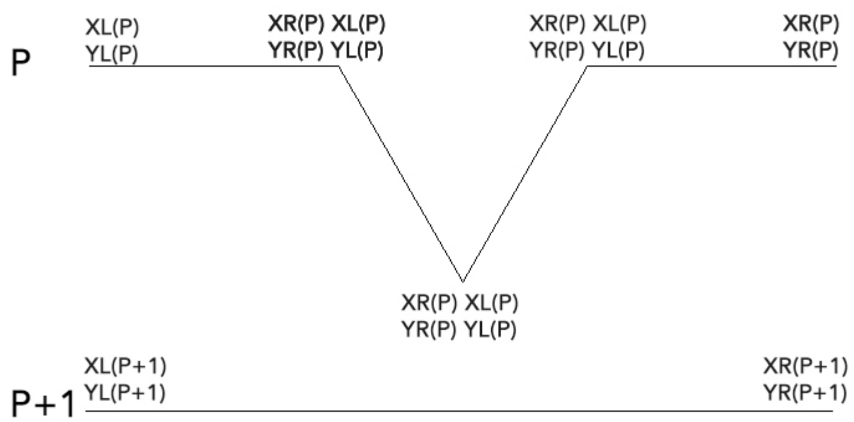


Abb. 7: Beispielhaft die aktuelle Iterationsstufe (P) und die vorherige Stufe (P+1) mit den Koordinaten der Endpunkte im rekursiven Teil des Programms

8.1 Berechnung des ersten Liniensegments

Die Koordinaten des linken Endpunkts (XL und YL) entsprechen denen des gleichen Endpunkts der entsprechenden Linie aus der vorherigen Iterationsstufe.

XL(P) = XL(P+1)

YL(P) = YL(P+1)

XR(P) = (1/3)*XR(P+1) + (2/3)*XL(P+1)

YR(P) = (1/3)*YR(P+1) + (2/3)*YL(P+1)

Der Rechte Endpunkt (XR,YR) liegt genau am Ende des ersten Drittels der Linie aus der vorherigen Stufe. Da wir uns im zweidimensionalen Raum befinden, können die jeweiligen Koordinaten dieses Punktes durch das Addieren von einem Drittel der Koordinate des rechten Punkts und zwei Dritteln des linken Punkts aus der vorherigen Stufe jeweils berechnet werden.

8.2 Berechnung des zweiten Liniensegments

Da das nächste Liniensegment am Endpunkt des vorherigen ansetzt, kann dessen rechter Punkt aus der gleichen Iterationsstufe übernommen werden.

XL(P) = XR(P)

YL(P) = YR(P)

XR(P) = (1/2)*XR(P+1) + (1/2)*XL(P+1) - S*(YL(P+1)-YR(P+1))

YR(P) = (1/2)*YR(P+1) + (1/2)*YL(P+1) + S*(XL(P+1)-XR(P+1))

Für den zweiten Endpunkt kann zunächst der Mittelpunkt der Linie aus der vorherigen Iterationsstufe ermittelt werden, indem jeweils die Hälfte von deren X- und Y-Koordinaten addiert werden. Nun wird das zu Beginn definierte S benötigt, das die Stärke der Abweichung dieses Punkts von der Linie der vorherigen Iterationsstufe festlegt. Es werden die Y-Koordinaten des rechten Punkts von den Y-Koordinaten des linken Punktes der Linie aus der vorherigen Stufe subtrahiert, um die X-Koordinaten eines Punktes zu errechnen, der abweichend von der Ursprungslinie die Spitze der Kochkurve ist. Für die Y-Koordinaten dieses Punktes werden dazu die X-Koordinaten der Endpunkte dieser Linie subtrahiert. Durch die Multiplikation mit S wird die Stärke der Abweichung von der ursprünglichen Linie festgelegt und im Falle der X-Koordinaten von denen des Mittelpunkts abgezogen, im Falle der Y-Koordinaten werden sie addiert.

8.3 Berechnung des dritten Liniensegments

Erneut ist der Endpunkt des vorherigen Segments der Anfangspunkt dieser Linie.

XL(P) = XR(P)

YL(P) = YR(P)

XR(P) = (2/3)*XR(P+1) + (1/3)*XL(P+1)

YR(P) = (2/3)*YR(P+1) + (1/3)*YL(P+1)

Der rechte Punkt (XR, YR) liegt wieder auf der Linie der vorherigen Stufe. Während er beim ersten Liniensegment am Ende des ersten Drittels der Linie liegt, wird nun das Ende des zweiten Drittels gesucht. Hierzu wird eine ähnliche Formel genutzt, nur dass zur Berechnung der Koordinaten zwei Drittel der entsprechenden Koordinaten des rechten Punkts aus der vorherigen Iterationsebene mit einem drittel der Koordinaten des linken Punkts der vorherigen Ebene addiert werden.

8.4 Berechnung des vierten Liniensegments

Für den Anfangspunkt des letzten Liniensegments können erneut die Koordinaten des Endpunkts des vorherigen Segments übernommen werden. Der Endpunkt dieses Segments ist hierbei identisch mit dem der Linie aus der vorherigen Stufe.

XL(P) = XR(P)

YL(P) = YR(P)

XR(P) = XR(P+1)

YR(P) = YR(P+1)

Nach der Durchführung der Rekursion in Zeile 430 (RETURN) geht das Programm zur Zeile 130, die auf den Beginn der Schleife folgt (GOSUB in Zeile 120). Falls der Nutzer keine erneute Berechnung der Koch-Kurve verlangt wird das Programm beendet.

8.5 Das Abspielen des Programms

Nachdem das Programm den Nutzer nach der gewünschten Iterationsstufe und Auflösung der abzubildenden Koch-Kurve fragt, läuft der rekursive Teil der Programmierung ab. Bei der Berechnung und Anzeige der Kurve in den ersten Iterationsstufen wird dabei direkt das gewünschte Fraktal angezeigt. Wenn die sechste Iterationsstufe vom Nutzer gewünscht ist, wird die Kurve zwar weiterhin nach weniger als einer Sekunde angezeigt, allerdings kann bei aufmerksamer Betrachtung festgestellt werden, dass der Aufbau am linken Bildrand beginnt. Die Koch-Kurve wird also nicht als ganzes berechnet und angezeigt, sondern die Ausgabe beginnt am linken Endpunkt und endet am rechten. Je höher die Iterationsstufe angesetzt wird, desto deutlicher lässt sich dies beobachten. Schon bei der achten Iterationsstufe liegen zwischen dem Beginn des Aufbaus und dem Erreichen des rechten Endpunkts mehr als zwei Sekunden. Wird die Iterationsstufe weiter erhöht, verlängert sich auch die Zeit, die das Programm benötigt, um die Kurve anzuzeigen, so dauert dies in der zwölften Stufe ca. 12 Minuten und 18 Sekunden. Die Anzeige lässt Rückschlüsse auf die Programmierung des Programms schließen. Durch den rekursiven Ansatz wird gleichzeitig stets ein Segment von der ersten bis zur gewünschten Stufe berechnet. Wenn das erste berechnet ist, wird es ausgegeben, woraufhin das nächste folgt. Schritt für schritt läuft das Programm so die einzelnen Elemente der jeweiligen Iterationsstufen ab, wodurch die Kurve von links nach rechts, Segment für Segment dargestellt wird, was bei der Anzeige einer höheren Iterationsstufe deutlich wird. Bei höheren Stufen verlängert sich die Zeit der Berechnung, da ein Vielfaches an Segmenten berechnet werden muss. Für die Anzeige der dritten Iterationsstufe etwa müssen nur die jeweiligen Koordinaten der Endpunkte von 64 Segmenten berechnet werden, was in einer Geschwindigkeit geschieht, die es für das menschliche Auge nicht möglich macht, den rekursiven Aufbau der Graphik zu erkennen. In der zwölften Iterationsstufe sind es dagegen schon 4^{12}, also 16.777.216 Segmente, deren 33.554.432 Eckpunkte kalkuliert und Linienverbindungen ausgegeben werden. Zum Festlegen dieser Punkte muss das Programm auch alle Eckpunkte aller vorherigen Stufen berechnen. Die unter DIM eingespeicherten Koordinaten sorgen dafür, dass auf diese Daten nach Berechnung zurückgegriffen werden kann. Der Performanz des Programms kann durch die Beobachtung der langsam aufbauenden Graphik zugesehen werden und damit gemessen werden, wie viel Zeit es zum Berechnen der Segmente benötigt. Die Geschwindigkeit des Aufbaus scheint sich dabei nicht zu verändern, so benötigt das Programm für den Aufbau der ersten Hälfte der Kurve in der zwölften Stufe auch die Hälfte der Zeit.

Aufgrund der begrenzten Auflösung der Graphikausgabe von BBC BASIC ist es nicht möglich, in höheren Iterationsstufen den Aufbau der einzelnen Segmente zu beobachten, jedoch müsste das Bild auch sehr stark vergrößert werden, da die Länge eines Segments der 12. Iterationsstufe etwa 0.0000019 mal der Länge der Ursprungslinie entspricht. Bei Höheren Iterationsstufen schränkt die Größe und Auflösung der Graphikausgabe die Unterscheidbarkeit ein (vgl. Abb 8). In der Tabelle 1 wurde vermerkt, wie viel Zeit zwischen dem Start des Programms und der Anzeige der gesamten Koch-Kurve verging, sowie die Anzahl der Elemente, deren Koordinaten dazu ermittelt werden mussten. Die Anzahl der Segmente umfasst alle Segmente der Iterationsstufe und der Stufen darunter. Als Auflösung wurde jeweils 1000 ausgewählt und die Messung fand per Hand mit einem Millisekunden-genauen Messgerät statt.

Iterationsstufe	Zeit in Min:Sek	Gesamtsegmente	Segmente/Sekunde
8	00:02:70	87.380	32.362
9	00:10	349.524	34.952
10	00:39	1.398.100	35.848
11	03:05	5.592.404	30.559
12	12:18	22.369.620	30.610
13	50:45	89.478.484	29.560

Tabelle 1: Iterationsstufen mit Zeit zum Aufbau und Anzahl zu berechnender Segmente

Aufgrund der Messungenauigkeit können die Werte in Tabelle 1 nur Richtwerte sein und gerade bei niedriger Aufbauzeit macht sich diese Ungenauigkeit bemerkbar, weshalb zur achten Iterationsstufe auch die Millisekunden hinzugenommen wurden. Schon eine Rundung auf 0,03 Minuten hätte die Zahl der Segmente, die durchschnittlich pro Sekunde ermittelt werden um mehr als 3.000 verringert. Wird die Messungenauigkeit hinzugezogen, die sich gerade bei den Stufen elf bis dreizehn prozentual verringern dürfte, lässt sich deuten, dass das Programm stets die gleiche Zeit benötigt, um die Koordinaten der einzelnen Segmente zu ermitteln. Der Rekursive Ansatz der Programmierung sorgt für einen geringen Bedarf an Zwischenspeicher, da zur Berechnung der Koordinaten die Endpunkte aus dem Segment der vorherigen Iterationsstufe und des zuvor ermittelten Endpunkts des letzten Segments der aktuellen Iterationsstufe verfügbar sein müssen. Grundsätzlich werden die Daten in verfügbaren Arrays unter DIM (Zeile 70) gespeichert. Die

Rekursion des Programms wird während der Durchführung in der Darstellung des Fraktals sichtbar.

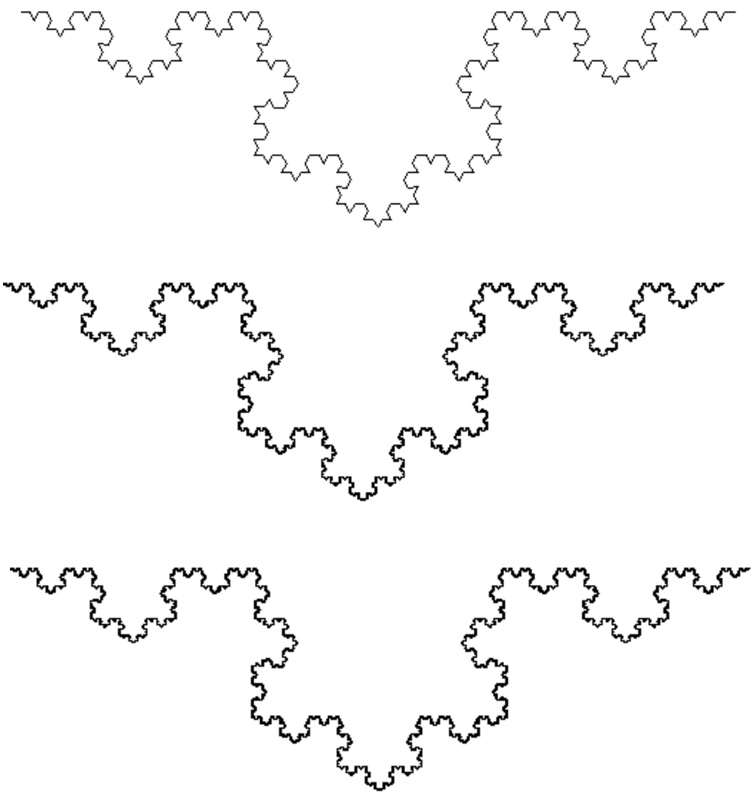


Abb. 8: Die Iterationsstufen 4, 8 und 12[37]

9 Die Arbeit mit BBC BASIC und die Entwicklung eines Fraktals

In den Ausführungen zum Hintergrund der Programmiersprache und bei der groben Erläuterung der Funktionsweise wurde die Nutzerfreundlichkeit von BASIC bereits deutlich. Als mit Programmierungen nicht unvertrauter, aber neuer Anwender dieser Sprache ist es vergleichsweise wenig aufwendig, sie zu verstehen und mit ihr arbeiten zu können. Eine Hilfe sind hierbei auch die erstaunlich detaillierten Anweisungen im Internet. Seiten wie bbcbasic.co.uk bieten zu den meisten Befehlen ausführliche Erläuterungen, weshalb bei der Anwendung der Sprache schnell die ersten Programme entstehen können. Komplexer ist die Nutzung von BASIC, bzw. in diesem Fall von BBC BASIC zur Erstellung von Fraktalen. Zwar erlaubt es das

[37] Erstellt durch den beschriebenen BBC BASIC-Code bei einer Auflösung von 1000.

Programm auf einfach Art und Weise Graphiken zu berechnen und anzeigen zu lassen, allerdings liegt im Wesen der Fraktale, als aus Iterationen hervorgebrachte, selbstähnliche Gebilde, nicht nur eine einzelne Graphik zu sein, die definiert werden muss, sondern statt der Anwendung einzelner Graphikbefehle müssen Formeln gefunden werden, die den Aufbau des Fraktals genau definieren. Diese Formeln müssen das Fraktal in jeder Iterationsebene korrekt und mit einer exakten Selbstähnlichkeit definieren und gleichzeitig so programmiert sein, dass sie für BBC BASIC lesbar sind. Die Entwicklung gänzlich neuer fraktaler Gebilde für eine Programmierung ist sehr komplex, da viele unterschiedliche Arten von Fraktalen in unterschiedlicher Form bereits gebildet wurden und es keine Grundlage gibt, die als Stütze zur Programmierung und Analyse genutzt werden kann.[38]

Bei der Erstellung eines Fraktals in BBC BASIC, zu dem noch keine direkte Programmierung vorliegt, bietet sich eine Annäherung über andere Sprachen und Dialekte an. Programme in Java etwa nutzen ebenfalls den rekursiven Ansatz und die Aufteilung einer Linie aus der vorherigen Iterationsstufe in einzelne Segmente mit der Bestimmung der zwei dazwischen liegenden Punkte und der Kalkulation des Vorsprungs aus dieser Linie durch einen dritten Punkt. In anderen BASIC-Dialekten zeigten sich alternative Ansätze, die ebenfalls eine Kochkurve hervorbrachten und so auch den Leistungsumfang dieser Sprache aufzeigen.[39] Durch die Analyse anderer Programmierungen konnte eine Annäherung an das finale Programm geschehen, wobei ein Problem am Prozess des Programmierens mit BASIC in Fehlermeldungen

---

[38] bei vielen ersten Ausführungen zeigten sich sowohl die Komplexität des Programmierens aufwändiger, geometrisch komplexer Figuren wie eben Fraktale, als auch Probleme mit der Definition des Fraktalbegriffs an sich. Die ersten, bereits sehr ausgeklügelten Programmierungen selbst gestalteter Figuren setzten auf die Form des Kreises und zeigten durch selbstähnliche Kopien einige Merkmale der Fraktale wie etwa einen zu Unendlichkeit strebenden Umfang. In einschlägiger Fachliteratur wurden kreisrunde Fraktale nicht behandelt, und es stellte sich nach Problemen in der Analyse und einer weiteren Recherche heraus, dass die Form des Kreises nicht klassisch selbstähnlich ist, bei der Vergrößerung eines Teilbereichs wird der Kreis zu einer Geraden. Aufgrund dessen kann weder eine Hausdorff-Dimension bestimmt, noch kann überhaupt von einem Fraktal gesprochen werden.

[39] interessant ist hierbei etwa das QBasic-Programm von Hans Lauwerier, das durch die Ordnungszahlen der einzelnen Liniensegmente im quaternären Zahlensystem und eine darauf aufbauende Zuordnung einer der vier möglichen Steigungsrichtungen die Koch-Kurve in beliebigen Iterationsstufen berechnet. Siehe Hans Lauwerier, *Fraktale verstehen und selbst programmieren – Band 1*, Hückelhoven 1992, S. 44ff. u. S. 172.

bei der Durchführung eines Programmes besteht. Diese Meldungen sind unbestimmt oder geben nur eine ungefähre Angabe über einen von möglicherweise vielzähligen Fehlern in der Programmierung, die für die Nichtdurchführbarkeit sorgen.

10 Fazit

Die Arbeit mit BBC BASIC zeigte wie simpel und schnell Programmierungen in dieser Sprache erstellt werden können. Ein Herantasten an eine gebräuchliche Programmiersprache der Mikrocomputer der 1980er-Jahre offenbarte, dass sich auch heute noch, in Zeiten viel leistungsfähigerer Computer und komplexer und funktionsreicherer Programmiersprachen, eine Anwendung aufgrund der Simplizität und der Möglichkeit, auch komplexere Berechnungen durchführen zu können, anbietet. Die grundsätzliche Benutzung dieser Sprache selbst erwies sich als problemlos, was auf ihre Simplizität zurückzuführen ist und die Anwendung bei Computern im Heim- und Bildungsbereich nachvollziehbar macht. Das Aufstellen der Funktionen zur Berechnung der Koch-Kurve verlangte die Möglichkeiten der Sprache über Grundsätze hinaus zu verstehen und sich noch genauer in ihre Funktionsweise hineinzuversetzen.

Die graphische Ausgabe der Koch-Kurve bietet gerade bei höheren Iterationsstufen die Möglichkeit, dem Programm bei der Berechnung insofern zuzusehen, dass die rekursiv berechneten Linien nacheinander ausgegeben werden. Die Prozessualität der ausgeführten Programmierung durch das BBC BASIC-Programm auf einem modernen Computer kann in Realzeit und messbar beobachtet werden. Wie schon Benoît Mandelbrot die Fähigkeiten des Computers nutzte, um Fraktale zu erzeugen, wird auch die Koch-Kurve in einer Geschwindigkeit berechnet und angezeigt, die für den Menschen unmöglich wäre zu erreichen. Gleichzeitig stößt das Programm aber auch an technische Grenzen. Die geringe Auflösung der Graphikausgabe verhindert in höheren Iterationsstufen, die Details der Koch-Kurve zu sehen. Die hohe Anzahl der zu leistenden Berechnungen und die endliche Leistungsfähigkeit des BBC BASIC-Programms machen die Darstellung sehr hoher Iterationsstufen zu zeitaufwändig. Auch ist es nur möglich bestimmte Iterationsstufen zu berechnen, nicht aber ein wirkliches Fraktal mit einer unendlichen Kurvenlänge. So stößt die einst auf

Mikrocomputern populäre Programmiersprache bei dieser geometrischen Form, die prädestiniert ist für die Berechnung und Darstellung durch den Computer, an Grenzen.

Literaturverzeichnis

ADDISON 1997
Addison, Paul S.: *Fractals and Chaos – An illustrated course*. London: Institute of Physics Publishing 1997

BLYTH 2012
Blyth, Tilly: *The Legacy oft he BBC Micro – effecting change in the UK's Culture of Computing*. London: Nesta 2012

BRAY 1983
Bray, Andrew C.; Dickens, Adrian C.; Holmes BA, Mark A.: *The Advanced User Guide for the BBC Microcomputer*. Cambridge 1983

BBC 1981
British Broadcasting Corporation: *BBC Continuing education Television: Computer Literacy Project*. 1981. In: *Centre of Computing History: BBC - Computer Literacy Project.* In: http://www.computinghistory.org.uk/det/7182/BBC-Computer-Literacy-Project/ (Abrufdatum 23.01.2018)

CENTRE OF COMPUTING HISTORY
Centre of Computing History: *BBC Continuing education Television: Computer Literacy Project.* In: http://www.computinghistory.org.uk/det/7182/BBC-Computer-Literacy-Project/ (Abrufdatum 23.01.2018)

DUMAS 2006
Dumas, Joseph D.: *Computer Architecture -Fundamentals and Principles of Computer Design*. Boca Raton FL: CRC Press 2006

FELDMAN 2012
Feldman, David P.: *Chaos and Fractals – An Elementary Introduction*. Oxford: Oxford University Press 2012

FERGUSON 1983
Ferguson, John D.; Gordon, John; Macari, Louie; Rushbrok Williams, Simon; Shaw, Anthony; Williams, Peter: *Programming the BBC Micro*. Kent: Newnes Technical Books 1983

GARDNER 1981
Gardner, Carl; Young, Robert M.: *Science on TV: A Critique.* In: *Popular television and film: a reader*. Hg. v. Tony Bennett. London: B.F.I. Publishing 1981, S. 173.

GAZZARD 2016
Gazzard, Alison: *Now the Chips Are Down: The BBC Micro*. Cambridge, MA: MIT Press 2016

JÜRGENS 2004
Jürgens, Hartmut; Peitgen, Heinz-Otto; Saupe, Dietmar: *Chaos and Fractals – new frontiers of science.* (2nd edition). New York: Springer 2004

JÜRGENS 1989
Jürgens, Hartmut; Peitgen, Heinz-Otto; Saupe, Dietmar: *Fraktale - eine neue Sprache für komplexe Strukturen*. In: *Chaos und Fraktale*. Heidelberg: Spektrum der Wissenschaft Verlagsgesellschaft 1989, S. 106-119.

KURTZ 1981
Kurtz, Thomas E.: *BASIC Session.* In: *History of programming languages*. Hg. v. Wexelblat, Richard L. New York: ACM Publications 1981, S. 515-537.

LAUWERIER 1992
Lauwerier, Hans: *Fraktale verstehen und selbst programmieren*. (2. Auflage). Hückelhoven: R. Wittig Buchverlag 1992. Übersetzung aus dem Niederländischen: *Fractals - Meetkundige figuren in eindeloze herhaling*. Amsterdam: Aramith Uitgevers 1987

MANDELBROT 1982
Mandelbrot, Benoît B.: *The Fractal Geometry of Nature*. New York: W.H. Freeman and Company 1982

NEIDHARDT 1993
Neidhardt, Wolfgang; Zeitler, Herbert: *Fraktale und Chaos – Eine Einführung*. Darmstadt: Wissenschaftliche Buchgesellschaft 1993

RUSSEL 2017 I
Russel, Richard: *BBC BASIC for Windows – VDU Emulation*. In: http://www.bbcbasic.co.uk/bbcwin/manual/bbcwin8.html#usermode (Abrufdatum: 23.01.2018)

RUSSEL 2017 II
Russel, Richard: *Chapter 14 - Grouping Data in Arrays.* In: http://www.bbcbasic.co.uk/bbcwin/tutorial/chapter14.html (Abrufdatum: 23.01.2018)

SANBURN 2010
Sanburn, Josh: *Dude, you have fractals on your wall*. In: Time, http://newsfeed.time.com/2010/10/18/dude-you-have-fractals-on-your-wall/ (Abrufdatum: 24.01.2018)

SANDERSON 1973
Sanderson, Peter C.: *Interactive Computing in BASIC - An introduction to interactive computing and a practical course in the BASIC language.* London: Newnes – Butterworths 1973

SCHROEDER 1994
Schroeder, Manfred: *Fraktale, Chaos und Selbstähnlichkeit - Notizen aus dem Paradies der Unendlichkeit*. Heidelberg: Spektrum Akademischer Verlag 1994. Aus dem Englischen übersetzt: *Fractals, Chaos, Power Laws*. News York: W.H. Freeman and Company 1991

VON KOCH 1904
Von Koch, Helge: *Sur une courbe continue sans tangente, obtenue par une construction géométrique élémentaire.* In: *Arkiv för matematik 1*. Stockholm: Norstedt & Soner 1904, S. 681-704.

VON KOCH 1906
Von Koch, Helge: *Une méthode géométrique élémentaire pour l'étude de certaines questions de la théorie des courbes planes*. In: *Acta Mathematica 30*. Djursholm: Institut Mittag-Leffler 1906, S. 145–174.