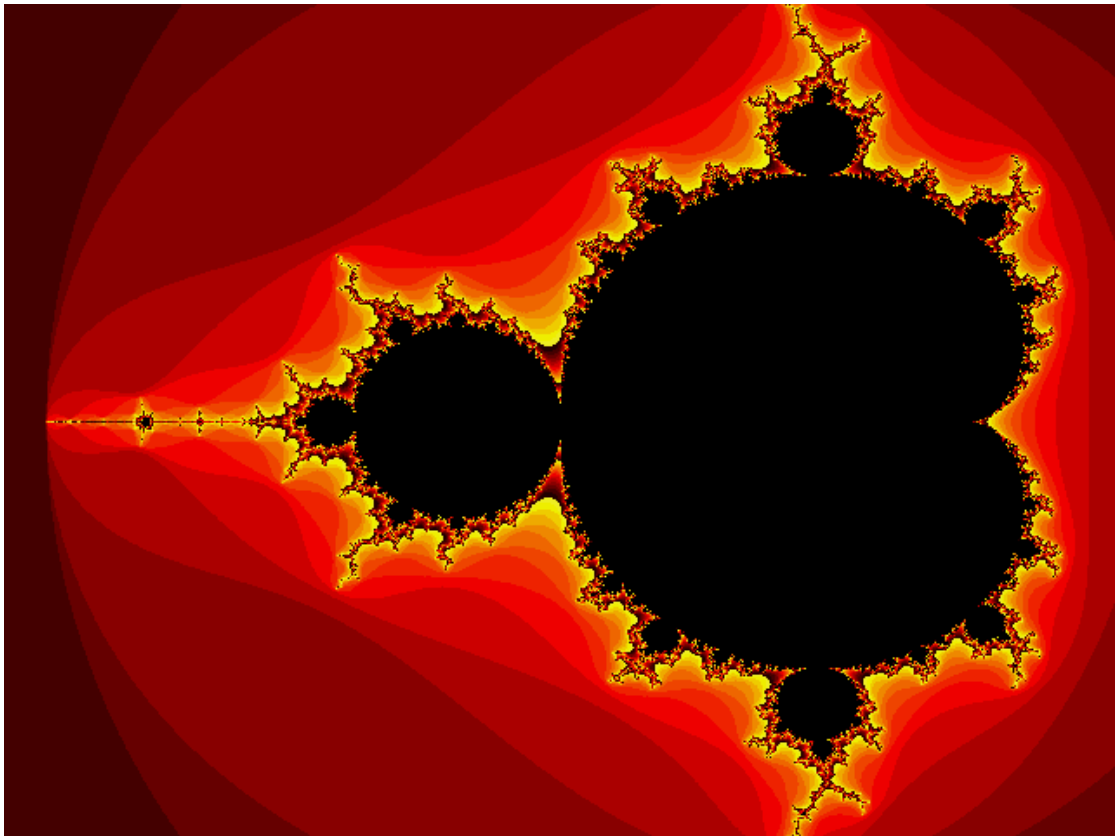


Werkstatt Multiplikation  
Posten: **Alice in Mandelland**

Informationsblatt für die  
Lehrkraft

# Alice in Mandelland



<b>Thema:</b>	Mandelbrotmenge
<b>Schultyp:</b>	Mittelschule, technische Berufsschule, Fachhochschule
<b>Vorkenntnisse:</b>	Grundlagen der komplexen Zahlen, Elementare Programmierkenntnisse
<b>Bearbeitungsdauer:</b>	90 Minuten
<b>Fassung vom:</b>	23.9.95
<b>Schülerprobung:</b>	nein

## Übersicht

Jeder von uns hat schon die faszinierenden Bilder der Mandelbrotmenge gesehen. Bilder, welche immer komplexer werden, je näher man sie sich anschaut. Sie eignen sich gut für attraktive T-Shirts und Poster und fesseln die Vorstellungskraft von Mathematikern und Laien gleichermaßen. Aber was steckt genau dahinter?

Der Posten macht erfahrbar, wie diese Bilder zustande kommen und gibt einen gewissen mathematischen Hintergrund. Die Schülerinnen und Schüler studieren zuerst die Theorie. Dann führen sie einfache Iterationen "von Hand" durch. Schliesslich implementieren sie den Algorithmus auf ihrem Computer.

## Lernziele

Die Mandelbrotmenge gehört zu den zahlreichen Dingen, denen wir im Umfeld von Mathematik und Computer immer wieder begegnen, ohne sie wirklich zu verstehen. Hier wird einerseits etwas mathematische Allgemeinbildung betrieben, andererseits wird ein Algorithmus vorgestellt, der in der Informatik oft als Benchmark für neue Prozessoren benutzt wird.

Nach Bearbeiten des Postens wissen die Schülerinnen und Schüler wie die Bilder der Mandelbrotmenge entstehen. Sie können das Iterationsverfahren an einfachen Beispielen selbst "von Hand" anwenden und sind auch in der Lage, den Algorithmus auf einem Computer zu implementieren

## Material

- *Theorie:* Alice in Mandelland - Das Geheimnis der Mandelbrotmenge
- *Computer* (für Auftrag 3)

## Quellen

Ewing J.: *Can We See the Mandelbrot Set?* - The College Mathematics Journal, Vol. 26, No. 2, March 1995

## Hinweise, Lösungen

### Lösung Auftrag 2

Auf dem Taschenrechner wird die Iteration für  $c = a + bi$  folgendermassen berechnet:

$$a := a_0 \quad / \quad b := b_0$$

WIEDERHOLE:

$$a_{\text{Neu}} = a^*a - b^*b + a_0$$

$$b_{\text{Neu}} = 2^*a^*b + b_0$$

$$a := a_{\text{Neu}}$$

$$b := b_{\text{Neu}}$$

a) (Fixpunkt)

$$0: a = 0.100 \quad / \quad b = 0.200$$

$$1: a = 0.070 \quad / \quad b = 0.240$$

$$2: a = 0.047 \quad / \quad b = 0.234$$

$$3: a = 0.048 \quad / \quad b = 0.222$$

$$4: a = 0.053 \quad / \quad b = 0.221$$

$$5: a = 0.054 \quad / \quad b = 0.223$$

$$6: a = 0.053 \quad / \quad b = 0.224$$

$$7: a = 0.053 \quad / \quad b = 0.224$$

b) (2er-Zyklus)

$$0: a = -0.900 \quad / \quad b = 0.100$$

$$1: a = -0.100 \quad / \quad b = -0.080$$

$$2: a = -0.896 \quad / \quad b = 0.116$$

$$3: a = -0.110 \quad / \quad b = -0.108$$

$$4: a = -0.900 \quad / \quad b = 0.124$$

$$5: a = -0.106 \quad / \quad b = -0.123$$

$$6: a = -0.904 \quad / \quad b = 0.126$$

$$7: a = -0.099 \quad / \quad b = -0.128$$

$$8: a = -0.907 \quad / \quad b = 0.125$$

$$9: a = -0.094 \quad / \quad b = -0.127$$

$$10: a = -0.907 \quad / \quad b = 0.124$$

$$11: a = -0.092 \quad / \quad b = -0.125$$

$$12: a = -0.907 \quad / \quad b = 0.123$$

$$13: a = -0.092 \quad / \quad b = -0.123$$

$$14: a = -0.907 \quad / \quad b = 0.123$$

$$15: a = -0.093 \quad / \quad b = -0.123$$

$$16: a = -0.906 \quad / \quad b = 0.123$$

$$17: a = -0.094 \quad / \quad b = -0.123$$

$$18: a = -0.906 \quad / \quad b = 0.123$$

$$19: a = -0.094 \quad / \quad b = -0.123$$

- c) (anfangs scheinbar chaotisch, nach ca. 1000 Iterationen jedoch ein 5er-Zyklus)
- 995: a= -0.510 / b= 0.592  
996: a= -0.591 / b= -0.004  
997: a= -0.150 / b= 0.605  
998: a= -0.843 / b= 0.418  
999: a= 0.036 / b= -0.105  
1000: a= -0.510 / b= 0.592  
1001: a= -0.591 / b= -0.004  
1002: a= -0.150 / b= 0.605  
1003: a= -0.843 / b= 0.418
- d) (geht gegen Unendlich)
- 0: a= -0.400 / b= 0.800  
1: a= -0.880 / b= 0.160  
2: a= 0.349 / b= 0.518  
3: a= -0.547 / b= 1.162  
4: a= -1.450 / b= -0.471  
5: a= 1.481 / b= 2.166  
6: a= -2.898 / b= 7.216  
7: a= -44.063 / b= -41.028  
8: a= 257.913 / b= 3616.439

### Lösung Auftrag 3:

Im folgenden ein PASCAL-Programm, welches die Mandelbrotmenge darstellt. Es lässt sich einfach auch auf andere Sprachen umschreiben. Der einzige maschinenabhängige Teil ist die Initialisierung eines Grafikfensters und die Ausgabe eines einzelnen Pixels in einer bestimmten Farbe.

```
CONST
  MaxDepth = 150; (* Max. Iterationstiefe *)
  HiColor = 15; (* Höchste Farbnummer *)

  xResolution = 320; (* Bildschirm-Auflösung *)
  yResolution = 240;

  xBegin = -2.1; (* Ausschnitt aus der *)
  xEnd = 0.6; (* Mandel-brotmenge *)

  yBegin = -1.1;
  yEnd = 1.1;

  Dx : REAL = (xEnd- xBegin) / xResolution;
  Dy : REAL = (yEnd- yBegin) / yResolution;

VAR
  xScreen, yScreen : WORD;
  xMandel, yMandel : REAL;
```

```
FUNCTION Col or(x0, y0: REAL): WORD;
VAR
  x, y          : REAL;
  xNeu, yNeu   : REAL;
  x2, y2       : REAL;
  debth        : WORD;

BEGIN
  x := x0;  y := y0;  debth := 0;

  REPEAT
    x2 := x*x;  y2 := y*y;
    xNeu := x2 - y2 + x0;
    yNeu := 2*x*y + y0;
    x := xNeu;  y := yNeu;
    INC(debth);
  UNTIL (x2+y2 > 4) OR (debth > MaxDebth);

  IF debth > MaxDebth THEN Col or := 0
  ELSE Col or := (debth MOD Hi Col or) + 1;

END; (* Col or *)

BEGIN
  InitGraph;  (* Grafik-Modus initialisieren *)
  yScreen := 0;
  yMandel := yBegin;

  WHILE yScreen < yResolution DO
    BEGIN (* Linie für Linie *)
      xScreen := 0;
      xMandel := xBegin;
      WHILE xScreen < xResolution DO
        BEGIN (* Punkte auf einer Linie *)
          PutPixel(xScreen, yScreen,
            Col or(xMandel, yMandel));
          xScreen := xScreen + 1;
          xMandel := xMandel + Dx;
        END; (* WHILE *)
        yScreen := yScreen + 1;
        yMandel := yMandel + Dy;
      END; (* WHILE *)
    END;

END. (* MANDEL *)
```

## Lehrer-Lernkontrolle / Test

### Aufgabe 1

Gehört der Punkt  $c = 0.1 + 0.1i$  zur Mandelbrotmenge? Notiere die einzelnen Iterationsschritte und interpretiere das Resultat.

### Aufgabe 2

Nach wie vielen Iterationen kann man mit Sicherheit entscheiden, ob ein Punkt zur Mandelbrotmenge gehört oder nicht?

## Lösungen und Taxierung

### Aufgabe 1

0:  $a = 0.100$  /  $b = 0.100$   
1:  $a = 0.100$  /  $b = 0.120$   
2:  $a = 0.096$  /  $b = 0.124$   
3:  $a = 0.094$  /  $b = 0.124$   
4:  $a = 0.093$  /  $b = 0.123$   
5:  $a = 0.094$  /  $b = 0.123$   
6:  $a = 0.094$  /  $b = 0.123$

Der "c-Prozess" mündet in den Fixpunkt  $0.094 + 0.123i$ . Der ursprüngliche Punkt  $c = 0.1 + 0.1i$  gehört also zur Mandelbrotmenge.

Die Aufgabe verlangt vom Schüler eine weitere Anwendung des Gelernten. Es wird hauptsächlich Wissen und etwas Verständnis gefragt. Die Aufgabe ist deshalb als **K2** zu taxieren.

### Aufgabe 2

Sobald ein Wert im "c-Prozess" den Radius 2 übersteigt, strebt der Prozess sicher gegen Unendlich. Der Punkt gehört dann nicht zur Mandelbrotmenge. Falls der "c-Prozess" in einen Fixpunkt oder Zyklus gerät, gehört der ursprüngliche Punkt sicher zur Mandelbrotmenge. In allen anderen Fällen kann man keine Aussage machen. Man muss weiterrechnen, bis einer der obigen Fälle eintritt. Daher kann man nie im Voraus sagen, wie viele Iterationen es braucht, um zu entscheiden, ob ein Punkt zur Mandelbrotmenge gehört oder nicht.

Die Aufgabe bedingt Verständnis und Analyse. Sie ist als **K3** zu taxieren.

## Was soll ich hier tun?

Die Mandelbrotmenge gehört zu den zahlreichen Dingen, denen wir im Umfeld von Mathematik und Computer immer wieder begegnen, ohne sie wirklich zu verstehen.

Hier entdeckst Du, welche mathematischen Regeln und Eigenschaften dahinterstecken. Schliesslich wirst Du auch in der Lage sein, selber Bilder der Mandelbrotmenge auf deinem Computer zu zeichnen.

Der Posten besteht aus den folgenden drei Aufträgen:

- (1) Studiere die **Theorie** "*Das Geheimnis der Mandelbrotmenge*"  
(15-20 Minuten)
- (2) Versuche die vier Beispiele im Theorieteil "von Hand" nachzuvollziehen. Zerlege dazu erst die Formel des "c-Prozesses" zur Berechnung der Iterationen auf dem Taschenrechner in Real- und Imaginärteil. Dann schreibe Dir alle Zwischenresultate auf.  
(10-15 Minuten)
- (3) Implementiere auf Deinem Computer ein Programm, welches die Mandelbrotmenge darstellt. Welche Ausschnitte führen zu den schönsten Bildern?  
(45-90 Minuten)

Den Theorieteil sollte jeder für sich studieren. Die Aufträge (2) und (3) kannst Du, wenn Du willst, auch in Gruppen (2 oder 3) bearbeiten.

## Das Geheimnis der Mandelbrotmenge

Jeder von uns hat schon die faszinierenden Bilder der Mandelbrotmenge gesehen. Bilder, welche immer mehr Details zeigen, je näher man sie sich anschaut. Sie eignen sich gut für attraktive T-Shirts und Poster und fesseln die Vorstellungskraft von Mathematikern und Laien gleichermaßen. Aber was steckt genau dahinter?

Die Bilder der Mandelbrotmenge kommen erstaunlicherweise durch eine sehr einfache mathematische Formel zustande. Für die Definition dieser Formel benötigen wir lediglich einige Grundlagen über komplexe Zahlen und die Tatsache, dass sich komplexe Zahlen auch als Punkte in einer Ebene darstellen lassen.

Für jede komplexe Zahl  $c$  können wir einen “ $c$ -Prozess” definieren. Wir beginnen mit  $c$ , quadrieren und addieren  $c$ , quadrieren das Resultat und addieren wieder  $c$  und so weiter:

$$c \quad > \quad c^2 + c \quad > \quad (c^2 + c)^2 + c \quad > \quad . . .$$

Am besten, Du experimentierst mit einigen Werten für  $c$  auf einem Taschenrechner. Hier einige Beispiele:

- Bei  $c = 0.1 + 0.2i$  erreicht der “ $c$ -Prozess” bald den Wert  $0.053 + 0.224i$  und bleibt stecken, man spricht von einem Fixpunkt.
- Bei  $c = -0.9 + 0.1i$  beginnt der “ $c$ -Prozess” nach einigen Iterationen zwischen den beiden Werten  $-0.094 - 0.123i$  und  $-0.906 + 0.123i$  zu oszillieren.
- Bei  $c = -0.5 + 0.6i$  wandert der “ $c$ -Prozess” lange hoffnungslos herum, ohne ein erkennbares Muster zu zeigen.
- Bei  $c = -0.4 + 0.8i$  erreicht der “ $c$ -Prozess” nach einigen Iterationen immer grössere Werte. Nach einem guten Dutzend Iterationen gibt der Taschenrechner langsam den Geist auf: die Werte werden grösser als  $10^{100}$ .

Was genau bei jedem  $c$  passiert ist schwer vorauszusehen und scheint manchmal auch etwas mysteriös. Findest Du noch andere Zusammenhänge? Ein Aspekt des Verhaltens ist jedoch immer sehr simpel: Entweder der “ $c$ -Prozess” gerät in irgendeine Zyklen oder er geht schliesslich gegen Unendlich.

## Von der Formel zum Bild

Die Mandelbrotmenge umfasst alle Punkte, für die der “ $c$ -Prozess” endlich bleibt. Können wir dies aber für jeden Punkt abschliessend entscheiden? Die Antwort ist etwas unbefriedigend: Bei vielen Punkten sind wir unsicher. Es ist oft einfacher zu entscheiden, ob der “ $c$ -Prozess” bei einem Punkt schliesslich gegen Unendlich geht. Man kann zeigen, dass dies der Fall ist, sobald der “ $c$ -Prozess” sich einmal weiter als 2 vom Nullpunkt entfernt hat.

Mit diesem Wissen können wir nun daran gehen, auf dem Computer ein Programm zu implementieren, welches ein Bild der Mandelbrotmenge zeichnet:

- Jede komplexe Zahl  $c = a + bi$  wird als Punkt  $(a, b)$  in der Ebene betrachtet.
- Unterteile einen Ausschnitt der Ebene (etwa Radius 2 um den Nullpunkt) in kleine Quadrate (Pixel).
- Benutze das Zentrum jedes Pixels als Startwert für den “ $c$ -Prozess”.



## Werkstatt Multiplikation Posten: **Alice in Mandelland**

## Theorie

- Führe für jedes Pixel den “c-Prozess” durch und zähle die Anzahl Iterationen bis der Wert den Radius zwei überschreitet. Brich nach etwa hundert erfolglosen Versuchen ab.
- Färbe das Pixel je nach Anzahl Iterationen mit einer anderen Farbe ein. Wurde der “c-Prozess” erfolglos abgebrochen, färbe das Pixel schwarz.

Farbig sind nun alle Pixel, die sicher nicht zur Mandelbrotmenge gehören. Bei allen schwarzen Pixeln, ist man sich nicht so sicher. Man weiss jedoch, dass sicher die gesamte Mandelbrotmenge darin enthalten ist.

Die Aufgabe ist relativ einfach zu programmieren, führt aber zu hohem Rechenaufwand im Fliesskommabereich. Aus diesem Grund wird die Berechnung von Bildern der Mandelbrotmenge oft auch als attraktiver Benchmark (Geschwindigkeitstest) für neue Prozessoren benutzt.