

Rechnen

Warum selber rechnen – es gibt doch Computer?

Teilnehmer:

Christian Draeger	Andreas-Oberschule
Dennis Menge	Heinrich-Hertz-Oberschule
Patrick Mikut	Immanuel-Kant-Oberschule
Andre Rozek	Herder-Oberschule
Patrick Klitzke	Andreas-Oberschule
Christopher Schmidt	Immanuel-Kant-Oberschule
Robert Schlämann	Immanuel-Kant-Oberschule

Gruppenleiter:

Falk Ebert
Technische Universität Berlin,
Mitglied im DFG-Forschungszentrum MATHEON
„Mathematik für Schlüsseltechnologien“

1 Einleitung

Wenn man einmal die Schüler eines Mathematikurses beobachtet, dann stellt man schnell fest, dass sie ihre Taschenrechner öfter benutzen als ihren Kopf. Wozu man sich auch die Mühe machen, einen Wert wie $(2^{50} + 1) - 2^{50}$ selbst auszurechnen, wenn der Taschenrechner schon griffbereit liegt?

Warum man seinem Taschenrechner nicht immer trauen kann, haben wir in diesem einwöchigen Kurs erfahren. Dazu muss man ersteinmal wissen, wie ein Rechner überhaupt rechnet, weshalb wir uns zuerst mit der Maschinearithmetik befasst haben. Weiterhin haben wir die möglichen Rechenoperationen betrachtet, denn für einen Computer sind Addition und Multiplikation mit relativ geringem Aufwand möglich. Wie man aber die Division und die Exponentialfunktion oder auch trigonometrische Funktionen nur auf diese beiden Operationen zurückführen kann, wollen wir in unserem Bericht zeigen. Um dies zu ermöglichen, benutzen wir Approximationen mit dem Newton-Verfahren und mit Taylor-Polynomen. Neben der Möglichkeit, Werte selbst zu errechnen, haben wir gezeigt, wie es durch Interpolation von Tabellen möglich ist, alle Werte exakt und effizient zu bestimmen.

2 Maschinearithmetik

Es gibt verschiedene Zahlensysteme. Das gängigste für uns ist das Dezimalsystem. In diesem System stehen uns die Ziffern 0 bis 9 zur Verfügung. Einem Computer bzw. einem prozessorgesteuerten Gerät stehen aber nur zwei Zustände zur Verfügung, nämlich an und aus. Diese Zustände werden üblicherweise durch die Ziffern 0 (für aus) und 1 (für an) dargestellt. Das Zahlensystem, das nur aus den Ziffern 0 und 1 besteht wird als Dual- oder auch Binärsystem bezeichnet.

2.1 Darstellung von Dualzahlen

Da die Darstellung und Umrechnung von Ganzzahlen im Dualsystem allgemein bekannt ist, möchten wir in unserem Bericht nur auf die Darstellung von Fließkommazahlen eingehen. Eine typische normierte Fließkommazahl wird folgendermaßen beschrieben:

$$\pm 0,1m_2m_3\dots m_{53} \cdot 2^{e_{10}e_9\dots e_0-1023}$$

Hierbei handelt es sich um ein 64bit-System. Das erste Bit wird für das Vorzeichen der Dualzahl verwendet: Eine 0 bedeutet, dass die Zahl positiv ist, eine 1 dagegen, dass sie negativ ist. Die Mantisse der Dualzahl beginnt mit 0,1. Dies ist eine Festlegung, um die Eindeutigkeit der Zahlen zu garantieren. Danach folgen 52 Ziffern der Mantisse, welche im Bereich von $0,5$ bis $1 - 2^{-53}$ liegt. Desweiteren

stehen für den Exponenten 11 Bit zur Verfügung. Von der sich daraus ergebene Zahl wird dann 1023 subtrahiert. Der Exponent ist dann eine Ganzzahl, die sich im Bereich von 1024 bis -1023 befindet.

Desweiteren gibt es folgende Festlegungen:

$$m_2..m_{53} = 0 \wedge E = 0 \rightarrow 0 \quad (1)$$

$$m_2..m_{53} = 0 \wedge E = 2047 \rightarrow \pm\infty \quad (2)$$

$$\text{mindestens ein } m_i \neq 0 \wedge E = 2047 \rightarrow NaN(\text{NotaNumber}) \quad (3)$$

E bezeichnet hierbei die durch e_0 bis e_{10} gebildete Zahl und NaN eine Zahl, die gar keine ist, wie z.B. $\frac{0}{0}$.

Im weiteren Verlauf dieses Kapitels, wird das 8Bit-System verwendet, wo die Mantisse 5 Stellen und der Exponent 2 Stellen besitzt. Damit lassen sich beispielsweise positive Zahlen im Bereich von $0,10000 \cdot 2^{-11} = 0,0625$ bis $0,11111 \cdot 2^{+11} = 7,75$ darstellen.

2.2 Maschinengenauigkeit

Jeder Computer macht beim Rechnen Fehler. Mathematisch bezeichnet man diesen Fehler wie folgt:

$$eps = \frac{b}{2} \cdot b^{-\mu},$$

wobei μ die Anzahl der Stellen der Mantisse und b die Basis des Zahlensystems angibt. Speziell für das Dualsystem gilt:

$$eps = \frac{2}{2} \cdot 2^{-5} = \frac{1}{32}$$

Im 8 Bit System ist die Maschinengenauigkeit also etwas mehr als eine Dezimalstelle. Allgemein gilt Folgendes:

1. $rd(x) = x(1 + \varepsilon_1)$
2. $x \otimes y = (x \times y)(1 + \varepsilon_2), \quad \times \in \{+, -, \cdot, \div\}$
3. $|\varepsilon_{1/2}| \leq eps$

Dies heißt, dass es sowohl zu Fehlern bei der Darstellung von Dezimalzahlen in dualen Zahlen kommt, als auch bei jeder möglichen Kombination von mehreren Dualzahlen. Diese relativen Fehler können maximal so groß sein, wie die Maschinengenauigkeit.

Beispielumwandlung Nun wollen wir die Maschinengenauigkeit anhand der Umwandlung von π von der dezimalen Zahl in die duale Zahl durch das 8Bit-System demonstrieren. Durch die Maschiengenauigkeit von $\frac{1}{32}$ ergibt sich, dass π zwischen $\frac{31}{32}\pi \approx 3,04$ und $\frac{33}{32}\pi \approx 3,24$ liegt. Zuerst berechnen wir den Exponenten. Es wird bald ersichtlich, dass π zwischen $0,5 \cdot 2^2 = 2$ und $0,5 \cdot 2^3 = 4$ liegt . Demnach ist der Exponent 2. Man kann dies auch durch den Logarithmus berechnen:

$$\begin{aligned} 0.5 \cdot 2^x &= \pi \\ \log(0.5 \cdot 2^x) &= \log(\pi) \\ \log(0.5) + x \cdot \log(2) &= \log(\pi) \\ x &= \frac{\log(\pi) - \log 0.5}{\log(2)} \\ x &\approx = 2.65 \end{aligned}$$

Hierbei wird abgerundet, da die eigentliche Mantisse größer als 0,5 sein kann. Nun muss nur noch die Mantisse berechnet werden. Dazu teilen wir die Zahl durch $2^{Exponent}$. Wir erhalten rund 0,7854. Nun wird diese Zahl mit zwei multipliziert. Wenn Sie größer als eins ist, ist die nächste Stelle der Mantisse ¹ eins und eins wird abgezogen, anderenfalls null. Dies wird nun solange gemacht, bis alle Mantissenstellen gefüllt sind:

$$\begin{aligned} \pi : 2^2 &\approx 0,7854 \\ 0,7854 \cdot 2 &= 0,5708 + 1 \\ 0,5708 \cdot 2 &= 0,1416 + 1 \\ 0,1416 \cdot 2 &= 0,2832 + 0 \\ 0,2832 \cdot 2 &= 0,5664 + 0 \\ 0,5664 \cdot 2 &= 0,1328 + 1 \\ 0,1328 \cdot 2 &= 0,2656 + 0 \\ 0,11001|0 \end{aligned}$$

Da das carry-Bit null ist, muss nicht gerundet und auch nicht normalisiert werden. Wir erhalten nun $0,11001 \cdot 2^{10} = 0,7854 \cdot 2^2 = 3.125$. Wir erhalten also 3.125 im 8Bit System für π .

2.3 Addition mit Dualzahlen

Vorgehensweise Um zwei Dualzahlen miteinander addieren zu können, müssen sie den gleichen Exponenten besitzen. Der kleinere wird an den größeren ange-

¹In unserem Fall die erste

glichen, indem die Mantisse der kleineren Zahl entsprechend oft durch 2 dividiert (also das Komma nach links verschoben) wird. Dieser Vorgang wird 'Shiften' genannt und ist auch in die andere Richtung möglich. Addiert man nun die Mantissen, muss die Dualzahl eventuell normalisiert werden, denn durch Überträge kann diese von der üblichen Darstellungsweise abweichen (z.B. $1, m_2 m_3 \dots$). Dies geschieht ebenfalls durch eine additive Veränderung des Exponenten bei entsprechender Multiplikation bzw. Division der Mantisse mit 2. Anschließend wird die Mantisse gerundet, was mit Hilfe des sogenannten 'carry-Bits' geschieht. Das carry-Bit wird nur kurzzeitig gespeichert und gibt die 6. Stelle der Mantisse an. Beträgt das carry-Bit 0, so wird abgerundet, anderenfalls aufgerundet. Die Aufrundung kann dazu führen, dass die Mantisse gleich 1 wird und ein weiteres mal normalisiert werden muss.

Beispielrechnung Nehmen wir nun ein einfaches Beispiel: Gegeben seien die zwei Zahlen 3 und 1,4375. Wenn wir diese in Binärzahlen umwandeln, ergibt sich $0,11000 \cdot 2^{10}$ und $0,10111 \cdot 2^{01}$. Die kleinere Zahl wird nun nach rechts verschoben: $0,010111 \cdot 2^{10}$. Nun addieren wir diese:

$$\begin{array}{r} 0,11000 \quad 2^{10} \\ +0,010111 \quad 2^{10} \\ \hline 1,00011|1 \quad 2^{10} \end{array}$$

Da nun die Mantisse größer als eins ist, wird sie normalisiert. Wir verschieben die Mantisse um eins nach rechts² und erhöhen den Exponenten um eins: $0,1000111 \cdot 2^{11}$. Da das carry-Bit³ 1 ist, wird aufgerundet. Daraus ergibt sich: $0,10010 \cdot 2^{11}$. Da die Mantisse nicht gleich eins ist, muss nicht noch einmal normalisiert werden. Wir erhalten $0,10010 \cdot 2^{11} = 4,5$. Der richtige Wert beträgt 4,4375. Der Fehler entsteht aufgrund der Maschinengenauigkeit.

2.4 Multiplikation von Dualzahlen

Vorgehensweise Die Dualzahlen werden multipliziert, indem die Exponenten miteinander addiert und die Mantissen miteinander multipliziert werden. Wie bei der Addition wird die Dualzahl nun mit einer geeigneten Shiftoperation normalisiert, anschließend gerundet und ggf. ein weiteres mal normalisiert.

²Division durch zwei

³6. Stelle

Beispielrechnung Auch an einer dieser Stelle ist eine Beispielsrechnung für die Anschaulichkeit und dem besseren Verständnis sehr hilfreich.

Aus Gründen der Einfachheit nehmen wir wieder die gleichen Zahlen wie auch beim Additionsbeispiel. Wir multiplizieren 3 mit 1,4375 im 8Bit-System:

$$\begin{aligned} 0,10111 \cdot 0,11000 &= 0,10111 \cdot 0,1 + 0,10111 \cdot 0,01 \\ &= 0,10001|01 \end{aligned}$$

Jetzt muss gerundet werden: Dabei wird die sechste und siebte Stelle der Mantisse abgeschnitten. Es wird nicht aufgerundet, da das carry-Bit null ist. Anschließend müssen die Exponenten noch addiert werden. Es ergibt sich dann $0,10001 \cdot 2^{10+01} = 0,10001 \cdot 2^{11} = 0.53125 \cdot 2^3 = 4.25$. Dieser stimmt gerundet mit der ersten Stelle des tatsächlichen Wertes von 4,3125 überein.

3 Approximation von Funktionswerten

Wir werden im Folgenden einige Techniken zum näherungsweise Auswerten von analytischen Funktionen entwickeln. Die Standard-Methode dazu ist die Taylor-Entwicklung. Andere spezielle Funktionswerte - insbesondere die Werte von Umkehrfunktionen - lassen sich mit Hilfe von Nullstellensuch bestimmen. Zu diesem Zweck wird das Newton-Verfahren vorgestellt.

3.1 Taylor-Entwicklung

Wir betrachten eine genügend oft differenzierbare Funktion f in der Nähe des Punktes x_0 . Es gilt dann, wenn man genügend nahe an x_0 bleibt, dass $f(x) \approx f(x_0)$. Dabei setzen wir $p_0(x) = f(x_0)$. Dies ist eine konstante Funktion. Eine bessere Näherung an f in der Nähe von x_0 ist zu erwarten, wenn man statt einer konstanten eine lineare Funktion p_1 zulässt. Dabei fordern wir zusätzlich zu $p_1(x_0) = f(x_0)$, dass $p_1'(x_0) = f'(x_0)$. Diese Gerade ist die Tangente an die Funktion f im Punkt x_0 . Eine Gleichung für p_1 ergibt sich mit der Grenzwertbetrachtung

$$\lim_{x \rightarrow x_0} \frac{f(x) - p_0(x)}{x - x_0}. \quad (4)$$

Zähler und Nenner dieses Ausdrucks konvergieren gegen 0, da $p_0(x) = f(x_0)$. Mit dem Satz von *L'Hospital* erhält man

$$\lim_{x \rightarrow x_0} \frac{f(x) - p_0(x)}{x - x_0} = \lim_{x \rightarrow x_0} \frac{f'(x) - 0}{1 - 0} = f'(x_0).$$

Ausmultiplizieren liefert

$$\lim_{x \rightarrow x_0} f(x) = \lim_{x \rightarrow x_0} f(x_0) + f'(x_0)(x - x_0) \stackrel{\text{def}}{=} \lim_{x \rightarrow x_0} p_1(x).$$

Diese Gerade stimmt in x_0 mit f sowohl im Wert als auch in der ersten Ableitung überein. Wir versuchen jetzt eine quadratische Parabel zu finden, die mit f in Wert, erster und zweiter Ableitung übereinstimmt. Dazu betrachten wir die Differenz $f(x) - p_1(x)$. An der Stelle x_0 ist diese Differenz null und auch die erste Ableitung. $f(x) - p_1(x)$ hat also einen kritischen Punkt an x_0 und verhält sich in einer kleinen Umgebung wie $\alpha(x - x_0)^2$. Um die Zahl α zu bestimmen, betrachten wir

$$\lim_{x \rightarrow x_0} \frac{f(x) - p_1(x)}{(x - x_0)^2}.$$

Die mehrfache Anwendung des Satzes von *L'Hospital* erhält man

$$\begin{aligned} \lim_{x \rightarrow x_0} \frac{f(x) - p_1(x)}{(x - x_0)^2} &= \lim_{x \rightarrow x_0} \frac{f(x) - (f(x_0) + f'(x_0)(x - x_0))}{(x - x_0)^2} & (5) \\ &= \lim_{x \rightarrow x_0} \frac{f'(x) - f'(x_0)}{2(x - x_0)} \\ &= \lim_{x \rightarrow x_0} \frac{f''(x)}{2} = \frac{f''(x_0)}{2}. \end{aligned}$$

Ausmultiplizieren liefert

$$\lim_{x \rightarrow x_0} f(x) = \lim_{x \rightarrow x_0} f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2 \stackrel{\text{def}}{=} \lim_{x \rightarrow x_0} p_2(x).$$

Mit der analogen Vorgehensweise wie in (4) und (5) erhält man für beliebige n

$$p_n(x) = \frac{f(x_0)}{0!} + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \dots + \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n. \quad (6)$$

Diese Näherung an f in der Nähe von x_0 nennt sich *Taylor-Polynom*.

Die Genauigkeit der Approximation von f durch p_n zeigen wir im Folgenden. Es gilt

$$(f(x) - p_n(x))^{(n+1)} = f^{(n+1)}(x) - p_n^{(n+1)} = 0$$

weil p_n ein Polynom n -ten Grades ist und die $(n + 1)$ te Ableitung gleich null ist. Weiterhin gilt

$$(f(x) - p_n(x))^{(n)} = \int_{x_0}^x (f(x) - p_n(x))^{(n+1)} = (f^{(n)}(x) - f^{(n)}(x_0)) - (p_n^{(n)}(x) - p_n^{(n)}(x_0)).$$

Mit dem Mittelwertsatz der Differentialrechnung ergibt sich

$$(f(x) - p_n(x))^{(n)} = f^{(n+1)}(\xi)(x - x_0) - \underbrace{p_n^{(n+1)}(\xi)} = 0 \quad (7)$$

für ein $\xi \in [x, x_0]$. Die n malige Integration von (7) liefert den Ausdruck

$$f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n + 1)!}(x - x_0)^{n+1}.$$

Beispiel

Wir nehmen $f(x) = e^x$ und $x_0 = 0$. Dann ergibt sich mit (6)

$$p_n(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \dots + \frac{x^n}{n!}. \quad (8)$$

Und es gilt

$$e^x - p_n(x) = e^\xi \frac{x^{n+1}}{(n+1)!} \leq \max(1, e^\xi) \frac{x^{n+1}}{(n+1)!}.$$

Um e^x mit $x = M \cdot 2^E$ zu bestimmen, kann man wie folgt vorgehen. Es gilt $|M| \in [\frac{1}{2}, 1)$ und $E \in \mathbb{Z}$.

$$e^x = (e^M)^{2^E}. \quad (9)$$

Es genügt also, die Exponentialfunktion für Werte in $[-1, 1]$ auswerten zu können. Dies kann mit Hilfe der Taylorpolynome p_n geschehen. Dabei ist nach (7) der Fehler maximal $\frac{e}{(n+1)!}$. Für 8 Stellen Genauigkeit benötigt man schon p_{12} . Die Nutzung der Potenzgesetze spart aber Rechenzeit.

$$e^M = (e^{M/256})^{256} = (((((((((e^{M/256})^2)^2)^2)^2)^2)^2)^2)^2. \quad (10)$$

- Bestimme $\frac{M}{2^8}$.
- Bestimme $p_4(M)$ mit p_4 aus (8).
- Quadriere $p_4(M)$ $|E| + 8$ mal.
- Wenn E negativ ist, bestimme das Reziproke des letzten Ergebnisses.
- Passe das Ergebnis an die Rechnerarithmetik an.

Dabei sind ein 8-Stellen Shift sowie 4 Additionen und 4 Multiplikationen für die Auswertung von p_4 nötig. Hinzu kommen weitere $|E| + 8$ Multiplikationen und eine potentielle Division. Da selbst in einer 64bit Arithmetik $e^{2^{10}} = Inf$ ist, ist auch die Anzahl an Multiplikationen auf 22 beschränkt.

Für höhere Genauigkeiten kann das benutzte p_n noch weiter erhöht werden.

3.2 Newton-Verfahren

Das Finden von Nullstellen von Funktionen ist nicht immer einfach. Aus diesem Grund gibt es Nährungsverfahren, wie z.B. das Newton-Verfahren.

Das Newton-Verfahren funktioniert am besten, wenn als Startpunkt ein Punkt gewählt wird, der möglichst nah an der gesuchten Nullstelle liegt. Wir versuchen in diesem Verfahren, die Nullstelle x^* durch mehrere Iterationen anzunähern.

a	3
M	0,75
E	2
$M/256$	1,4142136
$p_4(M/256)$	1,00293398
$(p_4(M/256))^2 56$	2,11700002
$((p_4(M/256))^2 56)^4$	20,0855369
M	0,627673039
E	5
e^3	20,0855369

Tabelle 1: Schritte bei der Bestimmung von e^3

Voraussetzungen dafür, dass das Newton-Verfahren überhaupt funktioniert, sind die Folgenden:

- eine *stetig differenzierbare* Funktion, die mindestens eine Nullstelle besitzt
- die erste Ableitung sollte nicht 0 werden an der Nullstelle

Es gibt verschiedene Herleitungen für dieses Verfahren. Im Folgenden stellen wir eine von diesen vor. Diese basiert auf der zuvor vorgestellten Taylor-Entwicklung. Sei $f(x)$ eine stetig differenzierbare Funktion. Weiterhin sei $f(x^*) = 0$, al die Funktion f besitzt eine Nullstelle bei x^* . Dann sei $x_0 \approx x^*$. Jetzt betrachten wir das p_1 -Polynom der Taylorentwicklung, also das Polynom bis zum Glied mit der 1. Ableitung.

$$f(x) \approx f(x_0) + f'(x_0) \cdot x - x_0 = p_1(x)$$

Jetzt bestimmen wir die Nullstelle x_1 dieses Polynoms.

$$f(x_0) + f'(x_0) \cdot (x_1 - x_0) = 0$$

Nun stellt man die Gleichung einfach nur noch nach x_1 um.

$$\begin{aligned} f'(x_0) \cdot (x_1 - x_0) &= -f(x_0) \\ x_1 - x_0 &= -\frac{f(x_0)}{f'(x_0)} \\ x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)} \end{aligned}$$

Diese Vorgehensweise kann beliebig oft wiederholt werden um immer bessere Näherungen für x^* zu erhalten und es ergibt sich dann folgende Iterationsvorschrift:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (11)$$

3.3 Konvergenz des Newton-Verfahren

Wir betrachten die Iterationsfunktion des Newton-Verfahrens

$$\varphi(x) = x - \frac{f(x)}{f'(x)}.$$

Es gilt für die Nullstelle x^* dass $\varphi(x^*) = x^*$. und weiterhin

$$\begin{aligned} \varphi'(x) &= 1 - \frac{(f'(x))^2 - f(x) \cdot f''(x)}{(f'(x))^2} \\ &= 1 - \frac{(f'(x))^2}{(f'(x))^2} + \frac{f(x) \cdot f''(x)}{(f'(x))^2} \\ &= \frac{f(x) \cdot f''(x)}{(f'(x))^2}. \end{aligned} \quad (12)$$

Damit gilt

$$\begin{aligned} x^* - x_{n+1} &= x^* - \varphi(x_n) \\ &= \varphi(x^*) - \varphi(x_n) \\ &= \varphi(x^*) - (\varphi(x^*) + \varphi'(x^*)(x_n - x^*) + \frac{\varphi(\xi)}{2}(x_n - x^*)^2). \end{aligned} \quad (13)$$

Dabei ist $\varphi(x_n)$ bis zum Polynom p_1 der Taylorentwicklung um x^* entwickelt und das Fehlerglied addiert worden. Mit (12) folgt $\varphi'(x^*) = 0$ da $f(x^*) = 0$ und $f'(x^*) \neq 0$. Für (12) folgt damit

$$x_{n+1} - x^* = \frac{\varphi''(\xi)}{2}(x_n - x^*)^2$$

für ein $\xi \in [x^*, x_n]$. Da der Abstand zur Nullstelle im $(n + 1)$ ten Schritt vom Quadrat des Abstands des n ten Schrittes abhängt, spricht man auch von *quadratischer Konvergenz*. Wenn $(x_n - x^*)$ genügend klein ist, dann ist $(x_{n+1} - x^*)$ noch sehr viel kleiner. Anschaulich kann man sagen, dass sich die Anzahl korrekter Stellen der Iterierten x_n in jedem Schritt etwa verdoppelt.

3.4 Division mit Hilfe des Newton-Verfahrens

Die Division mit Hilfe des Newton-Verfahrens beruht auf dem Prinzip, dass wir uns eine Funktionsschar suchen die als Nullstellen die Reziprokenwerte der Zahlen a hat, durch die wir dividieren möchten. Eine spezielle Wahl dieser Funktionen sorgt dafür, dass das Newton-Verfahren ohne Divisionen anwendbar ist.

Zur Bestimmung des Reziproken einer Zahl a kann man wie folgt vorgehen. Zur Vereinfachung gehen wir von $a > 0$ aus. Das Vorzeichen von a und a^{-1} ist identisch. Wir machen die folgenden Vorbetrachtungen

- $a = M \cdot 2^E$,
- $\frac{1}{a} = \frac{1}{M} \cdot 2^{-E}$,
- $f(x) = \frac{1}{x} - M$ hat $\frac{1}{M}$ als Nullstelle,
- $x_{n+1} = x_n - \frac{\frac{1}{x_n} - M}{-\frac{1}{x_n^2}} = x_n(2 - M \cdot x_n)$.

Man muss also nur das Reziproke von Zahlen $M \in [0, 5; 1)$ bestimmen können.

$$\begin{aligned} a &= M \cdot 2^E \\ x_{n+1} &= x_n - \frac{\frac{1}{x_n} - M}{-\frac{1}{x_n^2}} = x_n(2 - M \cdot x_n) \end{aligned}$$

Diese Gleichung dient uns zur Reziprokenberechnung. Da das Verfahren schneller konvergiert, wenn man Startwerte wählt, die in der Nähe der gesuchten Nullstelle liegen, benötigen wir ein Verfahren, das uns geeignete Startwerte (x_0) liefert, also solche, die bereits nahe an $\frac{1}{M}$ liegen. Eine einfache Möglichkeit ist eine Approximation von $1/M$ durch eine lineare Funktion der Form $g(x) = mx + n$.

Die Funktionsgleichung kann man durch folgendes Gleichungssystem gewinnen:

$$\begin{aligned} f(x) &= \frac{1}{x} \\ f\left(\frac{1}{2}\right) - g\left(\frac{1}{2}\right) &= \delta \\ f(1) - g(1) &= \delta \\ f(\xi) - g(\xi) &= -\delta \\ (f(\xi) - g(\xi))' &= 0 \end{aligned}$$

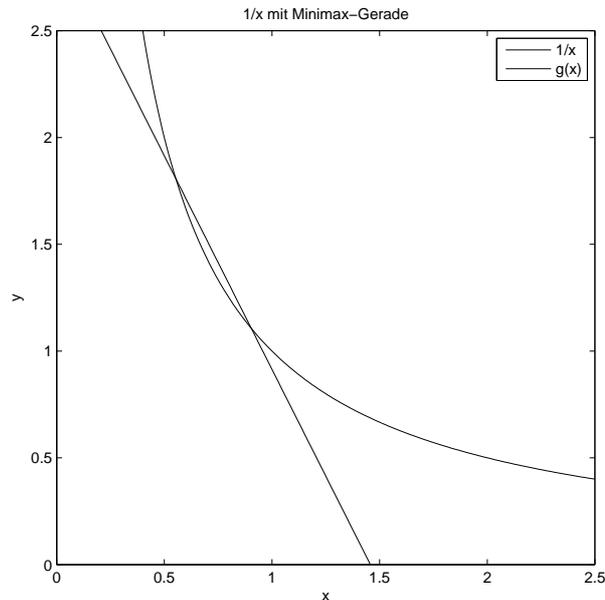


Abbildung 1: Minimax-Gerade

Dabei fordern wir, dass der maximale Abstand zwischen f und g an den Rändern und an einer Zwischenstelle angenommen wird und insgesamt minimal ist. Die Lösung dieses Gleichungssystems ist die sogenannte *Minimax-Gerade*

$$g(x) = -2 \cdot x + \frac{3}{2} + \sqrt{2}.$$

Mit diesem Startwert reichen für 8 Stellen Genauigkeit etwa 3 Newton-Schritte.

- Bestimme $x_0 = g(M) = -2 \cdot M + (\frac{3}{2} + \sqrt{2})$.
- Bestimme $x_3 \approx \frac{1}{M}$.
- Rechenbedarf: 7 Multiplikationen, 4 Additionen

Da $1/M \in (1; 2]$ liegt müssen wir nun noch die Zahl an den Mantissenbereich und den Exponenten anpassen.

Dieses Verfahren zur Division (nach Multiplikation mit dem entsprechenden Zähler) ist tatsächlich schneller als die Vorgehensweise wie beim schriftlichen Dividieren.

$$\frac{1}{a} \approx \frac{x_3}{2} \cdot 2^{-E+1}$$

a	3
M	0,75
E	2
x_0	1,4142136
x_1	1,3284271
x_2	1,3333153
x_3	1,3333333
M	0,6666667
E	-1
$1/a$	0,3333333

Tabelle 2: Schritte bei der Bestimmung von $1/3$

Anwendungen

Weitere Anwendungsmöglichkeiten sind zum Beispiel das Logarithmieren. Es gilt

- $a = M \cdot 2^E$,
- $\ln(a) = \ln(M) + \ln(2) \cdot E$,
- $e^x - M$ hat $\ln(M)$ als Nullstelle.
- $x_{n+1} = x_n - \frac{e^{x_n} - M}{e^{x_n}} = x_n - 1 + M \cdot e^{-x_n}$ (Newton)

Es muss also nur $\ln(M)$ für $M \in [\frac{1}{2}, 1)$ durch einige Newtoniterationen ausgewertet werden. Dafür sind nur Additionen und Multiplikationen notwendig. Zusätzlich muss $\ln(2)$ als Konstante zur Verfügung stehen. Die Minimax-Gerade für den Startwert ist $g(x) = 1,38629436x - 1,35646431$.

4 Der CORDIC-Algorithmus

Der CORDIC⁴ Algorithmus ist ein effektives Verfahren, um den Cosinus und den Sinus und damit den Tangens eines Winkels nährungsweise zu bestimmen. Der Algorithmus basiert auf folgender Iteration:

⁴Die Abkürzung steht für *Coordinate Rotation Digital Computer* und wurde von Jack E. Volder Ende der 50er Jahre entwickelt.

a	3
M	0,75
E	2
x_0	-0,316743539
x_1	-0,287255667
x_2	-0,287681982
x_3	-0,287682073
$\ln(2) \cdot 2$	1,38629436
$x_3 + \ln(2) \cdot 2$	1,09861229
M	0,549306144
E	1
$\ln(3)$	1,09861229

Tabelle 3: Schritte bei der Bestimmung von $\ln 3$

$$\begin{cases} x_{n+1} &= x_n - d_n \cdot y_n \cdot 2^{-n} \\ y_{n+1} &= y_n + d_n \cdot x_n \cdot 2^{-n} \\ z_{n+1} &= z_n - d_n \cdot \arctan 2^{-n} \\ d_n &= \text{sign}(z_n) \end{cases} \quad (14)$$

Alle möglichen Werte von $\arctan 2^{-n}$ für die verschiedenen n werden vorgespeichert, wobei üblicherweise n etwas größer als die Mantissenlänge gewählt wird. Wenn $|z_0|$ kleiner oder gleich

$$\sum_{k=0}^{\infty} \arctan 2^{-k} = 1.743286047... \quad (15)$$

ist, dann ergibt sich

$$\lim_{n \rightarrow \infty} \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix} = K \times \begin{pmatrix} x_0 \cdot \cos z_0 - y_0 \cdot \sin z_0 \\ x_0 \cdot \sin z_0 + y_0 \cdot \cos z_0 \\ 0 \end{pmatrix},$$

wobei der Skalierungsfaktor K gleich $\prod_{n=0}^{\infty} \sqrt{1 + 2^{-2n}} = 1.64676025...$ ist. Um nun den Cosinus und Sinus eines Winkels zu bestimmen, benötigt man noch die Startwerte für x , y und z . Die Zahl für die Cosinus bzw. Sinus bestimmt werden

sollen, ist θ . Dabei kann θ maximal so groß wie der bei (15) berechnete Wert sein.

$$\begin{aligned}x_0 &= \frac{1}{K} = 0,60725\dots \\y_0 &= 0 \\z_0 &= \theta\end{aligned}$$

Das Grundprinzip ist, dass man versucht durch Drehung eines Einheitsvektors um verschiedene immer kleiner werdende Winkel $\pm \arctan(2^{-n})$ einzukreisen. Ersetzt man die Drehung um $\pm \arctan(2^{-n})$ durch eine Drehstreckung mit dem zusätzlichen Faktor $\sqrt{1 + 2^{-2n}}$, dann sind diese Drehstreckungen durch den Algorithmus (14) realisiert. Zum Ausgleich der Streckung wird mit dem auf $1/K$ verkürzten Vektor begonnen. Der finale Vektor ist der um θ gedrehte Einheitsvektor, von dessen Komponenten $\cos \theta$ und $\sin \theta$ abgelesen werden können.

Die Vorteile bei diesem Verfahren sind zum einen, dass es sowohl Cosinus als auch Sinus gleichzeitig berechnet und dass man zum Berechnen nur addieren und Mantissen shiften können muss.

5 Oder doch lieber Tabellen?

5.1 Grundlagen

Neben der Möglichkeit die Werte einer Funktion auszurechnen, ist es auch möglich, Tabellen zur Wertebestimmung heranzuziehen. Dabei werden bestimmte Stellen – die sogenannten Stützstellen – und die dazugehörigen Werte gespeichert. Dieses Verfahren wurde bevor es Taschenrechner gab zur Bestimmung von beispielsweise Sinuswerten herangezogen. Damals ergab sich meist eine Genauigkeit von ca. 4 Stellen, weil die tabellarisierten Werte nicht genauer angegeben waren. Um die Werte zwischen zwei Stützstellen zu bestimmen, wurde einfach eine Gerade durch diese beiden Punkte gelegt und deren Werte als Näherungen angenommen. Dies ist jedoch bei einer geforderten Maschinengenauigkeit von mindestens 16 Stellen zu ungenau. Und besser als eine solche Gerade wäre ein Polynom $p(x)$ $(n - 1)$ -ten Grades, wobei n die Anzahl der Stützstellen ist.

$$p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$$

Dieses Polynom muss an allen Stützstellen die in der Tabelle abgespeicherten Werte annehmen, $p(x_i) = y_i$. Aber noch ist unklar ob dieses Polynom überhaupt existiert.

Um die Existenz dieses Polynoms zu beweisen, definieren wir zuerst ein sogenanntes *Lagrange-Polynom* $(n - 1)$ -Grades, für das gilt:

$$L_i(x_j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad (16)$$

Durch Multiplikation der Linearfaktoren ergibt sich:

$$\begin{aligned} l_i &= (x - x_1)(x - x_2) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n) \\ l_i(x_i) &= (x_i - x_1)(x_i - x_2) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n) \end{aligned}$$

Dadurch ist in l_i die Bedingung $l_i(x_j) = 0$ für $i \neq j$ erfüllt. Durch Teilung von l_i durch $l_i(x_i)$ ist auch die zweite Bedingung $l_i(x)/l_i(x_i) = 1$ für $i = j$ erfüllt und $L_i(x)$ ist nun definiert mit

$$L_i(x) = \frac{l_i}{l_i(x_i)}.$$

Daraus ergibt sich für p

$$p = a_1 L_1 + a_2 L_2 + \dots + a_n L_n.$$

Und für die Stützstellen x_i

$$p(x_i) = a_1 L_1(x_i) + a_2 L_2(x_i) + \dots + a_n L_n(x_i).$$

Dabei ist jedoch wegen Bedingung (16) jeder Summand außer $a_i L_i(x_i)$ gleich Null, denn L_i und a_i beziehen sich nur bei der Stützstelle x_i auf das gleiche i und damit ist der Faktor L_i in jedem anderem Summanden Null. Folglich ist $p(x_i) = a_i = y_i$. Durch Koeffizientenvergleich an allen Stützstellen ergibt sich

$$p(x) = y_1 L_1(x) + y_2 L_2(x) + \dots + y_n L_n(x).$$

Dieses Polynom existiert und erfüllt alle vorher an $p(x)$ gestellten Anforderungen. Auch wenn die Existenz dieses Polynoms nun bewiesen ist, ist noch nicht klar, dass dieses eindeutig ist, dies wie folgt zeigen.

Nehmen wir einmal an, es gibt neben dem Polynom $p(x)$ das Polynom $q(x)$,

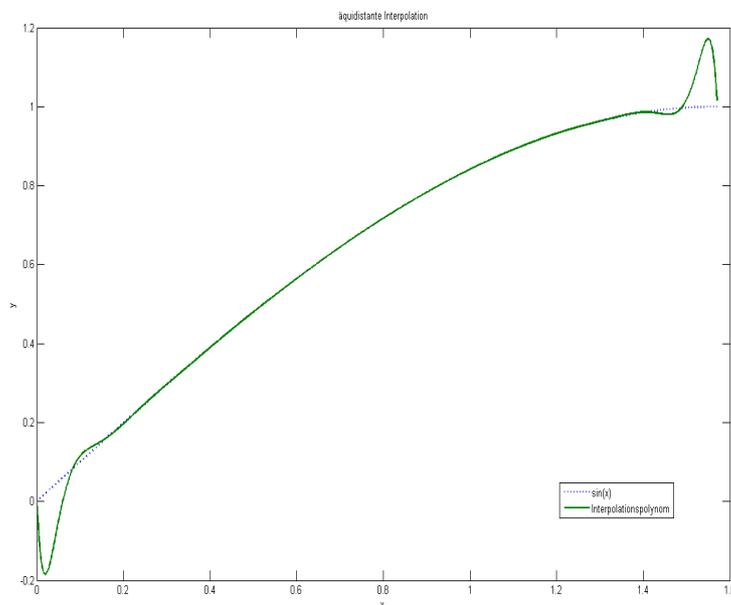
welches auch an allen Stützstellen x_i durch die zugehörigen Funktionswerte y_i läuft. $q(x)$ ist wie $p(x)$ ein Polynom $(n - 1)$ -ten Grades.

Wären $p(x)$ und $q(x)$ unterschiedlich, dann ließe sich die Differenz r durch $r = p - q$ beschreiben. Da dies an allen Stellen von p und q gilt, folgt daraus auch

$$r(x_i) = p(x_i) - q(x_i). \quad (17)$$

$r(x)$ ist ein Polynom $(n - 1)$ -ten Grades, da es die Differenz von zwei Polynomen $(n - 1)$ -ten Grades ist. Aus (17) geht hervor, dass $r(x_i)$ immer gleich Null ist, da p und q an den Stellen x_i gleich y_i sein müssen. x_i steht für die Stützstellen, deren Anzahl n ist (siehe oben). Damit hat r n Nullstellen. Es gibt nur ein Polynom, dessen Grad kleiner ist als die Anzahl seiner Nullstellen und das ist die Nullfunktion. Deshalb gilt $r(x) = 0$ woraus folgt $p(x) = q(x)$ und bewiesen wäre, dass nur ein Polynom $p(x)$ vom Grad $(n - 1)$ existiert, welches durch alle Stützstellen verläuft.

Durch Interpolation mit dem Polynom lassen sich die Werte zwischen den tabellarisierten Stützstellen approximieren. Um eine sehr hohe Genauigkeit zu erhalten, müsste nun eigentlich nur die Anzahl an Stützstellen entsprechend hoch gewählt werden. Wenn alle Stützpunkte gleich verteilt werden ergibt sich jedoch eine sehr große Abweichung an den Rändern des in der Tabelle angegebenen Intervalls. Dieses Bild verdeutlicht die Abweichung bei einer Annäherung an die Sinuskurve (gestrichelt) mit einem Polynom bei 20 gleich verteilten Stützstellen:



5.2 Tschebyschev-Interpolation

Um diese Abweichung zu beheben ist es sinnvoll die Verteilung der Stützstellen zu verändern, indem man mehr Stützstellen an den Rändern wählt. Um diese Verteilung zu optimieren wurde das sogenannte *Tschebyschev-Verfahren* entwickelt.

Bei diesem Verfahren wird nach einem Polynom n -ten Grades gesucht, das in dem Intervall $[-1; 1]$ den maximalen Betrag 1 hat sowie alle Maxima und Minima bei ± 1 . Dabei steht n für die Anzahl der späteren Stützstellen. Das Polynom, welches diese Bedingungen erfüllt heißt *Tschebyschev-Polynom* T_n :

$$\begin{aligned}n = 0 : T_0 &= 1 \\n = 1 : T_1 &= x \\n = 2 : T_2 &= 2x^2 - 1 \\&\vdots \\n = k : T_k &= \cos(k \cdot \arccos(x))\end{aligned}$$

Um die Nullstellen zu bestimmen wird T_n gleich Null gesetzt, da der Cosinus seine Nullstellen bei $\frac{\pi}{2} + (\pi)i$ für alle $i \in \mathbb{N}$ hat, ergibt sich:

$$\begin{aligned}k(\arccos(x)) &= \frac{\pi}{2} + (\pi)i \\ \arccos(x) &= \frac{\pi}{k} \left(\frac{1}{2} + i \right) \\ x &= \cos\left(\frac{\pi}{k} \left(\frac{1}{2} + i \right)\right)\end{aligned}$$

Der Cosinus nimmt auf dem Intervall $[0; \pi]$ die Werte aus dem Intervall $[-1; 1]$ an. Daraus ergibt sich die Ungleichung $0 \leq \frac{\pi}{k} \left(\frac{1}{2} + i \right) \leq \pi$, das heißt i nimmt alle Werte zwischen $i = 0$ und $i = k - 1$ an. Dies bedeutet auch, dass es insgesamt k Nullstellen gibt welche als Stützstellen dienen. Die Polynominterpolation einer beliebigen Funktion an diesen Stützstellen weist eine bedeutend höhere Genauigkeit auf als mit gleichverteilten Stützstellen. Eine Vergleichsrechnung zu den gleichverteilten Stützstellen zeigte nur noch Abweichungen in der Größenordnung der Maschinengenauigkeit und die Graphen der Funktion und der Interpolierenden waren deckungsgleich.

5.3 Das Horner-Schema

Wie bereits gesehen, basieren viele Näherungsverfahren auf Polynomauswertungen (Taylorpolynome, Interpolationspolynome). Dabei kommt der Effizienz und Rechengeschwindigkeit eine immer größere Bedeutung zu. Dies hat zur Folge, dass versucht wird, die Algorithmen zu optimieren.

Das Horner-Schema ist eine spezielle Anordnung der Faktoren eines Polynoms

$$p(x) = a_0 + xa_1 + x^2a_2 + x^3a_3 \dots + x^na_n$$

Bei Betrachtung ergibt sich, dass hier n Additionen benötigt werden. Hinzu kommen noch zur Auswertung der x^k -Potenzen jeweils $(k-1)$ Multiplikationen. Durch den steigenden Exponenten addiert mit den Multiplikationen der x -Potenzen und der dazugehörigen Koeffizienten a_k ergeben sich insgesamt $\frac{n(n+1)}{2}$ Multiplikationen.

Wird das x ausgeklammert ergibt sich folgendes Polynom:

$$p(x) = a_0 + x(a_1 + xa_2 + x^2a_3 \dots + x^{n-1}a_n)$$

Nun wird x im entstandenen Faktor wieder ausgeklammert so dass sich folgendes Polynom ergibt:

$$p(x) = a_0 + x(a_1 + x(a_2 + xa_3 \dots + x^{n-2}a_n))$$

Dieses Prinzip lässt sich n -mal wiederholen, sodass sich folgendes Polynom ergibt:

$$q(x) = a_0 + x(a_1 + x(a_2 + x(a_3 \dots + xa_n))) \dots$$

Dieses Polynom $q(x)$, das durch äquivalente Umformung aus dem ursprünglichen Polynom $p(x)$ hervorgegangen ist wird jetzt noch auf Effizienz untersucht. Genaues Zählen liefert, dass n Additionen und nur noch n Multiplikationen nötig sind.

6 Fazit

Wozu nun selber rechnen? Mit unserem Projekt sollte verdeutlicht werden, wie ein Computer rechnet und vor allem sollte gezeigt werden, wo die Grenzen unserer Rechner liegen. Wenn man diese Grenzen kennt, weiß man auch, ab wann man selbst rechnen muss. Auch wenn die Maschinengenauigkeit bei einem 64bit-Zahlensystem sehr klein erscheint, kann in längeren Rechnungen mit vielen Iterationen so oft aufsummiert werden, dass das Ergebnis entweder verfälscht oder gänzlich ungenau wird. Wenn man jedoch weiß, dass das Ergebnis eines Rechners

nicht immer exakt sein kann, lohnt es sich längere Rechnungen schon anfangs so einfach wie möglich zu halten, damit die vermutete Ungenauigkeit so gering wie möglich wird und besser berücksichtigt werden kann.

Komplexe Zahlen und Geometrie

Teilnehmer:

Niklas Rughöft	Herder-Oberschule
Jan Putzig	Heinrich-Hertz-Oberschule
Ron Wenzel	Heinrich-Hertz-Oberschule
Alexey Loutchko	Heinrich-Hertz-Oberschule
Joe Hannes Gerstung	Heinrich-Hertz-Oberschule
Jörn Brodthagen	Andreas-Oberschule

Gruppenleiter:

Heino Hellwig	Humboldt-Universität zu Berlin, Mitglied im DFG-Forschungszentrum MATHEON „Mathematik für Schlüsseltechnologien“
---------------	--

Unsere Arbeitsgruppe befasste sich mit den komplexen Zahlen und ihrer geometrischen Darstellung. Die komplexen Zahlen wurden dann zur Lösung einfacher geometrischer Probleme herangezogen. Die Inversion am Kreis und die stereographische Projektion wurden eingeführt und einige ihrer Eigenschaften bewiesen. Mit diesem Wissen konnte dann die Möbiustransformationen geometrisch gedeutet werden, welche im Kurzfilm *Möbius Transformations Revealed* auf eindrucksvolle Weise visualisiert wurden.

1 Der Körper der komplexen Zahlen

In diesem Abschnitt werden kurz die grundlegenden Eigenschaften der komplexen Zahlen wiederholt.

Definition: Die komplexen Zahlen $\mathbb{C} = \{\mathbb{R} \times \mathbb{R}, \cdot, +\}$ bestehen aus den geordneten, reellen Zahlenpaaren, versehen mit einer wie folgt definierten Addition

$$(x_1, y_1) + (x_2, y_2) := (x_1 + x_2, y_1 + y_2) \quad (1)$$

und Multiplikation

$$(x_1, y_1) \cdot (x_2, y_2) := (x_1x_2 - y_1y_2, x_1y_2 + x_2y_1). \quad (2)$$

Damit die Menge der komplexen Zahlen einen Körper bildet, müssen bestimmte Bedingungen erfüllt sein. Es müssen die Assoziativgesetze, die Kommutativgesetze und die Distributivgesetze gelten. Außerdem muss die Menge der komplexen Zahlen ein Nullelement (also ein neutrales Element für die Addition), ein Einselement (also ein neutrales Element für die Multiplikation) und ein inverses Element enthalten:

- **Assoziativgesetze:**

$$\begin{aligned} ((x_1, y_1) \cdot (x_2, y_2)) \cdot (x_3, y_3) &= (x_1, y_1) \cdot ((x_2, y_2) \cdot (x_3, y_3)) \\ ((x_1, y_1) + (x_2, y_2)) + (x_3, y_3) &= (x_1, y_1) + ((x_2, y_2) + (x_3, y_3)) \end{aligned}$$

- **Kommutativgesetze:**

$$\begin{aligned} (x_1, y_1) \cdot (x_2, y_2) &= (x_2, y_2) \cdot (x_1, y_1) \\ (x_1, y_1) + (x_2, y_2) &= (x_2, y_2) + (x_1, y_1) \end{aligned}$$

- **Distributivgesetze:**

$$\begin{aligned} (x_1, y_1) \cdot ((x_2, y_2) + (x_3, y_3)) &= (x_1, y_1) \cdot (x_2, y_2) + (x_1, y_1) \cdot (x_3, y_3) \\ ((x_1, y_1) + (x_2, y_2)) \cdot (x_3, y_3) &= (x_1, y_1) \cdot (x_3, y_3) + (x_2, y_2) \cdot (x_3, y_3) \end{aligned}$$

- **Existenz der neutralen Elemente:**

a) bzgl. Multiplikation:

$$(x, y) \cdot (1, 0) = (x, y)$$

b) bzgl. Addition:

$$(x, y) + (0, 0) = (x, y)$$

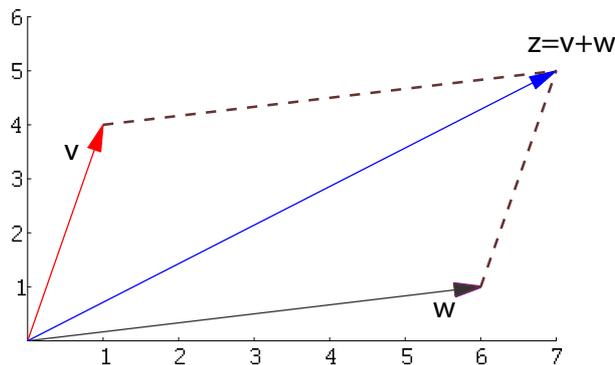


Abbildung 1: Die Addition komplexer Zahlen

- **Existenz der inversen Elemente:**

a) bzgl. Multiplikation:

$$(x_1, y_1) \cdot \left(\frac{x_1}{x_1^2 + y_1^2}, \frac{-y_1}{x_1^2 + y_1^2} \right) = (1, 0)$$

b) bzgl. Addition:

$$(x, y) + (-x, -y) = (0, 0)$$

Da alle Eigenschaften erfüllt sind, bildet die Menge der komplexen Zahlen tatsächlich einen Körper.

Dieser neue Zahlenbereich wurde geschaffen, da Gleichungen der Form

$$z^2 = -1 \tag{3}$$

in der Menge der reellen Zahlen keine Lösungen haben. Die so genannte **imaginäre Einheit i** (Euler, 1777) wird eingeführt als $i := (0, 1)$. Es gilt dann:

$$i^2 = i \cdot i = (0, 1) \cdot (0, 1) = (-1, 0) = -1.$$

Allgemein lässt sich eine komplexe Zahl z als geordnetes Paar zweier reeller Zahlen (x, y) darstellen, wobei der x -Wert den *Realteil* und der y -Wert den *Imaginärteil* der Zahl z bestimmt:

$$z = x + yi = \operatorname{Re}(z) + \operatorname{Im}(z)i.$$

2 Die Geometrie der komplexen Zahlen

Jede komplexe Zahl $z = x + yi$ lässt sich als Punkt $P = (\operatorname{Re}(z), \operatorname{Im}(z)) = (x, y)$ der Ebene geometrisch deuten. Die Addition und Multiplikation erlauben so geometrische Interpretationen. Um die Addition zweier komplexer Zahlen v und w

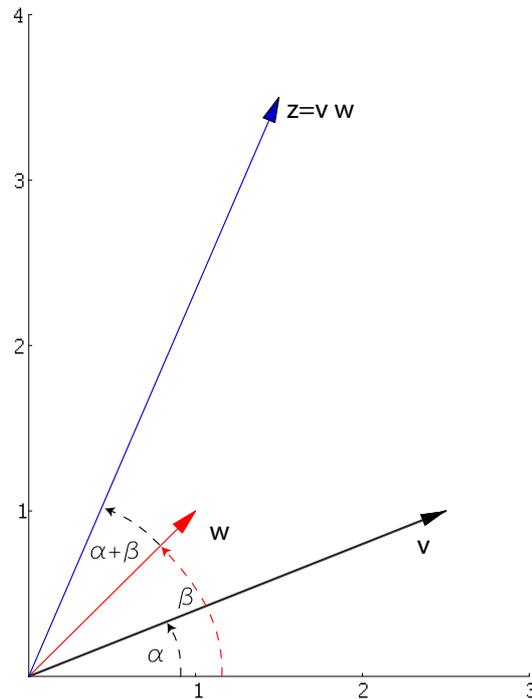


Abbildung 2: Die Multiplikation komplexer Zahlen

zu veranschaulichen, fasst man sie als Vektoren auf und verfährt nach der Parallelogrammregel, indem man v an w abträgt. Der daraus resultierende Vektor z ist die Summe der beiden komplexen Zahlen.

Die Multiplikation zweier komplexer Zahlen ist geometrisch gesehen eine Drehstreckung. Um zwei komplexe Zahlen v und w in Polarschreibweise zu multiplizieren, addiert man die Winkel α und β der beiden Zahlen und multipliziert die Beträge $|v|$ und $|w|$ der beiden Zahlen.

Definitionen: Für jede komplexe Zahl $z \in \mathbb{C}$ heißt $\bar{z} := x - iy$ die **konjugiert komplexe Zahl**. Der **Betrag** einer komplexen Zahl wird definiert durch

$$|z| := \sqrt{(z\bar{z})} = \sqrt{x^2 + y^2}.$$

Es gilt:

$$\operatorname{Re}(z) = \frac{1}{2}(z + \bar{z})$$

$$\operatorname{Im}(z) = 1/2i(z - \bar{z})$$

Eigenschaften der Konjugierten:

- i) $\overline{w + z} = \bar{w} + \bar{z}$
- ii) $\overline{w \cdot z} = \bar{w} \cdot \bar{z}$
- iii) $\overline{\bar{z}} = z$

Kreise in der Gauß'schen Zahlenebene:

In der Gauß'schen Zahlenebene gilt für den Abstand d zweier Punkte z_1 und z_2 :

$$d = |z_1 - z_2|.$$

Der Kreis mit dem Mittelpunkt $M(a, b)$ und dem Radius r wird mit $K_r(M)$ bezeichnet. Er besteht aus den Punkten, die von M den Abstand r haben. Mit $M = a + bi$ gilt.

$$K_r(M) := \{z : |M - z| = r\}$$

Da beim Rechnen die Betragsstriche stören, wird umgeformt:

$$|M - z|^2 = (M - z)\overline{(M - z)} = (M - z)(\overline{M} - \overline{z})$$

Daraus folgt:

$$\begin{aligned} (M - z)(\overline{M} - \overline{z}) &= r^2 \\ z\overline{z} - \overline{M}z - M\overline{z} + M\overline{M} - r^2 &= 0 \end{aligned}$$

Somit kann man Kreise in der Gauß'schen Zahlenebene wie folgt darstellen:

$$K_r(M) := \{z \in \mathbb{C} \mid z\overline{z} - \overline{M}z - M\overline{z} + M\overline{M} - r^2 = 0\}, \quad (4)$$

wobei $M\overline{M} - r^2$ stets eine reelle Zahl.

Komplexe Zahlen können sehr hilfreich bei geometrischen Problemen sein, die mit konventionellen Mitteln erheblich schwieriger zu lösen wären. Daher sollen die komplexen Zahlen nun genutzt werden, um einige Aufgaben der Elementargeometrie zu lösen.

Aufgabe:

Auf einer Schatzinsel stehen eine Kiefer, eine Linde und ein Galgen. Ein Pirat hat vor langer Zeit einen Schatz dort versteckt. Er ist vom Galgen zur Kiefer gegangen und ist um 270° gedreht die selbe Strecke nochmal gelaufen und hat diesen Punkt markiert. Dannach ist er vom Galgen zur Linde gegangen und ist um 90° gedreht diese Strecke nochmal gelaufen und hat diesen Punkt auch markiert. In der Mitte zwischen diesen beiden Punkten hat er den Schatz vergraben (siehe 2). Als er nach einigen Jahren wieder zurück zur Insel gekommen ist, war der Galgen weg. Wo ist der Schatz vergraben?

Lösung:

Wir legen uns über die Insel ein Koordinatensystem, sodass die Bäume an den Positionen $(-1, 0)$ und $(1, 0)$ sind. Der Galgen befindet sich an der Position $(a+bi)$. Nun verschieben wir den Galgen um 1 nach links und drehen ihn um 90° gegen den Uhrzeigersinn (durch Multiplikation mit $-i$):

$$-i((a-1) + bi) = -ia + i + b = b + i(-a+1).$$

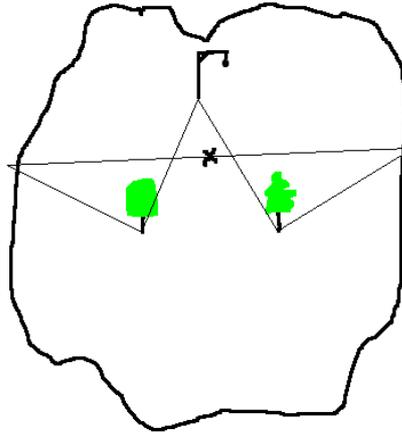


Abbildung 3: Position des Schatzes

Danach verschieben wir den Punkt wieder um 1 nach rechts: $w_1 = b + 1 + i(-a + 1)$ ist also der erste markierte Punkt. Analog machen wir dies nun für w_2 und erhalten $w_2 = -b - 1 + i(a + 1)$. Nun berechnen wir die Position des Schatzes:

$$s = \frac{w_1 + w_2}{2} = \frac{b + 1 + i(-a + 1) - b - 1 + i(a + 1)}{2} = \frac{ia - ia + 2i}{2} = i$$

Der Schatz ist also, *unabhängig* von der Position des Galgen, an der Stelle i begraben!

Satz:

Konstruiert man auf den Seiten eines beliebigen Vierecks Quadrate, so sind die Strecken, die die Mittelpunkte gegenüberliegender Quadrate verbinden, gleich lang und stehen senkrecht aufeinander.

Beweis:

Für den Beweis ziehen wir nun die komplexen Zahlen zur Hilfe. Wir wählen den Ursprung von \mathbb{C} als den Punkt A und beschreiben die weiteren Eckpunkte durch Addition von den komplexen Zahlen $a, b, c, d \in \mathbb{C}$. Die Eckpunkte haben die Darstellung:

$$B = 2a, \quad C = 2a + 2b, \quad D = 2a + 2b + 2c, \quad A = 2a + 2b + 2c + 2d$$

Unsere einzige Bedingung, damit das Viereck geschlossen ist, ist: $a + b + c + d = 0$. Um den Mittelpunkt p des Quadrates über der Strecke $\overline{AB} = 2a$ zu erhalten, nehmen wir nun erst die Hälfte der Seite und addieren die selbe Strecke im rechten Winkel zu \overline{AB} . Um eine komplexe Zahl um 90° gegen den Uhrzeigersinn zu drehen multiplizieren wir sie mit i :

$$p = a + ai.$$

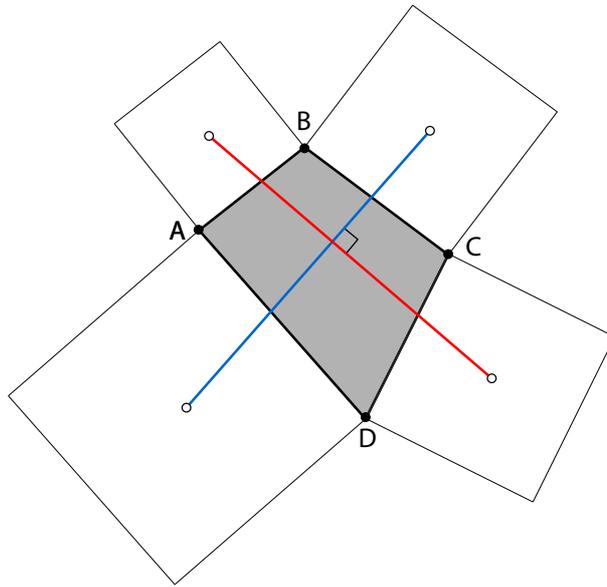


Abbildung 4: Quadrate über einem Sehnenviereck

Ebenso verfahren wir mit den anderen Quadratmittelpunkten und erhalten:

$$q = 2a + b + bi, \quad r = 2a + 2b + c + ci, \quad s = 2a + 2b + 2c + d + di.$$

Die den komplexen Zahlen $e, f \in \mathbb{C}$ mit $e = r - p$ und $f = s - q$ zugeordneten Ortsvektoren repräsentieren die Strecken zwischen den gegenüberliegenden Quadratmittelpunkten. Sie berechnen sich zu: $e = b + 2c + d + id - ib$ und $f = a + 2b + c + ic - ia$. Um nun zu zeigen, dass e und f den gleichen Betrag haben sowie orthogonal zueinander sind, müssen wir nur zeigen, dass $e + if = 0$ gilt. Wieder verwenden wir also die Multiplikation mit i , um eine Drehung um 90° zu erreichen:

$$e + if = b + 2c + d + id - ib + ia + 2ib + ic - c + a = a + b + c + d + i(a + b + c + d).$$

Nach unserer Voraussetzung gilt $a + b + c + d = 0$, also stimmt die Gleichung. q.e.d.

Weitere Anwendungen finden die komplexen Zahlen in der Zahlentheorie. Wir betrachten dazu das **Gaußsche Zahlengitter** Γ , welches aus allen Zahlen $z = a + bi$ mit $a, b \in \mathbb{Z}$ besteht. mit dessen Hilfe lässt sich leicht der folgende Satz beweisen lässt.

Satz (Zwei-Quadrate-Satz):

Können zwei ganze Zahlen M, N ausgedrückt werden als die Summe zweier Quadratzahlen, so gilt das auch für das Produkt.

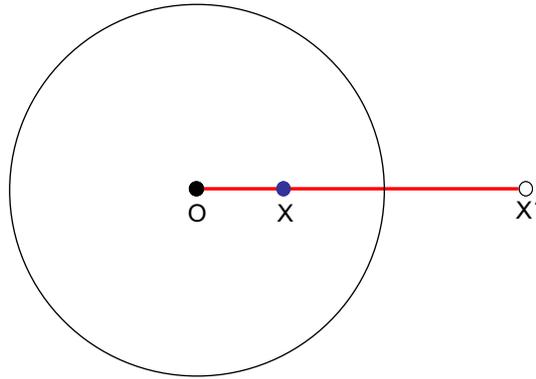


Abbildung 5: Die Inversion am Kreis

Beweis:

Sei $M = a^2 + b^2$, $N = c^2 + d^2$ und $z = a + ib$, $w = c + id$ mit $a, b, c, d \in \mathbb{Z}$. Beachte, dass gilt: $|a + ib|^2 = (a + ib)(a - ib) = a^2 + b^2$ und $|c + id|^2 = c^2 + d^2$. Dann ist:

$$M \cdot N = |a + ib|^2 \cdot |c + id|^2 = z\bar{z}w\bar{w} = zw\bar{z}\bar{w}$$

und da $zw \in \Gamma$ gilt: $zw := u = k + il$ mit $k, l \in \mathbb{Z}$ und daher auch:

$$k^2 + l^2 = |u|^2 = u \cdot \bar{u} = M \cdot N.$$

3 Inversion am Kreis

Lineare Abbildungen der komplexen Ebene auf sich selber haben die Form:

$$f(z) = az + b,$$

wobei $a, b \in \mathbb{C}$ und $|a| = 1$ und beschreiben aus Translation und Rotation zusammengesetzte orientierungserhaltende Bewegungen. Lineare Abbildungen der Form

$$f(z) = a\bar{z} + b,$$

wobei $a, b \in \mathbb{C}$ und $|a| = 1$ beschreiben zusätzlich noch die Spiegelung und sind nicht orientierungserhaltend. Eine weitere wichtige komplexe Abbildung ist die Inversion am Kreis.

Definition: Die **Inversion oder Spiegelung am Kreis** mit Mittelpunkt O und Radius r ist die Abbildung, welche jeden Punkt X einen Punkt X' zuordnet, derart, dass X' auf der Halbgerade OX liegt und die Abstandsgleichung:

$$|OX| \cdot |OX'| = r^2$$

erfüllt.

Als komplexe Abbildung wird die Inversion am Einheitskreis durch

$$f(z) = \frac{1}{\bar{z}} \quad (5)$$

gegeben.

Eine wichtige Eigenschaft der Inversion ist:

Satz:

Die Inversion bildet Kreise auf Kreise und Geraden ab.

Beweis:

Gegeben sei ein Kreis in allgemeiner Form:

$$f(x, y) = A(x^2 + y^2) + Bx + Cy + D = 0$$

Inversion am Einheitskreis liefert für das Bild des Punktes $P(X, Y)$ den Punkt $P(x, y)$ mit $x = \frac{X}{X^2 + Y^2}$, $y = \frac{Y}{X^2 + Y^2}$. Daher ist

$$f\left(\frac{X}{X^2 + Y^2}, \frac{Y}{X^2 + Y^2}\right) = A \frac{1}{X^2 + Y^2} + B \frac{X}{X^2 + Y^2} + C \frac{Y}{X^2 + Y^2} + D = 0$$

D.h.: $A + Bx + Cy + D(X^2 + Y^2) = 0$.

Es sind zwei Fälle zu unterscheiden:

i) $D = 0 \rightarrow A + Bx + Cy = 0$, also eine Gerade

ii) $A = 0 \rightarrow Bx + Cy + D(x^2 + y^2) = 0$, das ist ein Kreis durch den Ursprung.

□

Folgerung:

Insbesondere werden Kreise durch den Ursprung auf Geraden abgebildet.

Hilfssatz:

Wir betrachten die Punkte s, t und deren Bildpunkte S, T nach Inversion am Kreis mit Radius r . Die Streckenlänge $|st|$ wird wie folgt transformiert:

$$|ST| = \frac{r^2 \cdot |st|}{|Os| \cdot |Ot|}$$

Beweis:

Es gilt:

$$\triangle Ost \sim \triangle OTS.$$

Daher ist

$$\frac{|st|}{|ST|} = \frac{|Os|}{|OT|} \quad (6)$$

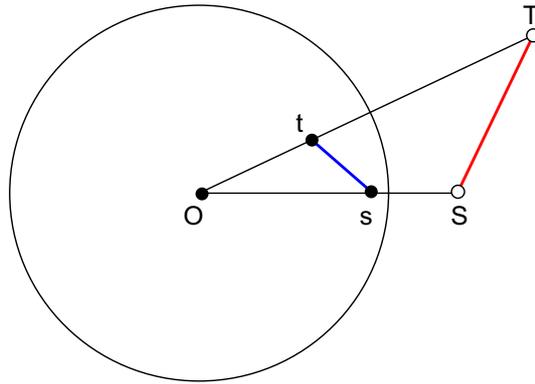


Abbildung 6: Ähnliche Dreiecke bei der Inversion am Kreis.

und wegen

$$|OT| = \frac{r^2}{|Ot|}. \quad (7)$$

Setzen wir (7) in (6) ein, so erhalten wir gerade die Behauptung.

Theorem von Ptolemaios:

Vorraussetzung: *Viereck ABCD mit $ABCD \in$ Kreis K.*

Behauptung: Die Summe der Produkte der gegenüberliegenden Seitenlängen ist gleich dem Produkt der Diagonalenlängen:

$$|AD| \cdot |BC| + |AB| \cdot |CD| = |AC| \cdot |BD| \quad (8)$$

Beweis:

Wir legen einen Inversionskreis K mit D als Mittelpunkt fest, der das Viereck komplett umfasst. Dann gilt:

$$|A'B'| + |B'C'| = |A'C'|$$

Mit obigem Hilfssatz folgt:

$$|A'B'| = \frac{r^2 \cdot |AB|}{|AD||BD|} \quad |B'C'| = \frac{r^2 \cdot |BC|}{|BD||CD|} \quad |A'C'| = \frac{r^2 \cdot |AC|}{|AD||CD|}.$$

Also:

$$\frac{|AB|}{|AD||BD|} + \frac{|BC|}{|BD||CD|} = \frac{|AC|}{|AD||CD|}$$

und daher:

$$|AB||CD| + |BC||AD| = |AC||BD| \quad \square$$

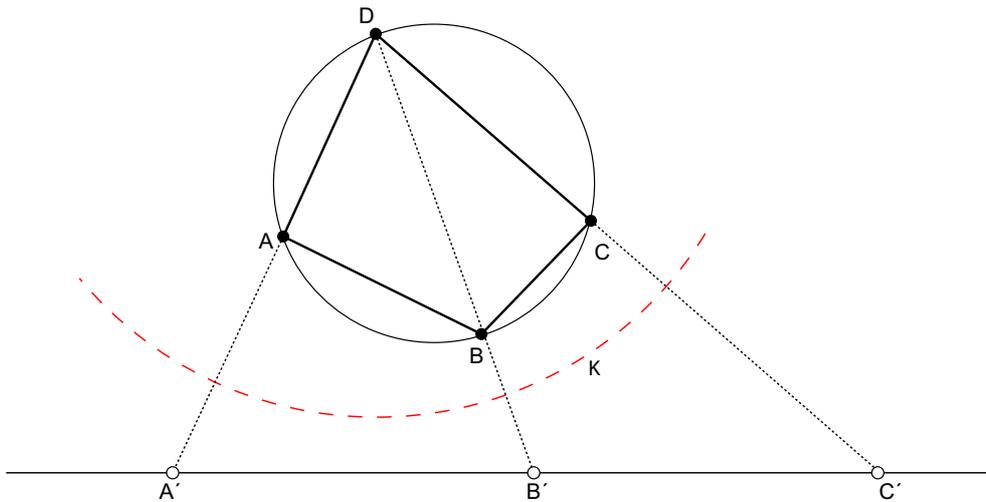


Abbildung 7: Anwendung der Inversion beim Beweis des Theorems von Ptolemaios.

4 Die stereografische Projektion

Da die Stützung $f(z) = \frac{1}{z}$ und die Inversion $g(z) = \frac{1}{\bar{z}}$, Kreise nicht nur auf Kreise und Geraden nicht nur auf Geraden, sondern auch Kreise auf Geraden und Geraden auf Kreise abbilden, ist es sinnvoll, diese Abbildungen nicht in der Zahlenebene sondern auf einer Kugel zu untersuchen. Die RIEMANN'schen Zahlenkugel hat den Radius $\frac{1}{2}$ und ihr Südpol liegt auf dem Ursprung. Die Stereografische Projektion ist die Abbildung, die jeden Punkt auf der Oberfläche der RIEMANN'schen Zahlenkugel einem Punkt der GAUSS'schen Zahlenebene zuordnet. Hierfür wird eine Gerade durch den Nordpol der Zahlenkugel und den entsprechenden Punkt auf der Kugeloberfläche gelegt. Der Schnittpunkt dieser Gerade mit der Zahlenebene ist der gesuchte Punkt auf der Zahlenebene. Der Nordpol $N(0,0,1)$ wird dabei dem formal eingeführten Punkt P_∞ zugeordnet. Punkte der Ebene werden auf die gleiche Weise Punkten der Kugeloberfläche zugeordnet. Interessanterweise werden Geraden und Kreise immer als Kreise auf die Kugeloberfläche projiziert. Der Beweis, dass jede Gerade als Kreis abgebildet wird ist denkbar einfach: Da die Schnittlinie einer Ebene und einer Kugel stets ein Kreis ist, ist auch das Bild einer beliebigen Gerade ein Kreis durch den Nordpol. Dieser Kreis entspricht der Schnittlinie von Kugel und Ebene, in der sowohl die Gerade, als auch der Nordpol liegt.

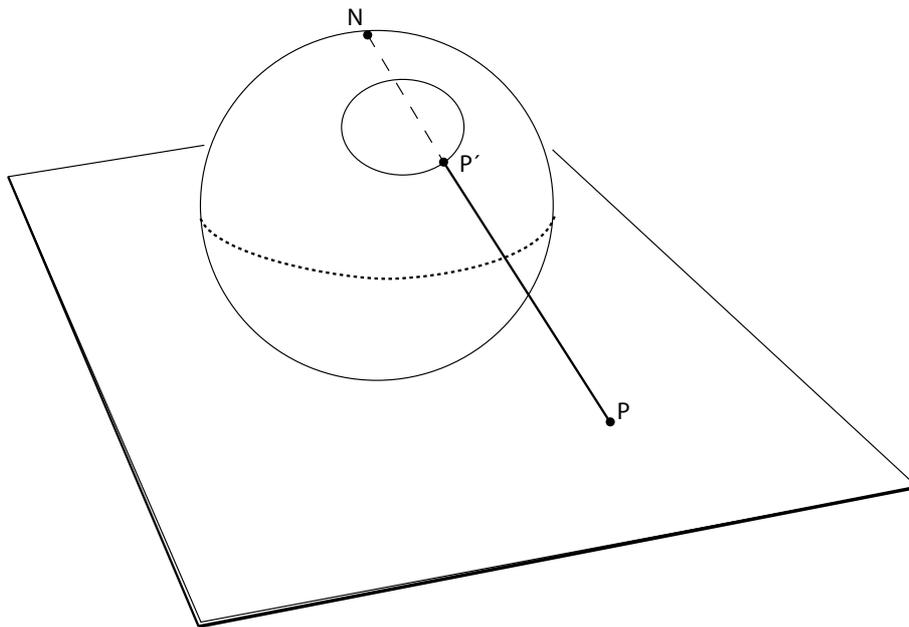


Abbildung 8: Die stereografische Projektion.

Satz:

Kreise werden bei der stereografischen Projektion wieder auf Kreise abgebildet.

Beweis:

Wir ziehen eine Gerade durch die Punkte N(Ordpol) $(0,0,1)$, $P'(u,v,w)$ und $P(x,y,0)$. Ausgehend von $(0,0,1) + \lambda((x,y,0) - (0,0,1))$ erhalten wir die Gerade $g : (\lambda x, \lambda y, -\lambda + 1)$. Durch Einsetzen in die Kugelgleichung $u^2 + v^2 + (w - \frac{1}{2})^2 = (\frac{1}{2})^2$, ergibt sich hier die Gleichung:

$$(\lambda x)^2 + (\lambda y)^2 + (-\lambda + \frac{1}{2})^2 = (\frac{1}{2})^2 \quad (9)$$

Davon ausgehend, das λ verschieden von 0 ist, ergibt sich:

$$(u, v, w) = \frac{(x, y, x^2 + y^2)}{x^2 + y^2 + 1} \quad (10)$$

Setzen wir diese Formeln für die entsprechenden Variablen in eine Ebenengleichung $au + bv + cw = d$ ein, so erhalten wir nach Umstellen:

$$ax + by = (d - c)(x^2 + y^2) + d \quad (11)$$

Dies ist die die Gleichung für einen Kreis in der komplexen Ebene.

Wie wirkt sich die Inversion am Einheitskreis auf die Punkte der Zahlenkugel aus? Es gilt:

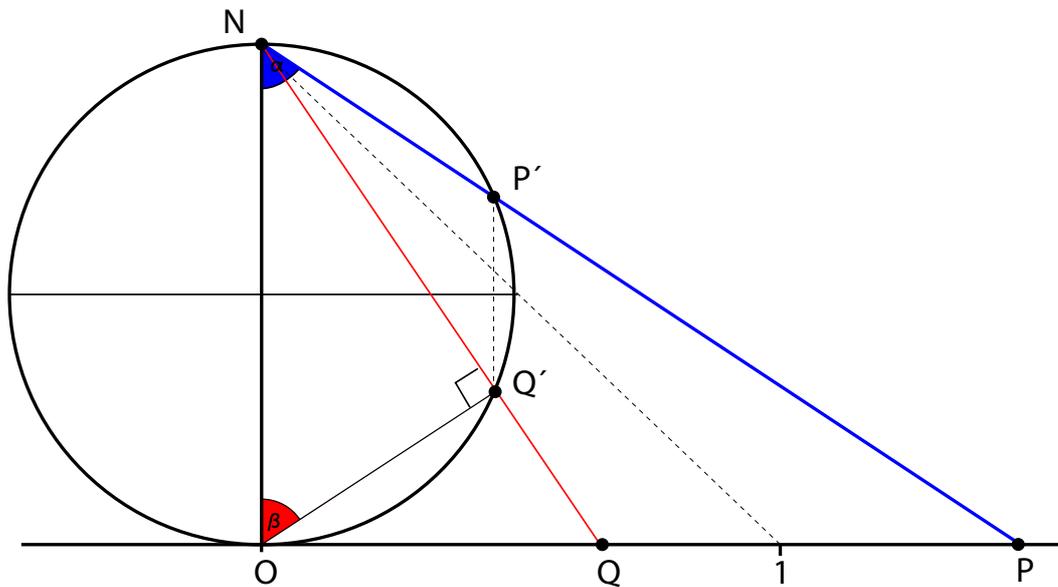


Abbildung 9: Spiegelung am Äquator und Inversion am Einheitskreis.

Satz:

Die Inversion am Einheitskreis entspricht in der Zahlenkugel einer Spiegelung am Äquator.

Beweis:

Wir betrachten die Punkte P, Q und deren Bildpunkte P', Q' nach der stereografischen Projektion. Es gilt dann:

$$\triangle NOP' \sim \triangle Q'ON,$$

da beide Dreiecke rechtwinklig sind und für die Kathetenverhältnisse

$$\frac{1}{|OP|} = \frac{|OQ|}{1}$$

gilt. Daher sind die Winkel $\angle ONP$ und $\angle Q'ON$ gleich groß, was bedeutet, dass Q' das Spiegelbild von P' am Äquator ist.

5 Die Möbiustransformationen

Eine Möbiustransformation ist eine Abbildung der Form

$$w = M(z) = \frac{az + b}{cz + d},$$

wobei $a, b, c, d \in \mathbb{C}$ und $ad - bc \neq 0$.

i) Ist $c = 0$, so ist

$$w = M(z) = \frac{az + b}{d} = a'z + b,$$

d.h. die Möbiustransformation ist in diesem Fall eine Verknüpfung einer Skalierung (um $|a'|$), einer Drehung (um $\arg(a')$) und einer Translation (um b).

ii) Ist $c \neq 0$, so setze $D := ad - bc$. Dann ist

$$w - \frac{a}{c} = \frac{az + b}{cz + d} - \frac{a}{c} = \frac{-D}{c(cz + d)}.$$

Die Möbiustransformation ist in diesem Fall eine Verknüpfung von:

- a) Verschiebungen (Translationen)
- b) Drehstreckungen
- c) Inversion am Einheitskreis und
- d) Spiegelung an der x-Achse.

Mit Hilfe der RIEMANN'schen Zahlenkugel können Möbiustransformationen veranschaulicht werden:

- Translationen entsprechen Verschiebungen der Kugel.
- Drehungen entsprechen Rotationen der Kugel entlang der vertikalen Achse.
- Streckungen entsprechen dem Anheben oder Absenken der Kugel.
- Inversionen am Kreis entsprechen Spiegelungen am Äquator.

Dieses wurde im Kurzfilm *Moebius Transformations Revealed* von Arnold und Rogness (<http://www.ima.umn.edu/arnold/moebius/>) gezeigt.

6 Literaturhinweise

- Niederdrenk-Felgner, C.: Themenheft: Komplexe Zahlen. Klett Verlag (2004).
- Rademacher, H.: Higher Mathematics from an Elementary Point of View, Birkhäuser, Boston u.a. (1982).
- Bakelman, I.J.: Spiegelung am Kreis, Leipzig (1976).

Lauschen zwecklos!

Teilnehmer:

Andrea Birth	Andreas-Oberschule
Nikolai Bobenko	Herder-Oberschule
Jonas Gätjen	Immanuel-Kant-Oberschule
Holger Hesse	Heinrich-Hertz-Oberschule
Julian Risch	Heinrich-Hertz-Oberschule
Sophie Spirkel	Evangelische Schule Frohnau

Gruppenleiter:

Jürg Kramer	Humboldt-Universität zu Berlin, Mitglied im DFG-Forschungszentrum MATHEON „Mathematik für Schlüsseltechnologien“
Anna v. Pippich	Humboldt-Universität zu Berlin, Mitglied im DFG-Forschungszentrum MATHEON „Mathematik für Schlüsseltechnologien“

Abhörsicheres Telefonieren mit dem Mobiltelefon oder die Sicherheit beim Einsatz von Chipkarten sind zwei von unzähligen Beispielen aus dem Alltagsleben, bei denen das sichere Verschlüsseln von Daten eine entscheidende Rolle spielt. Dabei sollte das Verschlüsseln dieser Daten so clever sein, dass ein Abhören durch Unbefugte wertlos ist, also: Lauschen zwecklos! Dies ist mit Mathematik möglich.

In unserem Sommerschul-Kurs haben wir ein 350 Jahre altes Resultat aus der Zahlentheorie hergeleitet, welches für das 1977 von R. Rivest, A. Shamir und L. Adleman erfundene Verschlüsselungsverfahren, das sogenannte *RSA-Verfahren*, die Grundlage bildet. Dies ist ein asymmetrisches Verschlüsselungsverfahren, das zwei verschiedene Schlüssel zum Ver- und Entschlüsseln verwendet. Weiter haben wir uns mit dem sogenannten *Faktorisierungsproblem* beschäftigt, auf welchem die Sicherheit des RSA-Verfahren beruht.

Ein moderneres Verfahren, das bei gleicher Sicherheitsleistung eine geringere Schlüssellänge benötigt, benutzt *elliptische Kurven*. Dabei verstehen wir unter einer elliptische Kurve eine kubische Kurve, die durch eine Gleichung der Form $y^2 = x^3 + a \cdot x^2 + b \cdot x + c$ mit $a, b, c \in \mathbb{Q}$ gegeben ist. Wir haben untersucht, wie elliptische Kurven zur Verschlüsselung herangezogen werden können. Schließlich haben wir die von uns erarbeiteten Verschlüsselungsverfahren programmiert.

1 Zahlentheoretische Grundlagen

1.1 Der größte gemeinsame Teiler

Definition 1.1. Die natürliche Zahl $d \in \mathbb{N}$ heißt größter gemeinsamer Teiler von $a \in \mathbb{Z}$ und $b \in \mathbb{Z}$, falls gilt:

- (1) $d|a$ und $d|b$;
- (2) Für alle $c \in \mathbb{Z}$ mit $c|a$ und $c|b$ gilt auch $c|d$.

Bezeichnung. $(a, b) :=$ größter gemeinsamer Teiler von a und b .

Beispiel. Die Zahlen $a = 30$ und $b = 12$ haben die gemeinsamen Teiler 1, 2, 3 und 6. Weiter gilt $1|6$, $2|6$, $3|6$ und $6|6$. Damit ist der größte gemeinsame Teiler von 30 und 12 gleich $(a, b) = (30, 12) = 6$.

1.2 Euklidischer Algorithmus

Normalerweise liefert die Primfaktorenzerlegung der Zahlen a und b auf einfache Weise den größten gemeinsamen Teiler. Zum Beispiel erhalten wir mit Hilfe der eindeutigen Zerlegungen $a = 30 = 2 \cdot 3 \cdot 5$ und $b = 12 = 2 \cdot 2 \cdot 3$ sofort den größten gemeinsamen Teiler $(a, b) = (30, 12) = 2 \cdot 3 = 6$.

Dieses Verfahren ist jedoch für große Zahlen für den Computer sehr aufwendig zu berechnen. Deshalb führen wir den Euklidischen Algorithmus ein.

Satz 1.1 (Euklidischer Algorithmus). *Seien $a, b \in \mathbb{Z}$ mit $a > b$ und $b \neq 0$. Wir betrachten dann die fortgesetzte Division mit Rest, welche zu dem Schema*

$$\begin{aligned} a &= q_1 \cdot b + r_1 & (0 < r_1 < |b|) \\ b &= q_2 \cdot r_1 + r_2 & (0 < r_2 < r_1) \\ r_1 &= q_3 \cdot r_2 + r_3 & (0 < r_3 < r_2) \\ &\vdots \\ r_{n-2} &= q_n \cdot r_{n-1} + r_n & (0 < r_n < r_{n-1}) \\ r_{n-1} &= q_{n+1} \cdot r_n + 0 \end{aligned}$$

führt. Dieses Verfahren bricht nach endlich vielen Schritten ab, d.h. es findet sich ein $n \in \mathbb{N}$ derart, dass $r_{n+1} = 0$ ist. Überdies ist der letzte nicht verschwindende Rest r_n ein größter gemeinsamer Teiler von a und b , d.h. es gilt

$$(a, b) = r_n.$$

Beweis. Da es nur endlich viele natürliche Zahlen r_1, \dots, r_n mit

$$0 \leq r_n < \dots < r_2 < r_1 < |b|$$

gibt, ist klar, dass die fortgesetzte Division mit Rest nach endlich vielen Schritten abbrechen muss. Sei r_n der letzte nicht verschwindende Rest. Wir zeigen zunächst, dass r_n die Eigenschaft (1) aus Definition 1.1 besitzt, wobei wir die Gleichheiten des obigen Schemas benutzen. Auf Grund der letzten Gleichung $r_{n-1} = q_{n+1} \cdot r_n$ dieses Schemas gilt

$$r_n | r_{n-1}. \quad (1.1)$$

Wegen $r_{n-2} = q_n \cdot r_{n-1} + r_n$ folgt mit (1.1), dass auch

$$r_n | r_{n-2}$$

gilt. Durch Fortsetzung dieses Verfahrens können wir damit sukzessiv die Eigenschaft (1) aus Definition 1.1 für r_n beweisen.

Nun zeigen wir, dass r_n die Eigenschaft (2) aus Definition 1.1 besitzt. Dazu beweisen wir, dass es $x, y \in \mathbb{Z}$ gibt, so dass

$$r_n = x \cdot a + y \cdot b \quad (1.2)$$

gilt. Dazu rollen wir das Schema der fortgesetzten Division mit Rest aus Satz 1.1 wie folgt rückwärts auf

$$\begin{aligned} r_n &= r_{n-2} - q_n \cdot r_{n-1} \\ r_n &= r_{n-2} - q_n \cdot (r_{n-3} - q_{n-1} \cdot r_{n-2}) \\ &= r_{n-2} \cdot (1 + q_n \cdot q_{n-1}) - r_{n-3} \cdot q_n \\ r_n &= (r_{n-4} - q_{n-2} \cdot r_{n-3}) \cdot (1 + q_n \cdot q_{n-1}) - r_{n-3} \cdot q_n \\ &= r_{n-4} \cdot (1 + q_n \cdot q_{n-1}) - r_{n-3} \cdot (q_{n-2} + q_n \cdot q_{n-1} \cdot q_{n-2} - q_n) \\ &\vdots \\ r_n &= x \cdot a + y \cdot b \end{aligned}$$

mit ganzen Zahlen $x, y \in \mathbb{Z}$, was (1.2) beweist. Sei nun $c \in \mathbb{Z}$ mit $c|a$ und $c|b$. Dann gilt auch

$$c|(x \cdot a + y \cdot b)$$

und somit wegen der Gleichheit (1.2) auch $c|r_n$. Insgesamt erhalten wir somit

$$(a, b) = r_n,$$

wie behauptet. □

Hierbei haben wir auch den für das Folgende wichtigen Satz bewiesen.

Satz 1.2 (Erweiterter Euklidischer Algorithmus). *Seien $a, b \in \mathbb{Z}$ mit $b \neq 0$. Dann gibt es $x, y \in \mathbb{Z}$, so dass gilt:*

$$(a, b) = x \cdot a + y \cdot b.$$

Sind speziell a und b teilerfremd, dann gilt

$$1 = x \cdot a + y \cdot b.$$

Beispiel. Für $a = 925$ und $b = 65$ berechnen wir

$$\begin{aligned} 925 &= 14 \cdot 65 + 15 \\ 65 &= 4 \cdot 15 + 5 \\ 15 &= 3 \cdot 5. \end{aligned}$$

Damit folgt, dass

$$(925, 65) = 5$$

gilt. Rollen wir das obige Schema rückwärts auf, erhalten wir

$$\begin{aligned} 5 &= 65 - 4 \cdot 15 \\ 5 &= 65 - 4 \cdot (925 - 14 \cdot 65) \\ 5 &= -4 \cdot 925 + 57 \cdot 65. \end{aligned}$$

Damit gilt

$$(925, 65) = -4 \cdot 925 + 57 \cdot 65.$$

1.3 Rechnen modulo p

Definition 1.2. Für $a, b \in \mathbb{Z}$ und eine natürliche Zahl $m > 0$ definieren wir die Operationen $\oplus : \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z}$ und $\odot : \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z}$ gemäß

$$\begin{aligned} a \oplus b &= R_m(a + b), \\ a \odot b &= R_m(a \cdot b); \end{aligned}$$

hierbei bezeichnet $R_m(n)$ den Rest der ganzen Zahl n nach Division durch m .

Beispiel. Ist $m = 5$, so gelten für $a = 5$ und $b = 3$ die Gleichheiten

$$\begin{aligned} 5 \oplus 3 &= R_5(5 + 3) = 3, \\ 5 \odot 3 &= R_5(5 \cdot 3) = 0. \end{aligned}$$

Definition 1.3. Wir definieren

$$a \equiv b \pmod{m} \iff R_m(a) = R_m(b),$$

in Worten: a heißt kongruent zu b modulo m , genau dann, wenn a und b nach Division durch m den gleichen Rest lassen.

Beispiel. Ist $m = 5$, so gilt für $a = 17$ und $b = 7$ die Äquivalenz

$$17 \equiv 7 \pmod{5} \iff R_5(17) = 2 = R_5(7),$$

d.h. 17 ist kongruent zu 7 modulo 5.

Bemerkung. Man kann bei einer Kongruenz modulo m fast wie bei echter Gleichheit rechnen. Zum Beispiel gilt für $a \equiv b \pmod{m}$ und $c \equiv d \pmod{m}$:

- (1) $a \pm c \equiv b \pm d \pmod{m}$,
- (2) $a \cdot c \equiv b \cdot d \pmod{m}$.

Beispiel. Wir erklären nun an Hand eines Beispiels, dass man die Gleichung

$$a \cdot x \equiv 1 \pmod{m},$$

modulo m lösen kann, d.h. wir suchen nach einer Lösung $x \in \mathbb{N}$ mit $0 \leq x < m$, vorausgesetzt, dass $(a, m) = 1$ ist. Dies ist grundlegend für die folgenden Kapitel. Sei dazu $a = 23$ und $m = 56$. Dann gilt

$$\begin{aligned} 23 \cdot x &\equiv 1 \pmod{56} \\ \iff 23 \cdot x + y \cdot 56 &\equiv 1 \pmod{56}, \end{aligned}$$

wobei $y \in \mathbb{Z}$ beliebig ist. Nun wird der Euklidische Algorithmus angewendet.

$$\begin{aligned} 56 &= 2 \cdot 23 + 10 \\ 23 &= 2 \cdot 10 + 3 \\ 10 &= 3 \cdot 3 + 1 \\ 3 &= 3 \cdot 1 + 0. \end{aligned}$$

Nun rollen wir den Euklidischen Algorithmus rückwärts auf.

$$\begin{aligned} 1 &= 10 - 3 \cdot 3 \\ 1 &= 10 - 3 \cdot (23 - 2 \cdot 10) \\ 1 &= 10 - 3 \cdot 23 + 6 \cdot 10 \\ 1 &= (56 - 2 \cdot 23) - 3 \cdot 23 + 6 \cdot (56 - 2 \cdot 23) \\ 1 &= 7 \cdot 56 - 17 \cdot 23. \end{aligned}$$

Damit gilt

$$\begin{aligned} 23 \cdot (-17) + 7 \cdot 56 &\equiv 1 \pmod{56} \\ \iff 23 \cdot (-17) + 23 \cdot 56 &\equiv 1 \pmod{56} \\ \iff 23 \cdot (-17 + 56) &\equiv 1 \pmod{56} \\ \iff 23 \cdot 39 &\equiv 1 \pmod{56}. \end{aligned}$$

Damit ist $x = 39$ eine Lösung der Gleichung $23 \cdot x \equiv 1 \pmod{56}$ mit $0 \leq x < 56$.

Bemerkung. Im Folgenden bezeichnen wir die Zahl $x \in \mathbb{N}$ mit $0 \leq x < m$ und $a \cdot x \equiv 1 \pmod{m}$ mit dem Symbol a^{-1} .

Bemerkung. Ist p eine Primzahl, dann ist die Menge $\mathbb{F}_p := \{0, 1, \dots, p-1\}$ der Reste modulo p ein Körper mit p Elementen.

1.4 Die Sätze von Fermat und Euler

Satz 1.3 (Satz von Fermat). *Es sei p eine Primzahl. Dann gilt für alle $a \in \mathbb{Z}$, die nicht Vielfache von p sind:*

$$a^{p-1} \equiv 1 \pmod{p}.$$

Beweis. Es sei nun p eine Primzahl und $a \in \mathbb{N}$ kein Vielfaches von p . Zunächst bemerken wir, dass sich die Vielfachen

$$a, 2a, 3a, \dots, (p-1)a$$

bis auf die Reihenfolge als

$$1 + k_1p, 2 + k_2p, 3 + k_3p, \dots, (p-1) + k_{p-1}p$$

darstellen lassen, wobei $k_1, \dots, k_{p-1} \in \mathbb{N}$ sind. Bilden wir das Produkt dieser Vielfachen, erhalten wir somit die Gleichheit

$$a \cdot 2a \cdot 3a \cdot \dots \cdot (p-1)a = (1 + k_1p) \cdot (2 + k_2p) \cdot (3 + k_3p) \cdot \dots \cdot (p-1 + k_{p-1}p),$$

welche man für ein $l \in \mathbb{Z}$ in der folgenden Form schreiben kann

$$\begin{aligned} (p-1)! \cdot a^{p-1} &= (p-1)! + l \cdot p \\ \iff (p-1)! \cdot a^{p-1} &\equiv (p-1)! \pmod{p}. \end{aligned} \quad (1.3)$$

Wegen $((p-1)!, p) = 1$ gibt es nach dem Satz vom erweiterten Euklidischen Algorithmus zwei Zahlen $x, y \in \mathbb{Z}$ mit $1 = x \cdot (p-1)! + y \cdot p$, d.h. mit

$$x \cdot (p-1)! \equiv 1 \pmod{p}. \quad (1.4)$$

Multiplizieren wir nun (1.3) mit x , so erhalten wir wegen (1.4) die Äquivalenz

$$\begin{aligned} x \cdot (p-1)! \cdot a^{p-1} &\equiv x \cdot (p-1)! \pmod{p} \\ \iff a^{p-1} &\equiv 1 \pmod{p}. \end{aligned}$$

Dies beweist die Behauptung. □

Der Schweizer Mathematiker Euler hat den kleinen Satz von Fermat wie folgt verallgemeinert.

Satz 1.4 (Satz von Euler). *Seien p, q verschiedene Primzahlen, $m = p \cdot q$ und $n = (p-1)(q-1)$. Dann gilt für alle $a \in \mathbb{Z}$, die teilerfremd zu m sind:*

$$a^n = a^{(p-1)(q-1)} \equiv 1 \pmod{m}.$$

Bemerkung. Es ist möglich, den Beweis analog zum Beweis des Satzes von Fermat zu führen. Wir verzichten allerdings darauf, den Beweis hier anzugeben.

2 Das RSA-Verfahren

Im folgenden Abschnitt wird gezeigt, wie das RSA-Verfahren funktioniert, und es wird bewiesen, dass es korrekt ist, d.h. dass der Empfänger die Nachricht des Senders immer richtig liest. Dieses Verfahren ist ein Public-Key-Kryptosystem, bei dem asymmetrisch chiffriert wird, das bedeutet, dass Absender A und Empfänger B verschiedene Schlüssel benutzen. Hierbei müssen sich A und B weder kennen, noch sich zu einem Schlüsselaustausch treffen.

Bevor der Austausch der Nachricht erfolgen kann, müssen folgende Vorbereitungen erfolgen: Der Empfänger B wählt zwei „große“ Primzahlen, d.h. zur Zeit etwa 200-stellige Primzahlen p und q , welche geheim gehalten werden müssen. Daraufhin berechnet er $m = p \cdot q$. Nun bestimmt B eine natürliche Zahl k , die zu $n = (p - 1) \cdot (q - 1)$ teilerfremd ist. Die Zahlen m und k bilden den öffentlichen Schlüssel, d.h. sie werden öffentlich an A übermittelt.

Der Absender A wandelt nun seine zu übermittelnde Nachricht in eine natürliche Zahl a ($1 < a < m$) um, z.B. mit Hilfe des ASCII-Codes. Danach verschlüsselt A die Nachricht a als

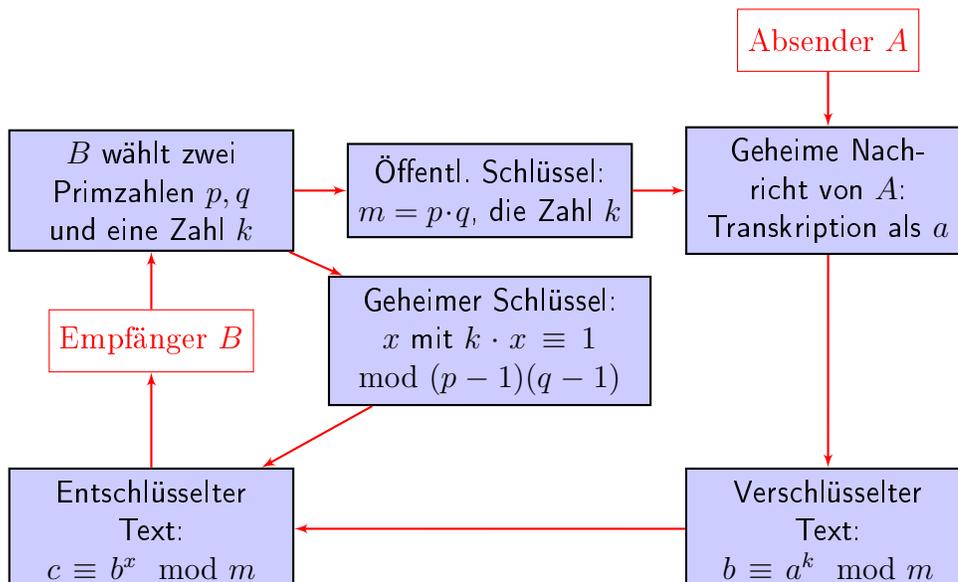
$$b \equiv a^k \pmod{m}$$

und sendet b öffentlich an B .

Damit der Empfänger B die Nachricht von A entschlüsseln kann, muss er zunächst eine ganze Zahl x bestimmt, die die Kongruenz $k \cdot x \equiv 1 \pmod{n}$ erfüllt. Mit dem so gewonnenen geheimen Schlüssel x berechnet er

$$c \equiv b^x \pmod{m}.$$

Damit ist die Nachricht entschlüsselt, denn es gilt $c = a$.



Nun wird gezeigt, dass das RSA-Verfahren korrekt ist. Dazu formulieren wir folgenden Satz:

Satz 2.1. *Seien p, q verschiedene Primzahlen und k eine natürliche Zahl, die zu $n = (p - 1) \cdot (q - 1)$ teilerfremd ist. Desweiteren seien a, b, c, m und x Zahlen entsprechend dem oben beschriebenen Vorgehen. Dann gilt $a \equiv c \pmod{m}$.*

Beweis. Aus $b \equiv a^k \pmod{m}$ und $c \equiv b^x \pmod{m}$ folgt

$$c \equiv (a^k)^x \equiv a^{kx} \pmod{m}.$$

Da $kx \equiv 1 \pmod{n}$ mit $n = (p - 1) \cdot (q - 1)$, existiert ein $y \in \mathbb{Z}$ mit

$$kx = 1 + yn.$$

Damit ergibt sich $a^{kx} = a^{1+yn} = a \cdot a^{yn} = a \cdot (a^n)^y$ und somit

$$c \equiv a \cdot (a^n)^y \pmod{m}.$$

Da a teilerfremd zu $m = p \cdot q$ ist, gilt nach Satz 1.4, dem Satz von Euler, dass $a^n \equiv 1 \pmod{m}$ gilt und damit

$$c \equiv a \cdot (a^n)^y \equiv a \cdot 1^y \equiv a \pmod{m},$$

d.h. $a \equiv c \pmod{m}$, wie behauptet. □

Beispiel. Zum besseren Verständnis betrachten wir ein kleines Beispiel. Der Empfänger B wählt die Primzahlen $p = 229$ und $q = 389$. Damit erhält er

$$n = (p - 1) \cdot (q - 1) = 228 \cdot 389 = 88464.$$

Nun wählt B beispielsweise $k = 43$. Da 43 eine Primzahl ist und n kein Vielfaches von 43 ist, ist k teilerfremd zu n , wie gewünscht. B gibt nun die Zahlen

$$\begin{aligned} m &= p \cdot q = 229 \cdot 389 = 89081, \\ k &= 43 \end{aligned}$$

öffentlich bekannt. Der Absender A transkribiert die Nachricht „PI“ mit Hilfe des ASCII-Codes als $a = 8073$ und übermittelt die verschlüsselte Nachricht

$$b \equiv 8073^{43} \equiv 30783 \pmod{89081}.$$

Währenddessen bestimmt B den geheimen Schlüssel x , indem er die Gleichung $k \cdot x \equiv 1 \pmod{n}$ löst. So erhält er die Lösung

$$x = 67891.$$

Nun kann der Empfänger B die Nachricht entschlüsseln, indem er $c \equiv b^x \pmod{m}$ berechnet; er erhält $c \equiv 30783^{67891} \equiv 8073 \pmod{89081}$, also die Nachricht „PI“.

Beispiel. Nun ein etwas realistischeres Beispiel. Der Empfänger B wählt die Primzahlen

$$\begin{aligned} p &= 1532495540865888858358347027150309183618739357528837633, \\ q &= 1532495540865888858358347027150309183618974467948366513. \end{aligned}$$

Damit erhält er

$$\begin{aligned} n &= (p - 1)(q - 1) \\ &= 2348542582773833227889480596789337027376043575908906788 \\ &\quad 406607163597747756552746892633980748733486828474179584. \end{aligned}$$

Nun wählt B wieder $k = 43$. Da 43 eine Primzahl ist und n kein Vielfaches von 43 ist, ist k teilerfremd zu n , wie gewünscht. B gibt nun die Zahlen

$$\begin{aligned} m &= p \cdot q \\ &= 2348542582773833227889480596789337027376043575908906791 \\ &\quad 471598245329525473269440946934599115971200653951383729, \\ k &= 43 \end{aligned}$$

öffentlich bekannt. Der Absender A transkribiert die Nachricht „MATHEMATIK“ mit Hilfe des ASCII-Code als

$$a = 77658472697765847375$$

und übermittelt die verschlüsselte Nachricht

$$\begin{aligned} b &\equiv 77658472697765847375^{43} \\ &\equiv 217819882953579407544224571425479958308096036559243448 \\ &\quad 980351224602119873496097431290450386913902399435406279 \pmod{m}. \end{aligned}$$

Währenddessen bestimmt B den geheimen Schlüssel x , indem er die Gleichung $k \cdot x \equiv 1 \pmod{n}$ löst. So erhält er

$$\begin{aligned} x &= 491555424301499977930356403979163563869404469376282816 \\ &\quad 178127080753016972301737721714088993920962359448084099. \end{aligned}$$

Nun kann der Empfänger B die Nachricht entschlüsseln, indem er $c \equiv b^x \pmod{m}$ bestimmt; er erhält

$$c \equiv 77658472697765847375 \pmod{m},$$

also wieder die Nachricht „MATHEMATIK“.

Die Sicherheit des RSA-Verfahrens beruht auf dem Faktorisierungsproblem. Da bisher kein Algorithmus zur Faktorisierung großer Zahlen allgemein bekannt ist, kann eine Person C, die die Kommunikation mitschreibt und somit die Zahlen m , k und b erhält, die Nachricht nicht entschlüsseln, da sie dafür m in seine Primfaktoren p und q zerlegen muss, um n und dann x , den Schlüssel, den sie zum Entschlüsseln benötigt, zu berechnen. Da sich die Leistung der Computer nach dem Mooreschen Gesetz, welches laut Intel bis 2029 Bestand haben soll, ständig verbessert und man somit immer größere Zahlen in annehmbarer Zeit zerlegen kann und dadurch auch früher abgefangene Nachrichten leichter lesen kann, werden zur Verschlüsselung meist sehr viel höhere Primzahlen als eigentlich nötig wären benutzt. In Kapitel 4 haben wir uns mit verschiedenen Faktorisierungsmethoden beschäftigt.

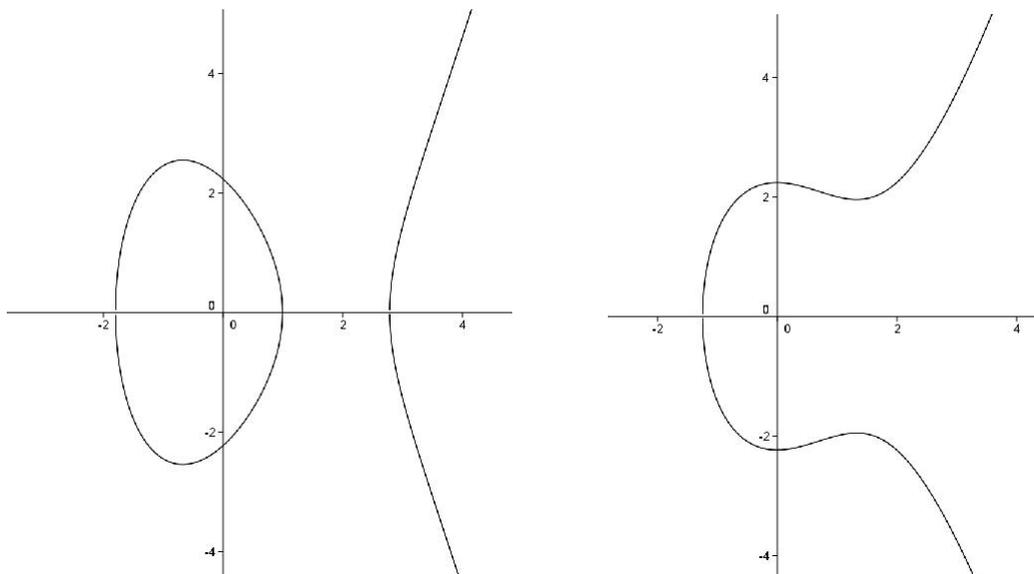
3 Elliptische Kurven

3.1 Definition

Definition 3.1. Eine kubische Kurve C , die durch die Gleichung

$$y^2 = x^3 + a \cdot x^2 + b \cdot x + c \tag{3.1}$$

festgelegt ist, heißt elliptische Kurve, falls das kubische Polynom auf der rechten Seite drei verschiedene Nullstellen hat (zwei dieser Nullstellen können auch komplex sein). Falls die Koeffizienten a, b, c rationale Zahlen sind, sagen wir, dass die elliptische Kurve C über den rationalen Zahlen \mathbb{Q} definiert ist.



3.2 Gruppenstruktur

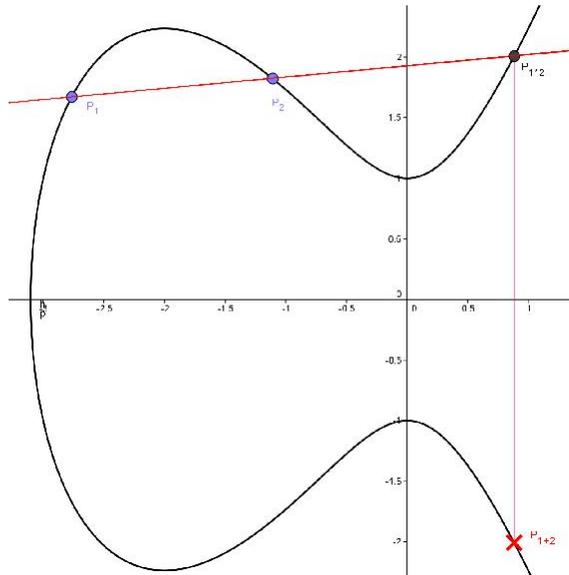
Definition 3.2. Die Menge der rationalen Punkte der elliptischen Kurve (3.5) ist gegeben durch die Menge

$$C(\mathbb{Q}) = \{(x, y) \in \mathbb{Q}^2 \mid y^2 = x^3 + a \cdot x^2 + b \cdot x + c\} \cup \{O\},$$

wobei O der unendlich ferne Punkt mit den Koordinaten (∞, ∞) ist; dieser ist von allen anderen Punkten der Kurve unendlich weit entfernt und, wenn man sich von einem Punkt in eine beliebige Richtung unendlich weit weg bewegt, dann landet man im unendlich fernen Punkt.

Die Besonderheit der Menge der rationalen Punkte $C(\mathbb{Q})$ liegt in der Existenz einer additiven Struktur, wodurch diese Menge zu einer abelschen Gruppe wird.

Nun zeigen wir, dass die Menge der rationalen Punkte $C(\mathbb{Q})$ zusammen mit einer von uns noch zu definierenden Operation $+$ eine abelsche Gruppe bildet. Um diese Addition zweier Punkte $P = (x_P, y_P)$ und $Q = (x_Q, y_Q)$ zu definieren, wird als erstes eine Gerade an diese beiden Punkte angelegt. Es entsteht immer ein dritter Schnittpunkt mit der elliptischen Kurve; diesen nennen wir $T = (x_T, y_T)$. Die Summe $R := P + Q$ von P und Q erhält man geometrisch, wenn der eben ermittelte Punkt T an der x -Achse gespiegelt wird, d.h. die Koordinaten (x_R, y_R) von $R = P + Q$ sind gegeben durch $x_R = x_T$ und $y_R = -y_T$.



Um mit algebraischen Methoden auf die Koordinaten des Punktes T zu kommen, betrachten wir die Gerade

$$y = \lambda x + \nu \tag{3.2}$$

mit Steigung λ und Achsenabschnitt ν , welche durch die Formeln

$$\lambda = \frac{y_Q - y_P}{x_Q - x_P} \quad \text{und} \quad \nu = y_P - \lambda x_P = y_Q - \lambda x_Q$$

gegeben sind. Nun setzt man die Geradengleichung (3.2) in die Gleichung der elliptischen Kurve (3.5) ein. Es ergibt sich eine Polynomgleichung vom Grad 3 in x , welche x_P und x_Q als Nullstellen besitzt. Mit Hilfe des Vietaschen Wurzelsatzes lässt sich dann x_T berechnen. Wir haben nämlich:

$$\begin{aligned} (\lambda x + \nu)^2 &= y^2 = x^3 + a \cdot x^2 + b \cdot x + c, \text{ d.h.} \\ x^3 + (a - \lambda^2)x^2 + (b - 2\lambda\nu)x + (c - \nu^2) &= 0. \end{aligned}$$

Der Vietasche Wurzelsatz für letztere kubische Gleichung besagt, dass die Summe der drei Nullstellen gleich (-1) mal der Koeffizient des quadratischen Terms ist, d.h.

$$\begin{aligned} x_P + x_Q + x_T &= -(a - \lambda^2), \text{ d.h.} \\ x_T &= \lambda^2 - a - x_P - x_Q. \end{aligned} \tag{3.3}$$

Die y -Koordinate y_T von T berechnet sich wie folgt. Man setzt (3.3) in (3.2) ein und erhält

$$y_T = \lambda x_T + \nu. \tag{3.4}$$

Nach Spiegelung an der x -Achse ergeben sich die Koordinaten von $R = P + Q$ zu (x_R, y_R) mit $x_R = x_T$ und $y_R = -y_T$.

Ein Spezialfall stellt die Addition eines Punktes $P = (x_P, y_P)$ mit sich selbst, also die Verdoppelung dar, weil hierbei die Tangente in P angelegt wird, deren zweiter Schnittpunkt mit der elliptischen Kurve T ist. Wenn man diesen so erhaltenen Punkt spiegelt, ergibt sich die Summe $P + P = 2P$. Hierbei wird die Steigung der Tangente an P durch

$$\left. \frac{dy}{dx} \right|_{(x,y)=(x_P,y_P)} = \frac{f'(x_P)}{2y_P}$$

gegeben.

Satz 3.1. *Die Menge der rationalen Punkte $C(\mathbb{Q})$ der elliptischen Kurve (3.5) bildet zusammen mit der oben definierten Addition $+$ eine abelsche Gruppe.*

Beweis. Die Tatsache, dass bei der Addition $+$ zweier beliebiger Punkte aus der Menge der rationalen Punkte einer elliptischen Kurve, die Summe wieder durch einen rationalen Punkt repräsentiert wird, ist aus der Gleichung (3.3) ersichtlich: Da sowohl die Steigung λ , der Koeffizient vor dem quadratischen Teil, sowie die

x -Koordinaten der Punkte P und Q rational sind, muss auch x_R rational sein. Daraus folgt, dass auch y_R rational ist und es sich bei der Summe um einen rationalen Punkt handelt. Somit ist $(C(\mathbb{Q}), +)$ abgeschlossen.

Auf den Beweis der Assoziativität der Operation $+$ soll hier verzichtet werden. Durch dynamische Geometriesoftware kann diese jedoch veranschaulicht werden. Das neutrale Element bezüglich der Operation $+$ ist der unendlich ferne Punkt O , da für alle Punkte P in $C(\mathbb{Q})$ die Gleichung $P + O = P = O + P$ gilt.

Das inverse Element bezüglich der Operation $+$ für den Punkt $P = (x_P, y_P)$ ist der Punkt $-P := (x_P, -y_P)$, da $P + (-P) = O = (-P) + P$ für alle P in $C(\mathbb{Q})$ gilt.

Wie aus den Gleichungen schließlich ersichtlich ist, gilt auch das Kommutativgesetz für $(C(\mathbb{Q}), +)$, da es irrelevant ist, ob man $P + Q$ oder $Q + P$ berechnet, das Ergebnis ist gleich.

3.3 Elliptische Kurven über endlichen Körpern

Unser Ziel ist es, elliptische Kurven in kryptographischen Verfahren einzusetzen. Dafür muss das Ergebnis der Entschlüsselung eines zuvor verschlüsselten Textes eindeutig sein. Dazu bietet sich das Rechnen mit elliptischen Kurven modulo p an. Wir führen dazu den Körper mit p Elementen ein.

Der endliche Körper \mathbb{F}_p wird als Menge dargestellt durch die Zahlen $\{0, \dots, p-1\}$, wobei p eine Primzahl ist. Addition, Subtraktion und Multiplikation werden dabei modulo p gerechnet, wie es im Abschnitt 1.3 beschrieben wurde; die Division durch von 0 verschiedene Zahlen wird auch wie im Abschnitt 1.3 vorgestellt mit Hilfe des Erweiterten Euklidischen Algorithmus durchgeführt.

Eine elliptische Kurve C über dem endlichen Körper \mathbb{F}_p wird durch die Kongruenz

$$y^2 \equiv x^3 + a \cdot x^2 + b \cdot x + c \pmod{p} \quad (3.5)$$

definiert, wobei das kubische Polynom rechter Hand drei verschiedene Nullstellen modulo p haben muss; die Koeffizienten a, b, c sind hierbei im Körper \mathbb{F}_p zu wählen. Die \mathbb{F}_p -rationalen Punkte von C sind gegeben durch die Menge

$$C(\mathbb{F}_p) = \{(x, y) \in \mathbb{F}_p^2 \mid y^2 \equiv x^3 + a \cdot x^2 + b \cdot x + c \pmod{p}\} \cup \{O\}.$$

Beispiel. Wir betrachten die elliptische Kurve C über dem Körper \mathbb{F}_{23} , welche durch die Kongruenz

$$y^2 \equiv x^3 + x \pmod{23}$$

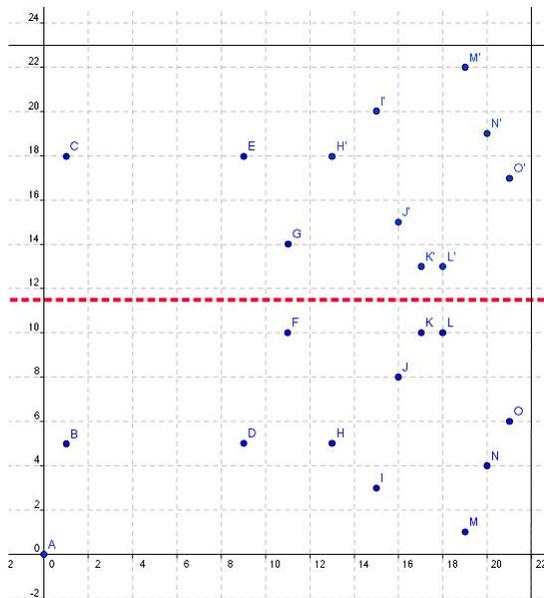
gegeben ist. Die Menge der \mathbb{F}_{23} -rationalen Punkte ist gegeben durch

$$\begin{aligned} C(\mathbb{F}_{23}) = \{ & O, (0, 0), (1, 5), (1, 18), (9, 5), (9, 18), (11, 10), (11, 13), (13, 5), \\ & (13, 18), (15, 3), (15, 20), (16, 8), (16, 15), (17, 10), (17, 13), (18, 10), \\ & (18, 13), (19, 1), (19, 22), (20, 4), (20, 19), (21, 6), (21, 17)\}. \end{aligned}$$

Beispielsweise gilt $(9, 5) \in C(\mathbb{F}_{23})$, da

$$5^2 \equiv 25 \equiv 738 \equiv 9^3 + 9 \pmod{23}$$

gilt.



Der Graphik kann man entnehmen, dass auch über dem Körper \mathbb{F}_p eine Achsensymmetrie besteht.

Analog den über den rationalen Zahlen angestellten Betrachtungen wollen wir im folgenden auch die (endliche) Menge der \mathbb{F}_p -rationalen Punkte als abelsche Gruppe erkennen. Dazu adaptieren wir die für die Addition hergeleiteten Formeln (3.3) und (3.4) an das Rechnen auf elliptischen Kurven modulo p . Für die Koordinaten (x_R, y_R) der „Summe“ $R = P + Q$ der beiden Punkte $P = (x_P, y_P)$ und $Q = (x_Q, y_Q)$ mit $x_P \not\equiv x_Q \pmod{p}$ gilt

$$x_R \equiv \lambda^2 - a - x_P - x_Q \pmod{p}, \quad (3.6)$$

$$y_R \equiv -\lambda x_R - \nu \pmod{p}, \quad (3.7)$$

wobei

$$\lambda \equiv (y_P - y_Q) \cdot (x_P - x_Q)^{-1} \pmod{p},$$

$$\nu \equiv y_P - \lambda x_P \pmod{p}.$$

Der negative Punkt $-P$ eines Punktes $P = (x_P, y_P) \in C(\mathbb{F}_p)$ ist durch $-P = (x_P, -y_P)$ gegeben.

Für den Spezialfall, dass $x_P \equiv x_Q \pmod p$ und $y_P \not\equiv y_Q \pmod p$ gilt $P + Q = O$; andernfalls ziehen wir die Verdoppelungsformeln für den Punkt $P + P = 2P$ mit den Koordinaten (x_R, y_R) heran, d.h. wir verwenden die Formeln

$$x_R \equiv \lambda^2 - a - 2x_P \pmod p, \quad (3.8)$$

$$y_R \equiv -\lambda x_R - \nu \pmod p, \quad (3.9)$$

wobei

$$\lambda \equiv (3x_P^2 + 2ax_P + b) \cdot (2y_P)^{-1} \pmod p,$$

$$\nu \equiv y_P - \lambda x_P \pmod p.$$

3.4 Analogon zum Diffie-Hellman Schlüsselaustausch

Alice und Bob einigen sich auf einen Punkt Q einer elliptischen Kurve C über dem endlichen Körper \mathbb{F}_p . Nun wählt Alice im Geheimen ein n und schickt Bob den Punkt $A = n \cdot Q = Q + \dots + Q \in C(\mathbb{F}_p)$ (Q wird n -mal zu sich selbst addiert). Analog dazu wählt Bob im Geheimen ein m und schickt Alice $B = m \cdot Q \in C(\mathbb{F}_p)$. Alice und Bob berechnen beide den geheimen Schlüssel

$$S := (n \cdot m) \cdot Q = n \cdot B = m \cdot A.$$

Auch wenn Charly Alice und Bob belauscht hat und somit A , B und Q kennt, kann er daraus nicht so einfach auf S , den geheimen Schlüssel, schließen. Das Problem für Charlie besteht darin, dass er zwar die Produkte $A = n \cdot Q$ und $B = m \cdot Q$, den Punkt Q und die Primzahl p kennt, daraus jedoch nicht so leicht auf m oder n schließen kann. Dies führt uns zum diskreten Logarithmus Problem:

Diskretes Logarithmus Problem für elliptische Kurven über \mathbb{F}_p :

Gegeben sind die Punkte $P, Q \in C(\mathbb{F}_p)$ mit der Eigenschaft $Q = k \cdot P$.

Gesucht ist k , welches der diskrete Logarithmus von Q zur Basis P genannt wird.

Diskretes Logarithmus Problem für die multiplikative Gruppe \mathbb{F}_p^\times :

Gegeben sind die zur Primzahl p teilerfremden ganzen Zahlen a, b mit der Eigenschaft $b \equiv a^k \pmod p$.

Gesucht ist k , welches der diskrete Logarithmus von b zur Basis a genannt wird.

Es gibt bis heute noch keine schnellen Algorithmen zur Lösung dieses Problems. Eine Möglichkeit ist die Berechnung der Vielfachen von P , bis Q erreicht wird, bzw. der Potenzen von a , bis b erreicht wird. Einige der gegenwärtig existierenden Algorithmen sind: der Babystep-Giantstep-Algorithmus, der Pohlig-Hellman-Algorithmus, der Index-Calculus-Algorithmus und die Pollard-Rho-Methode. Diese sind jedoch aufgrund ihrer langen Rechenzeit nicht praxisrelevant.

4 Faktorisierung

Das Faktorisierungsproblem ist eine klassische Aufgabe aus der Zahlentheorie. Die Aufgabe lautet, zu einer gegebenen Zahl alle Primfaktoren zu ermitteln. Für große Zahlen ist diese Aufgabe nur schwer zu lösen, insbesondere, wenn es sich um sogenannte schwere Zahlen handelt, d.h. Zahlen, die nur große Primfaktoren besitzen. Beispielsweise benötigten 600 Mitarbeiter und 1600 Rechner für die Faktorisierung einer 129-stelligen Dezimalzahl im Jahr 1994 ganze acht Monate. Im diesem Kapitel wollen wir einige Methoden zur Faktorisierung vorstellen.

4.1 Faktorisierung nach Fermat

Satz 4.1. *Es sei n eine ungerade, positive, natürliche Zahl. Dann kann man n faktorisieren, indem man für $t = [\sqrt{n}] + 1, [\sqrt{n}] + 2, \dots$ prüft, ob $t^2 - n$ eine Quadratzahl ist. Falls dem so ist, sind $t + \sqrt{t^2 - n}$ und $t - \sqrt{t^2 - n}$ Teiler von n .*

Beweis. Die Fermat-Faktorisierung beruht auf der zweiten bzw. dritten binomischen Formel. Es gilt nämlich

$$(t - \sqrt{t^2 - n})(t + \sqrt{t^2 - n}) = t^2 - (t^2 - n) = n.$$

□

Diese Methode funktioniert besonders gut, falls die Teiler von n relativ nahe beieinander liegen, da dann ihre Differenz, d.h.

$$(t + \sqrt{t^2 - n}) - (t - \sqrt{t^2 - n}) = 2\sqrt{t^2 - n},$$

relativ klein ist und somit $t^2 - n$ verhältnismäßig schnell als Quadratzahl erkannt wird, womit dann die gesuchte Faktorisierung gefunden ist.

4.2 Faktorisierung nach Pollard

Die Idee der Faktorisierung nach Pollard besteht darin, eine Zahl zu finden, die ein Vielfaches von einem Primteiler der vorgelegten natürlichen Zahl n ist, aber nicht von n selbst. Durch Bestimmung des größten gemeinsamen Teilers dieser Zahl mit n erhält man einen nichttrivialen Teiler von n . Indem man annimmt, dass n einen Primfaktor p besitzt, so dass $(p - 1)$ relativ kleine Primfaktoren hat, kann man n mit der $(p - 1)$ -Methode nach Pollard wie folgt faktorisieren.

Algorithmus. Wir wählen zunächst ein beliebiges $B \in \mathbb{N}$ und ein dazu passendes $k \in \mathbb{N}$, so dass k ein Vielfaches aller natürlichen Zahlen kleiner gleich B ist.

Die Zahl k könnte beispielsweise als das Produkt aller echten Teiler von $(p-1)$ gewählt werden. Man hofft nun, dass $(p-1)|k$ gilt. Außerdem wählt man ein $a \in \mathbb{N}$ mit $2 \leq a \leq (n-2)$ und bestimmt $a^k \pmod n$. Mit Hilfe des Euklidischen Algorithmus bestimmt man nun den größten gemeinsamen Teiler $(a^k - 1, n)$. Da aufgrund des Kleinen Satzes von Fermat wegen $(p-1)|k$ die Kongruenz

$$a^k \equiv 1 \pmod p$$

besteht, ergibt sich $p|(a^k - 1)$. Da voraussetzungsgemäß $p|n$ gilt, folgt $p|(a^k - 1, n)$, d.h., falls nicht $a^k \equiv 1 \pmod n$ ist, liefert diese Methode mit dem größten gemeinsamen Teiler $(a^k - 1, n)$ einen nichttrivialen Teiler von n .

4.3 Faktorisierung mit elliptischen Kurven

Im Jahr 1987 entwickelte H. W. Lenstra einen Faktorisierungsalgorithmus, welcher elliptische Kurven benutzt. Dieser ist von großer praktischer Bedeutung, weil er kleine Primfaktoren von n besonders schnell entdeckt.

In diesem Abschnitt sei n eine große natürliche Zahl, die nicht durch 2 und 3 teilbar ist und einen (noch unbekanntem) Primfaktor $p > 3$ besitzt. Zuerst wählt man eine beliebige elliptische Kurve C , welche durch die Gleichung

$$y^2 = x^3 + b \cdot x + c$$

mit $b, c \in \mathbb{Z}$ gegeben ist, und einen beliebigen Punkt $P = (x_P, y_P) \in C(\mathbb{Q})$.

Da die Primzahl p unbekannt ist, kann die Kurve C nicht über dem endlichen Körper \mathbb{F}_p betrachtet werden. Stattdessen rechnen wir modulo n , weswegen wir mit der folgenden Definition beginnen.

Definition 4.1 (Modulorechnung auf \mathbb{Q}). Es seien $n \in \mathbb{N}$ und $x_1, x_2 \in \mathbb{Q}$, derart, dass die Nenner von x_1 und x_2 teilerfremd zu n sind, d.h., derart, dass sie keine echten gemeinsamen Teiler mit n besitzen. Dann schreiben wir

$$x_1 \equiv x_2 \pmod n,$$

falls der Zähler des gekürzten Bruches $x_1 - x_2$ durch n teilbar ist.

Beispiel. Es sei $x_1 = 1/3$ und $x_2 = 11/5$. Für $n = 4$ gilt $(3, 4) = (5, 4) = 1$ und $x_1 - x_2 = 4/15$, also

$$\frac{1}{3} \equiv \frac{11}{5} \pmod 4.$$

Satz 4.2 (Kleinster nicht-negativer Rest). *Es sei n eine positive natürliche Zahl. Für alle $x \in \mathbb{Q}$ mit zu n teilerfremdem Nenner existiert genau ein $m \in \mathbb{N}$ mit $0 \leq m < n$ derart, dass*

$$x \equiv m \pmod n \tag{4.1}$$

gilt.

Bezeichnung. Die eindeutige Zahl m aus Satz 4.2 wird als „kleinster nicht-negativer Rest“ von x modulo n oder kurz als $x \pmod n$ bezeichnet.

Beweis. Existenz: Es sei r/q die gekürzte Bruchdarstellung von x ; hierbei ist der Nenner q teilerfremd zu n . Wegen $(n, q) = 1$ existieren $a, b \in \mathbb{Z}$ mit

$$a \cdot n + b \cdot q = 1.$$

Multiplikation dieser Gleichung mit r und Umstellung liefert die Kongruenz

$$r - (b \cdot r) \cdot q \equiv 0 \pmod n.$$

Indem wir nun $m \in \mathbb{N}$ mit $0 \leq m \leq (n-1)$ und $m \equiv b \cdot r \pmod n$ wählen, erhalten wir die Kongruenz

$$r - m \cdot q \equiv 0 \pmod n.$$

Wir haben

$$\frac{r}{q} - m = \frac{r - m \cdot q}{q};$$

diesen Bruch kann man nicht weiter kürzen, da $m \cdot q$ offensichtlich ein Vielfaches von q ist, r und q aber teilerfremd zueinander sind. Konstruktionsgemäß gilt für den Zähler $r - m \cdot q$ die Kongruenz

$$r - m \cdot q \equiv 0 \pmod n,$$

womit der Existenzbeweis geführt ist.

Eindeutigkeit: Angenommen, es existieren verschiedene $m_1 \in \mathbb{N}$ und $m_2 \in \mathbb{N}$ mit $0 \leq m_1, m_2 \leq (n-1)$, welche (4.1) erfüllen, dann gilt mit $x = r/q$ wie oben:

$$\begin{aligned} r - m_1 \cdot q &\equiv r - m_2 \cdot q \pmod n \\ \Leftrightarrow r - m_1 \cdot q &= r - m_2 \cdot q + \lambda \cdot n \\ \Leftrightarrow m_2 - m_1 &= \frac{\lambda \cdot n}{q} \end{aligned}$$

mit einem $\lambda \in \mathbb{Z}$. Da m_1 und m_2 natürliche Zahlen, und n und q teilerfremd sind, muss also λ ein Vielfaches von q sein, d.h. $|m_2 - m_1| \geq n$, im Widerspruch zu $0 \leq m_1, m_2 \leq n-1$. Damit ist auch die Eindeutigkeit bewiesen. \square

Beispiel. Es sei $x_1 = 1/3$ und $x_2 = 11/5$. Für $n = 4$ gilt

$$\frac{1}{3} \equiv \frac{11}{5} \equiv 3 \pmod 4.$$

Bei der Methode von Lenstra berechnen wir schrittweise das Vielfache kP ($k \in \mathbb{N}$) des gewählten Punktes $P \in C(\mathbb{Q})$ modulo n . Dies ist jedoch nur möglich, falls die auftretenden Nenner in jedem Rechenschritt zu n teilerfremd sind. Es gilt der folgende Satz.

Satz 4.3. *Seien n und C wie oben, wobei zusätzlich $(4b^3 + 27c^2, n) = 1$ gelte. Außerdem seien $P = (x_P, y_P), Q = (x_Q, y_Q) \in C(\mathbb{Q})$ mit $P \neq -Q$ und die rationalen Koordinaten x_P, y_P und x_Q, y_Q besitzen zu n teilerfremde Nenner. Dann sind die folgenden Aussagen äquivalent:*

- (a) *Die Summe $R := P + Q \in C(\mathbb{Q})$ besitzt rationale Koordinaten x_R, y_R mit zu n teilerfremdem Nenner.*
- (b) *Für jede Primzahl q mit der Eigenschaft $q|n$ gilt:*

$$R := P + Q \not\equiv O \pmod{q}$$

auf der elliptischen Kurve $C(\mathbb{F}_q)$.

Beweis. (a) \Rightarrow (b): Seien P, Q und $R = P + Q \in C(\mathbb{Q})$ mit rationalen Koordinaten, deren Nenner relativ prim zu n sind, gegeben. Weiter sei q ein beliebiger Primfaktor von n .

Falls $x_P \not\equiv x_Q \pmod{q}$ gilt, dann folgt sofort aus den Formeln (3.6) und (3.7) für die Addition modulo q , wobei $a = 0$ ist, dass $R \not\equiv O \pmod{q}$ ist.

Falls $x_P \equiv x_Q \pmod{q}$ gilt, unterscheiden wir folgende zwei Fälle: Falls $P = Q$ gilt, dann ist $R = 2P$, und die Koordinaten von R sind modulo q gegeben durch die Formeln (3.8), bzw. (3.9), wobei $a = 0$ ist. Wir müssen also zeigen, dass der Nenner von $2y_P$ nicht durch q teilbar ist. Angenommen, q teilt den Nenner von $2y_P$, dann muss q auch den Zähler von $3x_P^2 + b$ teilen, da der Nenner von x_R nicht durch q teilbar ist. Daraus folgt, dass das kubische Polynom der elliptischen Kurve C an der Stelle x_P eine doppelte Nullstelle modulo q besitzt, da Funktion und erste Ableitung dort den Wert 0 annehmen, im Widerspruch zu unserer Voraussetzung $(4b^3 + 27c^2, n) = 1$. Damit kann q nicht den Nenner von $2y_P$ teilen, was (b) beweist. Falls $P \neq Q$ gilt, kann man auf ähnliche Weise einen Widerspruch herbeiführen.

(b) \Rightarrow (a): Es sei (b) erfüllt. Wir müssen zeigen, dass die Koordinaten x_R, y_R Nenner teilerfremd zu n besitzen, d.h., dass jeder Primteiler q von n diese Nenner nicht teilt. Sei nun ein Primteiler q von n fixiert.

Falls $x_P \not\equiv x_Q \pmod{q}$ gilt, dann folgt sofort aus den Formeln (3.6) und (3.7) für die Addition modulo q , dass hier offensichtlich sind alle Nenner teilerfremd zu q sind, was (a) beweist.

Falls $x_P \equiv x_Q \pmod{q}$ gilt, dann folgt aus der Voraussetzung $R \not\equiv O \pmod{q}$, dass $y_P \equiv y_Q \not\equiv 0 \pmod{q}$ gelten muss. Ist nun $P = Q$, dann folgt damit wieder

aus den Additionsformeln (3.8), bzw. (3.9), dass die Koordinaten x_R, y_R Nenner besitzen, welche nicht durch q teilbar sind, d.h. Nenner, welche teilerfremd zu n sind, was (a) beweist. Falls $P \neq Q$ gilt, so verfahren wir auf ähnliche Weise. \square

Algorithmus (Methode von Lenstra). Sei n eine große natürliche Zahl, die nicht durch 2 und 3 teilbar ist und einen Primfaktor $p > 3$ besitzt. Zuerst wählt man eine beliebige elliptische Kurve C , welche durch die Gleichung

$$y^2 = x^3 + b \cdot x + c$$

mit $b, c \in \mathbb{Z}$ gegeben ist, und einen beliebigen Punkt $P = (x_P, y_P) \in C(\mathbb{Q})$. Daraufhin prüft man, ob $(4b^3 + 27c^2, n) = 1$ gilt, d.h. ob das kubische Polynom $x^3 + b \cdot x + c$ drei verschiedene Nullstellen modulo q für jeden Primfaktor q von n besitzt. Im Falle $1 < (4b^3 + 27c^2, n) < n$ hat man bereits einen Teiler von n gefunden und ist fertig. Im Falle $(4b^3 + 27c^2, n) = n$ wählt man eine neue elliptische Kurve und beginnt von vorne.

Als nächstes wählt man sich zwei Grenzen B und C und ein

$$k = q_1^{\alpha_1} \cdot \dots \cdot q_r^{\alpha_r} \in \mathbb{N}$$

als Produkt aller Primzahlpotenzen $q_j^{\alpha_j} \leq C$, wobei q_j eine Primzahl und $\alpha_j \in \mathbb{N}$ ($j = 1, \dots, r$) ist. B soll dabei eine obere Grenze für die Primteiler q_j von k sein, d.h. es gilt $q_j \leq B$ für $j = 1, \dots, r$. Falls B sehr groß ist, ist die Wahrscheinlichkeit höher, dass $kP \equiv O \pmod{p}$ für ein p mit $p|n$, allerdings wird mehr Zeit für die Berechnung von kP benötigt. Falls man einen Primfaktor der Größe $q \sim \sqrt{n}$ sucht, so wählt man C (nach dem Satz von Hasse) so, dass $q + 1 + 2\sqrt{q} < C$ gilt. Mit diesem k berechnet man nun schrittweise das Vielfache kP des Punktes P modulo n . Zuerst berechnet man $q_1P, q_1(q_1P), \dots, q_1^{\alpha_1}P$, danach $q_2(q_1^{\alpha_1}P), q_2(q_2 \cdot q_1^{\alpha_1}P), \dots, q_2^{\alpha_2}q_1^{\alpha_1}P$, und so fort. Wenn nun die Berechnung eines dieser Vielfachen fehlschlägt, dann liegt eine rationale Koordinate mit einem Nenner vor, welcher nicht teilerfremd zu n ist. Mit der Bestimmung des größten gemeinsamen Teilers dieses Nenners und n erhält man also entweder einen echten Teiler von n und man ist fertig, oder man erhält n selbst. In diesem Fall wiederholt man den Algorithmus mit einer neuen elliptischen Kurve und einem neuen Punkt. Genauso verfährt man, falls die Berechnung von kP in keinem der Schritte fehlschlägt.

Beispiel. Sei $n = 5429$. Zuerst wählen wir die elliptische Kurve C , welche durch die Gleichung $y^2 = x^3 + 2 \cdot x - 2$ gegeben ist, und den Punkt $P = (1, 1) \in C(\mathbb{Q})$. Da $4 \cdot 2^3 + 27 \cdot 2^2 = 140 = 2^2 \cdot 5 \cdot 7$, gilt $(4 \cdot 2^3 + 27 \cdot 2^2, 5429) = 1$. Wir wählen $B = 3$. Suchen wir einen Primfaktor der Größe $\sqrt{n} \sim 73$, so wählen wir wegen $73 + 1 + 2\sqrt{73} < 92$ die Schranke $C = 92$. Damit ist $k = 2^6 3^4$. Berechnen wir nun schrittweise die Vielfachen $2P, 2(2P), \dots, 2^6P, 3(2^6P)$, und so fort, so schlägt die Berechnung bei $3^2 2^6P$ fehl, d.h. wir erhalten einen Nenner welcher nicht teilerfremd zu n ist, sondern mit n den größten gemeinsamen Teiler 61 besitzt.

Schuss und Tor – Flug eines Balls

Mathematische Beschreibung, Eigenschaften, Visualisierungen

Teilnehmer:

Paul Grau	Immanuel-Kant-Oberschule
Matthias Holz	Herder-Oberschule
Lukas Neumann	Herder-Oberschule
Andreas Dietrich	Heinrich-Hertz-Oberschule
Benjamin Herfort	Immanuel-Kant-Oberschule
Artemij Amiranashvili	Herder-Oberschule

Gruppenleiter:

René Lamour	Humboldt-Universität zu Berlin, Mitglied im DFG-Forschungszentrum MATHEON „Mathematik für Schlüsseltechnologien“
-------------	--

Als mathematisches Modell eines Problems bezeichnet man ein System von Gleichungen, dessen Lösung die realen Gegebenheiten ausreichend gut beschreibt. Im Allgemeinen existieren für solche Gleichungen keine expliziten Lösungen, so dass man auf Näherungsverfahren zu ihrer Berechnung zurückgreifen muss.

Wir haben ausgehend von den physikalischen Grundlagen möglichst realistische Modelle des Fluges eines Balles in Form von Differentialgleichungen aufgestellt. Diese Gleichungen haben wir mittels numerischer Verfahren gelöst und durch die Variation von Einflussparametern, z.B. in Reibungsgesetzen und der Kraftrichtungen, versucht reale Bahnen wie beim Fußball oder Tischtennis zu modellieren.

1 Physikalische Grundlagen

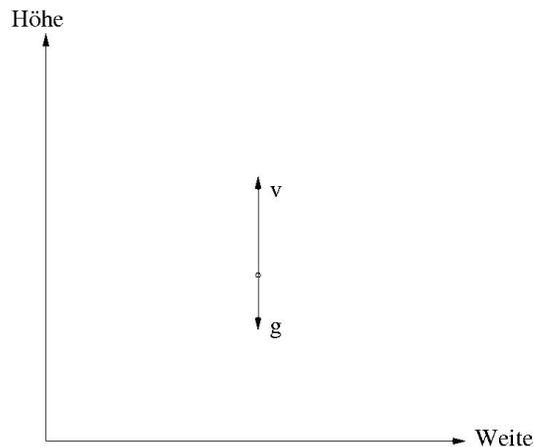
Bei der Betrachtung des Wurfes müssen wir mehrere Kräfte berücksichtigen. Es wirkt die Gewichtskraft, die Luftreibung, doch auch den Magnuseffekt gilt es mit einzubeziehen. Beginnen wollen wir jedoch mit den Newton'schen Axiomen:

1. Ein Körper mit der Masse m ist in Ruhe oder in gleichförmiger Bewegung, solange keine Kraft F auf ihn wirkt (Trägheitsgesetz).
2. Es gilt: $F = \frac{d}{dt}(mv) \Rightarrow F = \frac{d}{dt}(ms')$ (Newton'sches Grundgesetz);
 v sei die Geschwindigkeit und s der Weg.
3. Aktio = Reaktio (Wechselwirkungsgesetz).
4. Das Superpositionsprinzip, auch Überlagerungsprinzip genannt, beinhaltet die Addition von Vektoren.

Besonders das zweite Newton'sche Axiom ermöglicht es uns, den Flug des Balles mit Hilfe von Differentialgleichungen zu beschreiben.

1.1 Gravitationskraft

Die Gravitationskraft g wirkt nur senkrecht nach unten und ist eine Komponente des schrägen Wurfes.

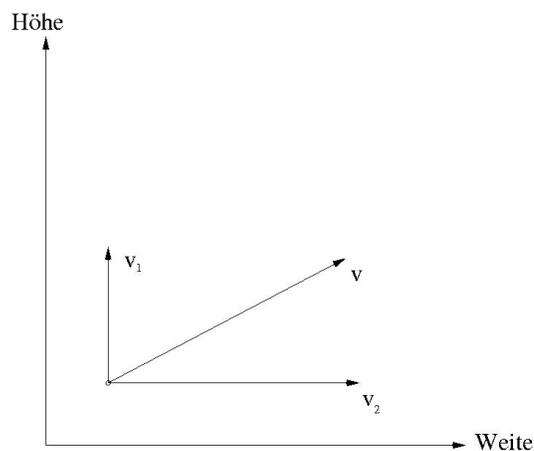


Aus der Schule kennen wir für den senkrechten Wurf den Zusammenhang

$$x(t) = x_0 - \frac{g}{2}t^2 + v_0t$$

x_0 und v_0 sind die Startposition und -geschwindigkeit.

Der Geschwindigkeitsvektor v_0 wirkt hier nur in eine Richtung, senkrecht nach oben, wohingegen beim schrägen Wurf auch eine Geschwindigkeitskomponente in der Horizontale existiert. In diese Richtung ist die Geschwindigkeit gleichförmig.

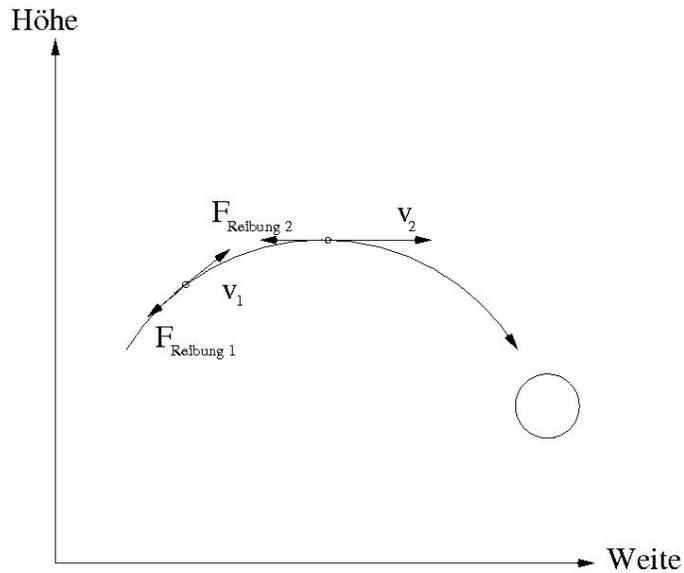


1.2 Luftwiderstand

Die Luftreibung wirkt immer entgegen der Wurfrichtung bzw. Flugrichtung (siehe Grafik) und hängt linear bis quadratisch von der Größe der Geschwindigkeit ab.

R sei die Reibungskraft, ε die spezifische Stoffkonstante. s ist bei niedrigen Geschwindigkeiten 1 und nimmt bei höheren Geschwindigkeiten bis auf 2 zu. $|v|$ bezeichnet die Länge des Vektors v und entspricht der Wurzel der Summe der Quadrate der einzelnen Komponenten

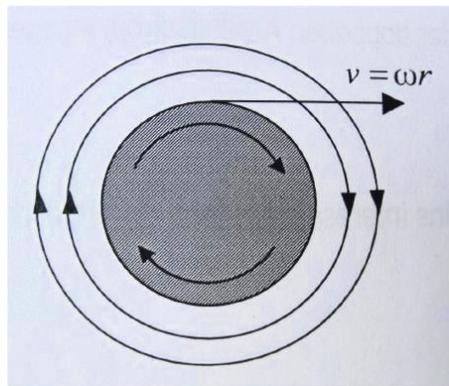
$$|v| = \sqrt{\sum_{i=1}^3 v_i^2}.$$



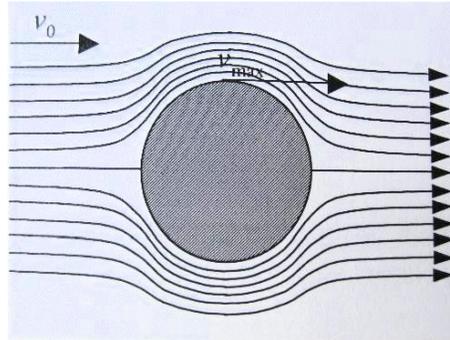
$$R = -\varepsilon v |v|^{s-1} \text{ mit } s \geq 1$$

1.3 Magnuseffekt

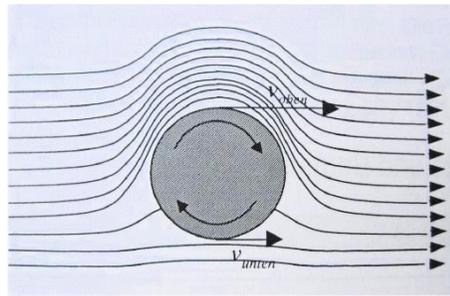
Betrachten wir zunächst einen um sich selbst rotierenden Ball mit dem Radius r . Die Luftmassen um ihn herum werden auf Grund der Reibung an der Balloberfläche ebenfalls in Bewegung versetzt und es entsteht eine Kreisströmung.



Wenn dagegen der Ball nicht rotiert und von einer laminaren Strömung umströmt wird, werden die Teilchen mit der gleichen Geschwindigkeit oberhalb und unterhalb abgelenkt. Hier wirkt der Magnuseffekt noch nicht.



Wenn beide Effekte miteinander verbunden werden, dann verändert sich das Stromlinienbild. Die Geschwindigkeit der Teilchen, die den Ball oberhalb umströmen ist höher als die der Teilchen, die ihn unterhalb umströmen.



Dadurch ändert sich das Druckverhältnis. Daraus resultieren unterschiedliche Drücke und der Ball wird in Richtung des höheren Druckes abgelenkt. Diese Kraft hat die Größe

$$|M| = \pi \rho \omega r |v|.$$

Dabei ist ρ die Luftdichte, ω die Rotationsgeschwindigkeit und v die Fluggeschwindigkeit des Balles.

Die reale Flugbahn des Balles kommt durch die Überlagerung der 3 Effekte Gravitation, Luftreibung und Magnuseffekt zustande. Dies gilt es nun mit Hilfe mathematischer Differentialgleichungen auszudrücken.

2 Differentialgleichungen

2.1 Ohne Reibung

Wir suchen eine Gleichung zur Beschreibung der Wurfbahn in Abhängigkeit von der Zeit. Nach dem 2. Newton'schen Axiom gilt:

$$F(t) = (mv(t))' = ms(t)''$$

mit $s(t), v(t), F(t) \in \mathbb{R}^3$, wobei die Masse m als konstant angenommen wird.

Wir formen das System 2. Ordnung in ein System 1. Ordnung um.

Dazu definieren wir einen sechsdimensionalen Vektor x , der in den ersten 3 Komponenten den Ort $s(t)$ und in den letzten 3 Komponenten die Geschwindigkeit $v(t)$ enthält. Dann enthält der sechsdimensionalen Vektor x' , sowohl die Geschwindigkeit v als auch die Kraft dividiert durch die Masse $F(t)/m$. Für diesen Vektor gilt demnach:

$$\begin{aligned}x'_1 &= x_4 \\x'_2 &= x_5 \\x'_3 &= x_6\end{aligned}$$

Wenn man die Reibung vernachlässigt, dann gilt für x :

$$x' = \begin{pmatrix} x_4 \\ x_5 \\ x_6 \\ -g \\ 0 \\ 0 \end{pmatrix}$$

wobei g die Fallbeschleunigung ist.

Also ist

$$\begin{aligned}x'_4 &= -g \\ \Rightarrow \int_{t_0}^t x'_4(\xi) d\xi &= - \int_{t_0}^t g d\xi \\ \Leftrightarrow x_4(t) - x_4(t_0) &= -g(t - t_0) \\ \Leftrightarrow x_4(t) &= x_4(t_0) - g(t - t_0)\end{aligned}$$

Wegen $x'_1 = x_4$ folgt:

$$\int_{t_0}^t x'_1(\xi) d\xi = \int_{t_0}^t (x_4(t_0) - g(\xi - t_0)) d\xi$$

$$\Leftrightarrow x_1(t) - x_1(t_0) = x_4(t_0)(t - t_0) - \frac{g}{2}(t - t_0)^2$$

$$\Leftrightarrow x_1(t) = x_4(t_0)(t - t_0) - \frac{g}{2}(t - t_0)^2 + x_1(t_0)$$

Offensichtlich entspricht das der theoretischen physikalischen Formel für den senkrechten reibungslosen Wurf.

Zur exakten Bestimmung der Lösung benötigt man zusätzlich Anfangswerte. Das führt auf ein Anfangswertproblem:

$$x' = f(x, t)$$

$$x(t_0) = x_0$$

2.2 Mit Reibung

Für den Betrag der Reibungskraft R gilt:

$$|R| = -\varepsilon|v|^s.$$

mit $s \geq 1$. Nun wollen wir die Richtung von R bestimmen.

Da $|\alpha v| = \sqrt{\sum_{i=1}^3 v_i^2 \alpha^2} = |\alpha||y|$, gilt $|\frac{y}{|y|}| = 1$, d.h. der Vektor $|\frac{y}{|y|}|$ hat die Länge 1.

Also ist:

$$R = -\varepsilon \frac{v}{|v|} |v|^s = \begin{pmatrix} -\varepsilon v_1 |v|^{s-1} \\ -\varepsilon v_2 |v|^{s-1} \\ -\varepsilon v_3 |v|^{s-1} \end{pmatrix} = \begin{pmatrix} -\varepsilon x_4 |v|^{s-1} \\ -\varepsilon x_5 |v|^{s-1} \\ -\varepsilon x_6 |v|^{s-1} \end{pmatrix}$$

2.3 Mit Magnuseffekt

Wir beschränken uns bei der Darstellung des Magnuseffekts auf die Drehung um eine vertikale Achse. Der Vektor der Magnuskraft sei M , wobei daher $M_1 = 0$ gilt. Der dreidimensionale Vektor y habe die gleiche Richtung wie M . Per Definition gilt für orthogonale Vektoren:

$$v \perp y \Leftrightarrow v^T y = 0$$

Also gilt :

$$v_1 y_1 + v_2 y_2 + v_3 y_3 = 0$$

y ist nicht eindeutig bestimmbar, da es z.B. mehrere Vektoren unterschiedlicher Länge gibt, die die Gleichung erfüllen. Wir können allerdings einen bestimmten Vektor finden, indem wir $y_3 = 1$ setzen. Dann ist $y_2 = -\frac{v_3}{v_2} = -\frac{x_6}{x_5}$

Also ist

$$y = \begin{pmatrix} 0 \\ -\frac{x_6}{x_5} \\ 1 \end{pmatrix}$$

ein Vektor in Richtung der Magnuskraft. Außerdem kennen wir bereits den Betrag der Magnuskraft: $|M| = \pi \rho \omega r^2 |v|$

Diesen Betrag multiplizieren wir mit einem Vektor der Länge 1 in Richtung von y , um auf die tatsächlich wirkende Magnuskraft zu kommen:

$$M = \begin{pmatrix} 0 \\ -\frac{x_6}{x_5} \\ 1 \end{pmatrix} \frac{\pi \rho \omega r^2 |v|}{\sqrt{\frac{x_6^2}{x_5^2} + 1}} = \begin{pmatrix} 0 \\ -\frac{x_6}{x_5} \\ 1 \end{pmatrix} \frac{x_5 \pi \rho \omega r^2 |v|}{\sqrt{x_6^2 + x_5^2}} = \begin{pmatrix} 0 \\ -x_6 \\ x_5 \end{pmatrix} \pi \rho \omega r^2$$

3 Näherungsverfahren

Um Näherungsverfahren für ein Anfangswertproblem zu konstruieren, wird die Differentialgleichung

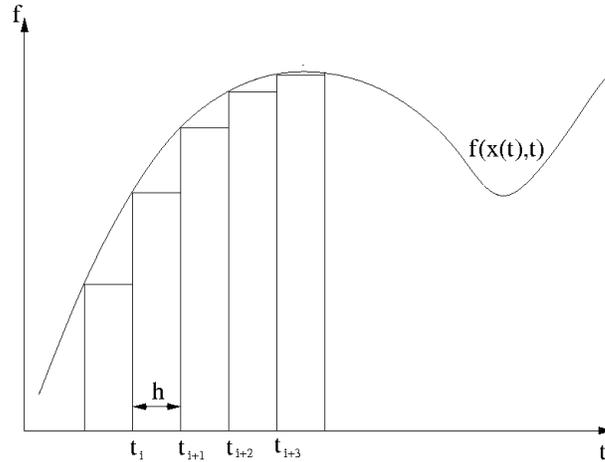
$$x'(t) = f(x(t), t)$$

von t_0 bis t integriert.

$$x(t) - x(t_0) = \int_{t_0}^t f(x(\xi), \xi) d\xi$$

$$x(t) = x_0 + \int_{t_0}^t f(x(\xi), \xi) d\xi$$

3.1 Das explizite Euler-Verfahren



Wie in der Graphik zu erkennen, bestimmen wir näherungsweise das Integral mithilfe von Rechtecken.

Mit $x_0 = x(t_0)$ und t_0 haben wir die Anfangswerte für das Näherungsverfahren. Für die Bestimmung der weiteren Reihenglieder ergibt sich die rekursive Bildungsvorschrift:

$$x_{i+1} = x_i + hf(x_i, t_i)$$

Die Summe der Volumen der Rechtecke ist die Annäherung an das Integral, deren Genauigkeit vom Abstand h abhängt.

3.1.1 Genauigkeit des Euler-Verfahrens

Zur Bestimmung der Genauigkeit setzen wir in die Näherungsformel die exakten Werte ein. Der entstehende Fehler τ bezeichnet den sogenannten *lokalen Diskretisierungsfehler*.

Mithilfe der Taylor-Reihen

$$\begin{aligned} a(t+h) &= \frac{h^0}{0!}a(t) + \frac{h^1}{1!}ha'(t) + \frac{h^2}{2!}a''(t) + \frac{h^3}{3!}a'''(t) + O(h^4) \\ &= a(t) + ha'(t) + \frac{h^2}{2!}a''(t) + \frac{h^3}{3!}a'''(t) + O(h^4) \end{aligned}$$

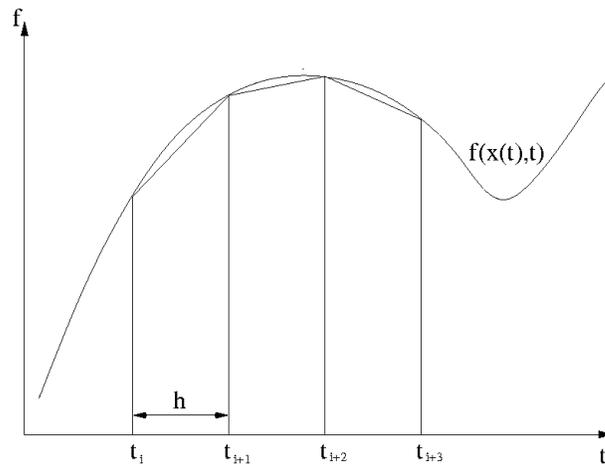
approximieren wir die Funktion durch eine Polynomfunktion. Das ist möglich, wenn $x(t)$ hinreichend oft differenzierbar ist.

$$x(t+h) = x(t) + hx'(t) + \frac{h^2}{2}x''(t) + O(h^3)$$

Das setzen wir nun in die Gleichung für den lokalen Diskretisierungsfehlers für das Eulerverfahren ein:

$$\begin{aligned} \tau_E &= \frac{x(t) + hx'(t) + \frac{h^2}{2}x''(t) + O(h^3) - x(t)}{h} - f(x(t), t) \\ &= \frac{hx'(t) + \frac{h^2}{2}x''(t) + O(h^3)}{h} - x'(t) \\ \Rightarrow \tau_E &= \frac{h}{2}x''(t) + O(h^2) \end{aligned}$$

3.2 Das Euler-Heun-Verfahren



Nun bestimmen wir das Integral mithilfe einer genaueren Annäherung durch Trapeze. Für die Fläche eines Trapezes ergibt sich:

$$A_{i+1} = \frac{h}{2}(f(x_i, t_i) + f(x_{i+1}, t_{i+1}))$$

Dadurch erhalten wir die rekursive Bildungsvorschrift:

$$x_{i+1} = x_i + \frac{h}{2}(f(x_i, t_i) + f(x_{i+1}, t_{i+1}))$$

Dies ist allgemein bekannt als *Trapezregel*. Dafür reicht allerdings x_i als Anfangswert für jeden Schritt nicht aus, sodass wir x_{i+1} unter Anwendung des expliziten Euler-Verfahrens näherungsweise bestimmen:

$$\tilde{x}_{i+1} = x_i + hf(x_i, t_i)$$

Eingesetzt erhalten wir das Euler-Heun-Verfahren

$$x_{i+1} = x_i + \frac{h}{2}(f(x_i, t_i) + f(\tilde{x}_{i+1}, t_{i+1}))$$

3.2.1 Genauigkeit des Euler-Heun-Verfahren

Wir berechnen wieder den lokalen Diskretisierungsfehler, indem wir das exakte Ergebnis in das Euler-Heun-Verfahren einsetzen. Allerdings beschränken wir uns der Einfachheit halber auf skalare f :

$$\tau_{EH} = \frac{x(t+h) - x(t)}{h} - \frac{1}{2}(f(x(t), t) + \underbrace{f(x(t) + hf(x(t), t), t+h)}_{y(t+h)})$$

Für $y(t+h)$ wenden wir wieder die Taylor-Reihe an. Das Problem dabei besteht darin, dass wir die Taylor-Reihe bzgl. h in t anwenden müssen:

$$y(t+h) = f(x(t), t) + h \underbrace{(f_x f + f_t)}_{x''(t)} + \frac{h^2}{2} x'''(t) + O(h^3)$$

wegen

$$x''(t) = \frac{d}{dt} x'(t) = \frac{d}{dt} f(x(t), t) = f_x x' + f_t = f_x f + f_t$$

Beim Einsetzen in die Gleichung für den lokalen Diskretisierungsfehler erhalten wir:

$$\begin{aligned} \tau_{EH} &= \frac{hx' + \frac{h^2}{2}x'' + \frac{h^3}{3!}x''' + O(h^4)}{h} - \frac{1}{2}(2x' + hx'' + \frac{h^2}{2}x''' + O(h^3)) \\ &= -\frac{h^2}{12}x''' + O(h^3) \end{aligned}$$

Für Polynomfunktionen 2. Grades wird x''' und alle weiteren Ableitungen 0, so dass $\tau_{EH} = 0$ folgt. Also ist das Verfahren für quadratische Funktionen exakt.

4 Implementierung

Zur Implementierung der Wurfvorgänge benutzen wir das Programm Matlab. Dies zeichnet sich durch einfache Handhabung und praktische Anwendung bei komplexen mathematischen Problemen aus.

Das Prinzip der Implementierung besteht aus der Darstellung der Differentialgleichung und der näherungsweise Berechnung dieser.

4.1 Euler'sches Näherungsverfahren

Wir gehen von der Gleichung $x_{i+1} = x_i + h * f(x_i, t_i)$ aus. Diese implementieren wir, indem wir von Anfangswerten ausgehen:

```
x=x0 ;  
t=t0 ;
```

und näherungsweise mit Hilfe einer for-Schleife die weiteren Werte berechnen.

```
for i=1:N          N steht für die Anzahl der Schritte  
    x=x+h*f eval(f,x,t) ;    h steht für die Schrittweite  
    t=t+h ;              t wird um die Schrittweite erhöht  
end
```

Der Befehl feval berechnet die Funktion f mit den Parametern x und t .

4.2 Funktion f

Die Funktion gibt die zeitliche Änderung der Werte von x_1 bis x_6 aus. Die zeitliche Änderung, also Ableitung der Koordinaten x_1 bis x_3 ist die Geschwindigkeit zu diesem Zeitpunkt. Diese Werte sind in x_4 bis x_6 gespeichert und werden in y_1 bis y_3 ausgegeben.

```
y(1)=x(4) ;  
y(2)=x(5) ;  
y(3)=x(6) ;
```

Nacheinander berücksichtigen wir die Veränderung der Geschwindigkeit (also die Beschleunigung) beim freien Flug, Flug mit Reibung und Flug mit Magnuseffekt.

4.2.1 Freier Flug

Beim freien Flug wirkt nur die Gravitationskraft in x_1 Richtung und demnach muss die Funktion f nur durch folgenden Code vervollständigt werden.

```
g=9.81 ;  
y(4)=-g ;  
y(5)=0.0 ;  
y(6)=0.0 ;
```

4.2.2 Flug mit Reibung

Durch den Luftwiderstand verändern sich die wirkenden Kräfte. Nach der physikalischen Herleitung gilt $R = -\varepsilon v|v|^{s-1}$, wobei ε der Reibungsfaktor ist. Zuerst berechnen wir den Betrag von v :

```
betrag_v=sqrt(x(4)*x(4)+x(5)*x(5)+x(6)*x(6)) ;
```

Jetzt berechnen wir die Reibung für die einzelnen Komponenten und subtrahieren sie von den Beschleunigungen:

```
Ra=reibungsfaktor*betrag_v^(s-1) ;  
R(1)=Ra*x(4) ;  
R(2)=Ra*x(5) ;  
R(3)=Ra*x(6) ;  
  
y(4)=-g-R(1) ;  
y(5)=-R(2) ;  
y(6)=-R(3) ;
```

4.2.3 Flug mit Magnuseffekt

Für den Magnuseffekt gilt die Formel $F = \pi \rho \omega r v$. Die Kraft berechnen wir nun in Abhängigkeit von v für die senkrechte Drehachse.

```

rho=1.293 ;
omega=2*(2*r*pi) ;
v2=v(1) ;
v3=v(2) ;
Magnus=[-v3;v2]*pi*rho*omega*r*r ;

```

Nun werden für y_5 und y_6 die Kraft des Magnuseffektes addiert, y_4 bleibt gleich.

```

y(4)=-g-R(1) ;
y(5)=-R(2)+Magnus(1) ;
y(6)=-R(3)+Magnus(2) ;

```

4.3 Euler-Heun'sches Näherungsverfahren

Diese andere Möglichkeit der Approximation liefert genauere Werte als das Euler'sche Näherungsverfahren. Es muss nur die for-Schleife verändert werden.

```

for i=1:N
    fn=feval(f,x,t);
    xL=x+h*fn ;
    t=t+h ;
    x=x+h/2*(fn+feval(f,xL,t));
end

```

5 Experimente

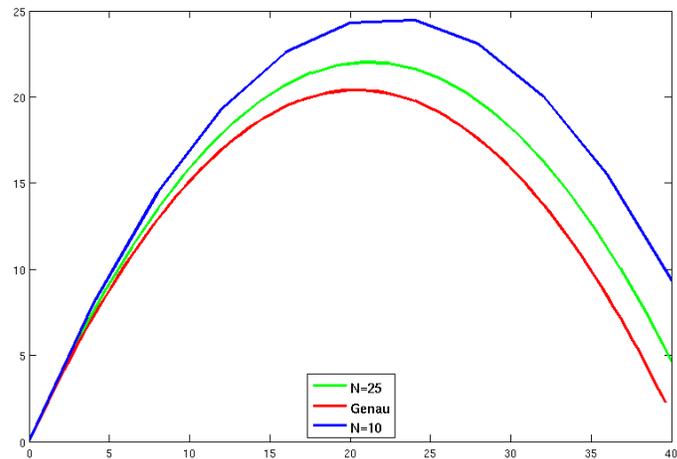
Nachdem wir alles implementiert haben, fangen wir nun an Flugkurven zu zeichnen. Gerade mit Matlab lässt sich dies leicht verwirklichen. Alle berechneten Punkte werden nacheinander gespeichert und dann gezeichnet. Nachdem wir uns in die Materie des Plot-Befehls eingearbeitet haben, konnten wir mit den geeigneten Experimenten beginnen.

5.1 Vergleich der Näherungsverfahren

Zuerst betrachten wir den Fall des Schusses ohne Luftreibung, um die Approximationsverfahren auf ihre Genauigkeit zu untersuchen.

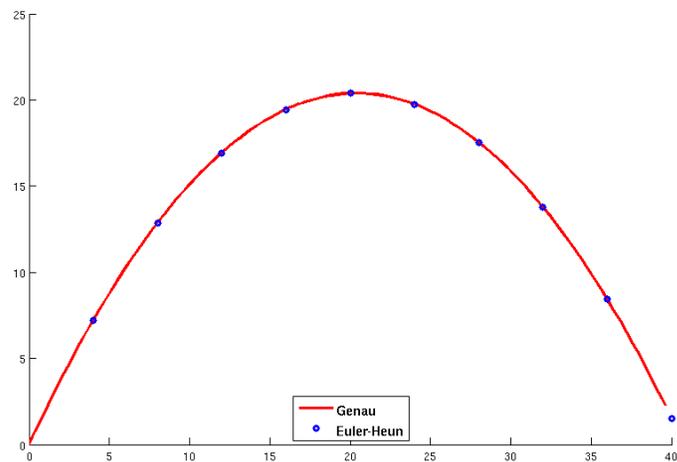
5.1.1 Euler-Verfahren

In dieser Grafik werden die genaue Kurve, sowie die Annäherungen mit 10 und 25 Schritten dargestellt.



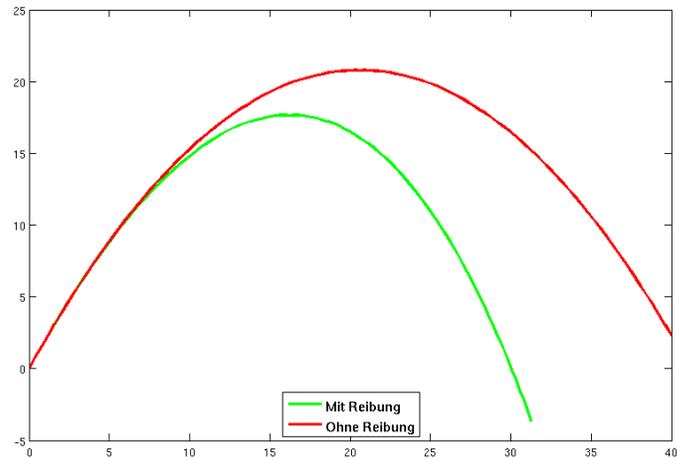
5.1.2 Euler-Heun-Verfahren

Jetzt vergleichen wir das Euler-Heun Approximationsverfahren mit der genauen Kurve. Nach unseren Berechnungen sollte das Verfahren mit den exakten Werten übereinstimmen.



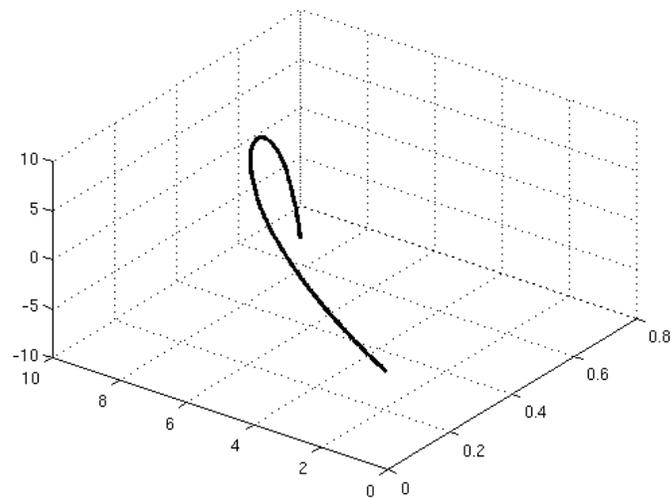
5.2 Flug mit Luftreibung

Hier berechnen wir die Flugkurve unter Berücksichtigung der Luftreibung im Vergleich zum reibungsfreien Schuss.

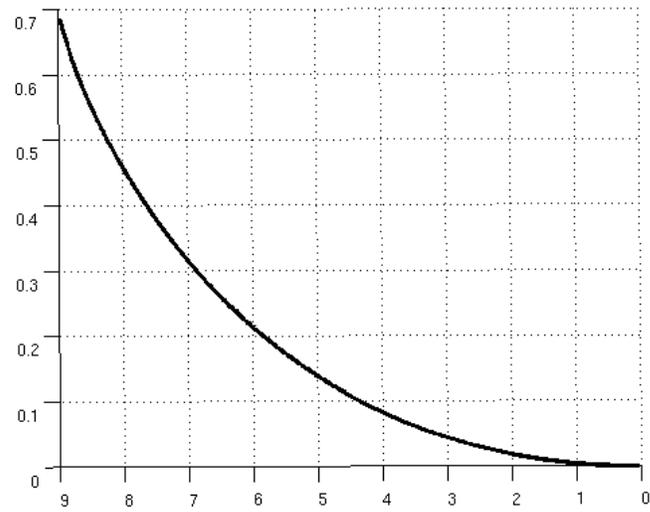


5.3 Flug mit Magnuseffekt

Zuletzt betrachten wir den Flug unter Berücksichtigung des Magnuseffektes. Da diese Kraft zur Seite wirkt, wird die Kurve dreidimensional dargestellt.



Jetzt den Effekt von oben betrachtet:



Hier nochmal alle Flugbahnen im Vergleich:

