PH Schwäbisch Gmünd
University of Education

Prof. Dr. Helmut Albrecht:

Computer als Arbeitsmittel in den MINT-Fächern

The purpose of computing is insight, not numbers.

R. W. Hamming

Institut für Mathematik und Informatik

Inhalt

0 Unvermeidliche Vorbemerkungen ... 5

1 Tabellenkalkulation ... 9
1.1 Notentabelle ... 10
1.1.1 Mathematische Grundlagen ... 10
1.1.2 Verweise in EXCEL ... 10
1.1.3 Formeleingabe ... 11
1.1.4 Absolute und relative Bezüge ... 11
1.1.5 Kopieren von Zellen ... 11
1.1.6 Wertetabellen ... 12
1.2 Euklidischer Algorithmus ... 14
1.2.1 Mathematische Grundlagen ... 14
1.2.2 Realisation in EXCEL ... 14
1.2.3 Bedingte Ausführung ... 15
1.3 Galtonbrett ... 17
1.3.1 Mathematische Grundlagen ... 17
1.3.2 Simulation in EXCEL ... 19
1.3.3 Zurücksetzen des Modells ... 20
1.3.4 Graphische Darstellung ... 21
1.4 Optimieren mit EXCEL ... 23
1.4.1 Mathematische Grundlagen ... 23
1.4.2 Realisation in EXCEL ... 25
1.5 Weitere Anwendungen in EXCEL ... 30
1.5.1 Erstellen von Wahrheitswertetafeln ... 30
1.5.2 Mathematisches Golfspiel ... 31
1.5.3 Diophantische Gleichungen ... 32

2 Dynamische Geometriesysteme ... 45
2.1 Besondere Punkte im Dreieck ... 46
2.1.1 Konstruktion der besonderen Punkte ... 46
2.1.2 Mögliche Orte der besonderen Punkte ... 46
2.1.3 Die Eulergerade ... 47
2.2 Der Feuerbach- oder Neun-Punkte-Kreis ... 48
2.2.1 Konstruktion des Kreises ... 48
2.2.2 Eigenschaften des Kreises ... 49
2.3 Darstellung von Funktionsgraphen ... 51
2.3.1 Cinderella als Funktionsplotter ... 51
2.3.2 Newton-Verfahren ... 52
2.3.3 Kurvendiskussion mit Cinderella ... 55
2.3.4 Affine Transformationen ... 58
2.4 Abbildungen ... 60
2.4.1 Kongruenzabbildungen ... 60
2.4.2 Ähnlichkeitsabbildung ... 62
2.4.3 Konforme Abbildungen ... 62
2.5 Weitere Anwendungen in Cinderella ... 65
2.5.1 Makros in Cinderella ... 65
2.5.2 Maße feststellen ... 66

3 Computer-Algebra-Systeme ... 68
3.1 IBAN-Bestimmung und Überprüfung ... 69
3.1.1 Prüfverfahren ... 69
3.1.2 Ermitteln der Prüfzahl ... 69
3.1.3 Sicherheit des Prüfzahlverfahrens ... 69
3.1.4 IBAN aus BLZ und Kontonummer erzeugen ... 70

3.1.5 Funktion für IBAN-Erstellung ... *71*
3.1.6 Funktion zur Überprüfung einer IBAN ... *72*
3.2 Der Einkommensteuertarif ... 74
3.3 Newton-Verfahren ... 77
3.3.1 Durchführung im Direktmodus ... *77*
3.3.2 Erstellen einer eigenen Funktion ... *77*
3.4 RSA-Verschlüsselung ... 79
3.4.1 Öffentlicher Schlüssel ... *79*
3.4.2 Verschlüsseln der Nachricht ... *80*
3.4.3 Privater Schlüssel ... *81*
3.4.4 Entschlüsseln der Nachricht ... *82*
3.4.5 RSA-Verschlüsselung mit Maxima ... *82*
3.4.6 Konkrete Durchführung in Maxima ... *84*
3.5 Die Methode der kleinsten Quadrate ... 87
3.5.1 Minimierung von Fehlern ... *87*
3.5.2 Realisation in Cinderella ... *88*
3.5.3 Proportionalität ... *88*
3.5.4 Linearität ... *90*
3.5.5 Bestimmung mit Maxima ... *94*

Literatur:

- Wilhelm Haager: Computeralgebra mit Maxima. Grundlagen der Anwendung und Programmierung. Carl Hanser Verlag München, 2014. ISBN 978-3-446-44203-0. Als eBook: 978-3-446-43730-2
- Volker Hole: Mathe mit EXCEL. Data Becker, 1994
- Ulrich H. Kortenkamp: The Cinderella.2 Manual: Working with the Interactive Geometry Software. Springer, Berlin und Heidelberg, 2012.
- und viele mehr ...

Studienorganisatorische Hinweise:

Die Veranstaltung *Computer als Arbeitsmittel in den MINT-Fächern* ist in der PO 2015 im Modul *Grundlagen und Schlüsselqualifikationen II* und in der *Medienbildung (Modul 2)* des Bachelorstudiums mit einem Umfang von 2 cp verortet.

Die Veranstaltung wird als Kombination zwischen der Form des *inverted classrooms* und einer *Übung* angeboten. Mit Hilfe des Skripts und der angegebenen Literatur sollen Sie sich vorbereitend eigenständig in die jeweilige anstehende Thematik, insbesondere in die den einzelnen Themen zugrundeliegenden mathematischen Grundlagen einarbeiten. In der Präsenzzeit können diese mathematischen Grundlagen diskutiert und erläutert werden, außerdem sollten Sie in dieser Zeit das jeweils anstehende Problem in einem der behandelten Softwarepakete realisieren. Hierfür sind im Skript jeweils Anleitungen enthalten, außerdem steht der Lehrende bei Problemen und für Fragen zur Verfügung.

Als Hilfsmittel werden wir die Tabellenkalkulation EXCEL, das Computeralgebrasystem *Maxima* sowie die dynamische Geometriesoftware *Cinderella* verwenden. Es ist sinnvoll, wenn Sie die Veranstaltung mit ihrem eigenen Computer durchführen. In diesem Fall müssen Sie diese Systeme natürlich auf Ihrem Rechner installiert haben. Anmerkungen hierzu erhalten Sie in den Vorbemerkungen weiter unten.

Noch ein Wort zum Arbeitsaufwand, den Sie mit dieser Veranstaltung haben werden, bzw. der von Ihnen erwartet wird: Die Veranstaltung *Computer als Arbeitsmittel in den MINT-Fächern* ist im Vorlesungsverzeichnis mit 2 ECTS-Punkten versehen. Jeder ECTS-Punkt steht für einen „workload“ von 30 Zeitstunden, dies ergibt in der Summe 60 Stunden, welche Sie für den Besuch der Veranstaltung sowie Ihre Vor- und Nachbereitungen aufwenden sollen. Wenn Sie von dieser Summe die 14 Vorlesungswochen mit der insgesamt 1,5-stündigen Veranstaltung (=21 Stunden) in Abzug bringen, so verbeiben noch 39 Stunden, welche für Ihre Vor- und Nachbereitungen vorgesehen sind.

Teilen Sie diese Zeitspanne über die Vorlesungszeit auf, so bleiben

$$39 : 14 \approx \textbf{2,8 Stunden}$$ an wöchentlicher Vor- und Nachbereitung

für diese Veranstaltung.

Denken Sie an diese Zahl, wenn Ihnen der Arbeitsaufwand für diese Veranstaltung hoch erscheint. Und bedenken Sie bitte, dass die genannten Festlegungen mit den ECTS-Punkten nicht von uns stammen, sondern letztlich europaweit gelten.

0 Unvermeidliche Vorbemerkungen

Mit Beiträgen um den Computereinsatz in der Mathematik im Allgemeinen und im Mathematikunterricht im Besonderen können ganze Regale gefüllt werden. Mit diesen Vorbemerkungen soll diese Diskussion keinesfalls erweitert werden, die sinnvolle Nutzung des Computers bzw. entsprechender Programme steht (jedenfalls für mich) längst außer Frage. Hier soll lediglich dargestellt werden, wie es (neben vielen anderen mindestens ebenso guten Ideen) zu einer möglichen vernünftigen Einbindung in die Ausbildung zukünftiger Mathematiklehrerinnen und -lehrer kommen kann.

Vorausgeschickt zum besseren Verständnis doch noch meine Argumentation für die Einbindung des Computers: Es ist nun wirklich jedem klar, dass die Herausbildung menschlicher Fähigkeiten und Fertigkeiten der Übung bedarf. Niemand wird ernsthaft in Abrede stellen, dass jemand – um ein guter Musiker zu werden – täglich fleißig üben muss. Bei einem Profi bedingt schon der Beruf, dass er täglich zum Instrument greift. Möchte jemand im Sport Erfolge erzielen, dann wird er oder sie gut daran tun, mehrmals die Woche oder gar täglich zu trainieren. Schwimmen lernt man eben nur im Wasser und nicht vom bloßen Zuschauen.

Was nun für das Erlernen eines Instruments oder einer Sportart gilt, lässt sich nicht nur auf alle anderen zu erlernenden Fähigkeiten – einschließlich der Mathematik! – übertragen, vielmehr ist eine stetige, übende Beschäftigung mit dem Lerngegenstand eine unerlässliche Notwendigkeit auf dem Weg zum Meister, von denen bekanntlich noch keiner vom Himmel gefallen ist. Warum nun der größte Teil der Menschheit glaubt, gerade beim Erlernen von Mathematik von dieser ehernen Regel abweichen zu können, wird mir immer ein Rätsel bleiben! Besonders rätselhaft ist dieses Phänomen bei Lehramtsstudierenden, welche neben Musik oder Sport in einem weiteren Fach Mathematik studieren. So ist diese Spezies neben den (theoretischen) Vorlesungen und Seminaren ihres Sport- oder Musikstudiums beinahe rund um die Uhr in der Sporthalle, auf dem Sportplatz und im Schwimmbad bzw. übend am Instrument anzutreffen und es ist wirklich erstaunlich, zu welcher Perfektion es dadurch viele Studierende in sportlicher oder musikalischer Hinsicht bringen. Die Mathematik hingegen glaubt man mit dem bloßen Lauschen der Worte des oder der Lehrenden im Hörsaal hinreichend verstehen zu können. Konkrete Arbeitsaufträge (Übungsaufgaben) werden in aller Regel nur widerwillig ausgeführt oder einfach ignoriert. Dass es dabei zu einem Missverhältnis der erzielbaren Leistungen in Musik oder Sport auf der einen und in der Mathematik auf der anderen Seite kommen muss, sollte eigentlich niemand verwundern.

Vieleicht liegt das ganze Missverständnis und Missverhältnis ja auch daran, dass man wohl allgemein weiß, wie man eine Sportart trainiert oder am Instrument übt, es hingegen den meisten Menschen rätselhaft ist, wie man (wirklich produktiv) Mathematik übt. Das Eine (Musik, Sport) macht man im Allgemeinen handelnd auf der enaktiven Ebene, Mathematik hingegen findet andererseits weitgehend abstrakt auf einer symbolischen Ebene statt und man fragt sich, wie man auf dieser Ebene überhaupt "üben" kann. Nun gibt es hierfür eine ganze Menge an sinnvollen Vorschlägen, von denen ich im Hinblick auf meine Intention zielgerichtet nur einen einzigen herausgreifen möchte:

Grundsätzlich richtig ist sicher, dass man die zu übende Prozedur zunächst gedanklich analysiert und durchdenkt. Dies wird ein Sportler machen, der einen Bewegungsablauf trainieren will, ebenso wie ein Musiker, der eine schwierige Stelle in einem Musikstück üben will und insofern unterscheidet sich ein Üben am Turngerät oder am Instrument überhaupt nicht vom Üben von Mathematik. Erst nach einem gedanklichen Durchdringen wird man die entsprechenden Bewegungsabläufe ausführen, in der Ausführung beobachten (oder vom Trainer oder Musiklehrer beobachten lassen) und aufgrund dieser Rückmeldungen zuerst den gedanklichen Prozess und danach den enaktiven Prozess verbessert wiederholen. Dieses Korrektiv scheint dem Mathematik-Übenden zu fehlen, da er ja den gedanklich vollzogenen Prozess in aller Regel nicht in Bewegungen umsetzen kann. Als Korrektiv muss hier ein Partner dienen, mit dem die

gefundenen Überlegungen diskutiert werden können. Dies kann der Lehrende oder Tutor sein, dies kann ein Lernpartner oder eine Lerngruppe sein. Doch nicht immer steht ein geduldiger Mensch zur Verfügung, mit dem man seine Gedanken über die dazu benötigte Zeitspanne ordnen, strukturieren und weiterentwickeln kann.

Es gilt weiterhin, dass man einen Stoff nur dann wirklich verstanden hat, wenn man diesen lehren – also Anderen beibringen – kann. Auch hierfür sind nun wieder Personen nötig, die zudem noch kompetent sein müssen, um entscheiden zu können, ob der "vermittelte Stoff" richtig widergegeben wird.

Fehlen solche menschlichen Trainer, Lehrer, Partner, ..., dann bietet sich der PC mit entsprechenden Softwaresystemen als "Sparringspartner" an: Wer glaubt, dass er den euklidischen Algorithmus zur ggT-Bestimmung zweier Zahlen verstanden hat, der bilde diesen doch einfach einmal in einem Tabellenkalkulationssystem ab. Wer dies schafft, kann mit Fug und Recht behaupten, dass er den Algorithmus verstanden hat. Und für denjenigen, der ihn noch nicht so richtig kapiert hat, ist der PC ein geduldiger, aber auch unerbittlicher Sparringspartner: Man kann seine Ideen korrigieren, weiterentwickeln, alternative Wege versuchen, ... Der Computer nimmt dies alles klaglos zu jeder Tages- und Nachtzeit hin und belohnt die ganze Mühe am Ende mit einem funktionierenden "Programm". Dabei ist er ein unerbittlicher "Lehrer": Erst wenn er alle Anweisungen lückenlos "verstanden" hat, wird er das Gewünschte auch wirklich ausführen. Das Glücksgefühl ist wirklich unbeschreiblich, wenn nach vielen Versuchen, Fehlschlägen und Irrwegen das selbst erstellte "Programm" endlich funktioniert!

Neben dieser praktischen und motivierenden "Übemöglichkeit", die der PC dem Lernenden bietet, ist als weiteres Argument für die Nutzung des Computers die Beschäftigung mit mathematischen Werkzeugen zu nennen, die heute untrennbar mit einer professionellen Auseinandersetzung mit der Mathematik einhergehen. Jeder Mathematiker bedient sich entsprechender Softwaresysteme, zu denen beispielsweise Computer-Algebra-Systeme gehören. In den Schulen haben längst dynamische Geometriesysteme Fuß gefasst und ein Lehramtsstudent tut gut daran, neben der Tabellenkalkulation auch mindestens noch diese beiden Werkzeuge frühzeitig kennen zu lernen, so dass sie ihm nicht nur für den späteren Einsatz in der Schule, sondern schon für die eigene Ausbildung während des Studiums gute Dienste leisten können.

Nicht zuletzt ist die Verfügbarkeit eines mathematischen Werkzeugs, mit dem relativ schnell und mit nur geringem Aufwand viele Beispiele erprobt, durchgerechnet und dargestellt werden können, für das eigene Verstehen von Mathematik hilfreich. Die Darstellung der Potenzmenge der Menge {a, b, c} mag ja noch schnell von Hand gelingen, wählt man aber als Ausgangsmenge das gesamte Alphabet, dann wird die Notation der Potenzmenge deutlich aufwändiger. Für das Erkennen geometrischer Zusammenhänge sind häufig mehrere leicht abgewandelte Konstruktionen notwendig, dynamische Geometriesysteme liefern diese quasi „frei Haus“. Aber nur, wer bereits zu Beginn die vielfältigen unterstützenden Eigenschaften entsprechender PC-Programme kennen gelernt hat, kann diese während seines Studiums gewinnbringend einsetzen!

Die konkrete Auswahl entsprechender Vertreter ist natürlich subjektiv. Über allen weiteren Kriterien schwebt jedoch – insbesondere in der Ausbildung von Studierenden – der Preis dieser Softwaresysteme. Schielt man mit einem Auge noch auf den Funktionsumfang und die Leistungsfähigkeit der auf dem Markt befindlichen Produkte, so gibt es im Bereich von Computer-Algebra-Systemen eigentlich nur einen Vertreter, der für Ausbildungszwecke in Frage kommt, dies ist das Open-Source-Projekt "Maxima".

Die Wahl im Bereich der dynamischen Geometriesoftware fiel auf "Cinderella". Dies zum einen objektiv aufgrund des dort vorhandenen enormen Funktionsumfangs und der kostenlosen Nutzungsmöglichkeit durch Studierende unserer Hochschule sowie subjektiv aufgrund einer nicht nur in Anwendungsfragen ungemein hilfreichen freundschaftlichen Verbundenheit zu Uli

Kortenkamp, seit er vor vielen Jahren einige Zeit als Kollege bei uns an der PH in Schwäbisch Gmünd war.

Beide Pakete haben noch einen weiteren, nicht zu unterschätzenden Vorteil: Sie sind auf mehreren Betriebssystemen (Windows, Mac OS, Linux) lauffähig und die Installation verläuft in aller Regel völlig problemlos. Am besten beziehen Sie die für Ihr Betriebssystem passende Version von Maxima über die Seite

http://maxima.sourceforge.net/

und installieren diese auf Ihrem privaten Rechner. Die Installation verläuft auf Windowsrechnern meist problemlos, auf dem Mac wird eine Installation aufgrund der restriktiven Sicherheitspolitik von apple leider immer aufwändiger. Falls Sie mit einem mac arbeiten, finden Sie auf meiner Seite

https://www.albrecht57.de/index.php/mathematik/computereinsatz/maxima/installation-auf-dem-mac

detaillierte Installationshinweisungen.

Zur Handhabung von Maxima gibt es viele hilfreiche Seiten im Netz, die Verweise hierzu finden Sie unter

http://maxima.sourceforge.net/documentation.html

z.B. (ohne Anspruch auf Vollständigkeit):

http://def.fe.up.pt/dynamics/maxima_tutorial.html

Nach der Installation sollten Sie zuerst die nachfolgend genannte Datei herunterladen, in Maxima öffnen und die enthaltenen Anweisungen Schritt für Schritt abarbeiten. Es ist meine Übersetzung des Tutorials: 10 minute (wx)Maxima tutorial: a quick introduction to wxMaxima and Maxima (by Žiga Lenarčič). Sie müssen diesen Link mit der **rechten** Maustaste anklicken, aus dem Kontext-Menü die Option "Datei herunterladen" wählen und diese auf Ihrem Rechner speichern – die genaue Vorgehensweise hängt von Ihrem Browser ab!

Es gibt schließlich noch mein (wohl immer unvollständig bleibendes) Maxima-Handbuch, das sich zum Einarbeiten wie zum Nachschlagen eignet. Wesentlich fleißiger war der Kollege Wilhelm Haager von der HTL St. Pölten, der mit seinem Buch "Computeralgebra mit Maxima" seit Ende 2014 eine wirkliche Lücke geschlossen und ein Standardwerk über Maxima vorgelegt hat.

Cinderella beziehen Sie in der benötigten Betriebssystemversion über die Seite

http://beta.cinderella.de/public/

Sollten Sie auch noch die Lizenzdatei benötigen, so erhalten Sie diese über die Software-Seite des Instituts. Es steht ebenfalls eine kleine, aber sicher nicht vollständige Cinderella-Anleitung von mir für Ihre ersten Schritte in Cinderella zur Verfügung.

Schließlich und endlich wird davon ausgegangen, dass Sie auf Ihrem Rechner ein Tabellenkalkulationssystem zur Verfügung haben.

Da Sie mit einem Tabellenkalkulationssystem bereits in der Schule in Berührung gekommen sein dürften, werden wir im Rahmen dieser Veranstaltung mit diesem Werkzeug beginnen. Relativ neu dürfte für Sie hingegen der Einsatz dynamischer Geometriesysteme sein. Die Beschäftigung mit einem Computer-Algebra-System wird schließlich diese Veranstaltung abrunden.

Der grundsätzliche Umgang mit diesen drei Programmpaketen (EXCEL, Cinderella, Maxima) wird in dieser Veranstaltung jeweils in einer Veranstaltung an einer ersten Aufgabenstellung vom Dozenten erläutert. Weitere Probleme sollen Sie dann anhand von Anleitungen selbst bearbeiten, wobei hierbei der Dozent für Rückfragen zur Verfügung steht. Schließlich bearbeiten Sie das Problem der kleinsten Quadrate in allen drei Programmpaketen. Alle diese Probleme sind natürlich mathematischer Natur, deren fachliche Grundlagen in den Übungen abwechselnd von den Teilnehmern bzw. Teilnehmergruppen dargestellt werden. Vorbereitend zur wöchent-

lichen Veranstaltung sollten Sie genau diese fachlichen Grundlagen durcharbeiten, um diese dann mit Hilfe der Programmpakete einer Lösung zuführen zu können.

Während der Übungszeit ist es nicht nur erlaubt sondern sogar beabsichtigt, dass Sie die anstehenden Aufgaben gemeinsam lösen! Fragen Sie (zunächst) die Umsitzenden, wenn Sie ein Problem haben. Helfen Sie Ihren Kommilitoninnen und Kommilitonen, wenn Sie sehen, dass diese nicht weiterkommen. Für hartnäckige Probleme steht Ihnen selbstverständlich der Veranstaltungsleiter zur Verfügung.

Um eine erfolgreiche Teilnahme bescheinigt zu bekommen, bringen Sie in der letzten Veranstaltung des Semesters eine einfache Problemstellung allein und ohne fremde Hilfe zur Lösung.

Außer im Angebot des Fachs Mathematik wird diese Veranstaltung auch noch unter den Grundlagen des Studiums II sowie in der Medienbildung (Modul 2) ausgebracht. Um für einen dieser Bereiche eine erfolgreiche Teilnahme bescheinigt zu bekommen, werden Sie in der letzten Veranstaltung des Semesters eine einfache Problemstellung allein und ohne fremde Hilfe zur Lösung bringen.

1 Tabellenkalkulation

Die Tabellenkalkulation dürfte das älteste, für durchschnittliche Anwender bedienbare Rechen-Hilfsprogramm auf dem Computer sein. Heute gehört es zur Standartausstattung eines jeden Laptops, auch sehr viele (schul-)mathematische Themen lassen sich damit sinnvoll erarbeiten.

Die in dieser Veranstaltung abgehandelten Probleme und Beispiele werden anhand von EXCEL, dem Tabellenkalkulationssystem aus dem Office-Paket von Microsoft erarbeitet. Falls Sie privat mit der Tabellenkalkulation *Calc* des Open-Office-Pakets arbeiten, so ist dies in aller Regel kein Problem, da dieses System sehr eng an EXCEL angelehnt ist. Eine deutlich andere Philosophie verfolgt *Numbers*, die Tabellenkalkulation von apple. Zwar können Sie auch mit *Numbers* die hier behandelten Beispiele und die dabei zum Zuge kommenden Verfahren prinzipiell nachvollziehen, allerdings gibt es in *Numbers* deutliche Einschränkungen, da beispielsweise keine manuelle Berechnung initiiert werden kann und auch eine iterative Berechnung nicht möglich ist. Ich selbst arbeite weder mit *Calc*, noch mit *Numbers*, so dass ich Ihnen bei auftretenden Problemen mit diesen Programmen nur sehr rudimentär weiterhelfen kann.

Auf jeden Fall sollten Sie auf Ihrem Rechner eine Tabellenkalkulation zur Verfügung haben und mit deren Umgang vertraut sein, dies sieht zumindest der Lehrplan der Gymnasien vor. Die grundlegende Arbeitsweise sollte Ihnen somit geläufig sein. Im ersten, einführenden Beispiel werden diese Grundideen wohl nochmals dargestellt und gemeinsam erarbeitet, die nachfolgenden Probleme werden allerdings dann deutlich anspruchsvoller.

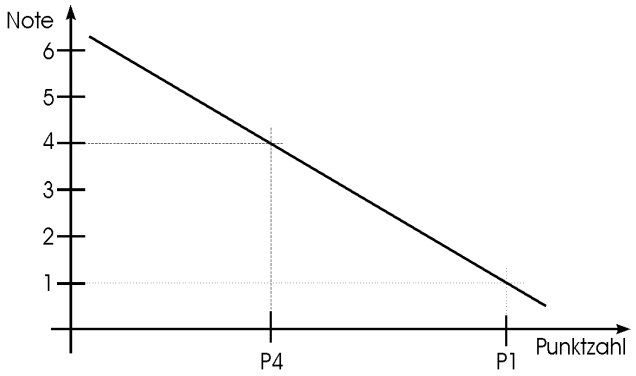
1.1 Notentabelle

1.1.1 Mathematische Grundlagen

Für jede Klassenarbeit muss die von den einzelnen Schülern erreichte Punktezahl in eine Note umgerechnet werden. Eine Möglichkeit besteht in einer linearen Zuordnung zwischen Punkten und Noten gemäß nebenstehender Skizze. Vom Lehrer wird festgelegt, mit welcher Punktezahl P_1 die Note 1 und mit welcher Punktezahl P_4 die Note 4 gegeben werden soll. Es ist somit die Geradengleichung $y = ax + b$ gesucht mit den beiden Fixpunkten $(P_4, 4)$ und $(P_1, 1)$.

Die Steigung der Geraden wird über das Steigungsdreieck ermittelt:

$$m = \frac{\Delta y}{\Delta x} = \frac{4-1}{P_4 - P_1} = \frac{3}{P_4 - P_1}$$



Aus der Steigung und einem zugeordneten Punktepaar – beispielsweise $(P_1, 1)$ – kann dann über die Punkt-Steigungs-Form die Standardform der Geradengleichung ermittelt werden:

$$m = \frac{y-1}{x-P_1} \quad \Leftrightarrow \quad y - 1 = mx - mP_1 \quad \Leftrightarrow \quad y = mx + 1 - mP_1$$

Ersetzt man in der resultierenden Gleichung schließlich die Steigung m durch den oben gefundenen Bruch für m, so erhalten wir:

$$y = \frac{3}{P_4 - P_1} \cdot x + \left(1 - \frac{3 \cdot P_1}{P_4 - P_1}\right)$$

1.1.2 Verweise in EXCEL

	A	B	C	D	E	F
1	Notentabelle					
2						
3	Punkte(max):	60		Punkte		Note
4	Punkte(1):	60		60	1,00	1
5	Punkte(4):	30		59	1,10	1
6				58	1,20	1-
7				57	1,30	1-
8				56	1,40	1-2
9				55	1,50	1-2
10				54	1,60	1-2
11				53	1,70	2+
12				52	1,80	2+

Aus den von der Lehrerin oder dem Lehrer festzulegenden Parametern P_1 (Punktezahl, welche für die Note 1 erreicht werden muss) und P_4 (Punktezahl, welche für die Note 4 erreicht werden muss) lassen sich mit Hilfe dieser Gleichung die zu jeder Punktzahl x gehörige Note y errechnen. Diese Berechnungen lassen wir vorteilhaft EXCEL durchführen und uns eine Notentabelle nach dem nebenstehenden Vorbild erstellen.

In die Zellen `B3` bis `B5` werden die relevanten Parameter eingetragen und daraus in den Spalten `D`, `E` und `F` die eigentliche Notentabelle erstellt.

Der Startwert der Punktezahl in `D4` wird durch einen *Verweis* aus `B3` übernommen. Solche Verweise oder Bezüge sind grundlegend für die Arbeit in einer Tabellenkalkulation, sie besagen, dass man nicht den Wert (hier 60) in die Zelle `D4` schreibt, sondern stattdessen mit dem Verweis `=B3` EXCEL anweist, den in der Zelle `B3` befindlichen Wert zu übernehmen. Ändert man später den Wert in `B3` so ändert sich damit auch automatisch der Wert in `D4`. Solche Verweise kann man nach getipptem Gleichheitszeichen von Hand eintippen, besser jedoch klickt man mit der Maus auf diejenige Zelle, deren Wert übernommen werden soll.

1.1.3 Formeleingabe

In der Zelle `E4` soll die zugehörige Note nach unserer oben hergeleiteten Formel errechnet werden. Man gibt daher in `E4` nach einem einleitenden Gleichheitszeichen die Formel ein, wobei man statt der Zahlenwerte wiederum Verweise auf diejenigen Zellen einfügt, welche diese Zahlenwerte enthalten. Die Berechnungsvorschrift (die sogenannte *Formel*) in `E4` lautet dann nach unserer oben hergeleiteten Geradengleichung:

```
=3/($B$5-$B$4)*D4+(1-3*$B$4/($B$5-$B$4))
```

Am besten gibt man nur die Rechenzeichen und Klammern über die Tastatur ein und erzeugt die Verweise durch Klicken mit der Maus.

1.1.4 Absolute und relative Bezüge

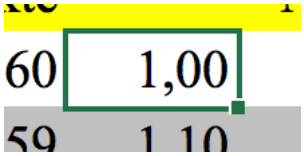
Es fällt auf, dass außer `D4` alle anderen Verweise ein $-Zeichen vor dem Buchstaben als auch vor der Zahl haben. Hierbei handelt es sich um *feste oder absolute Bezüge*, während `D4` einen *relativen Bezug* darstellt. Hier ist nun eine kurze Klärung nötig, was denn relative und was absolute Bezüge sind:

Wenn Sie sich nochmals genau die Formel zur Notenberechnung anschauen so wird deutlich, dass dafür drei Größen notwendig sind: Die erreichten Punkte, die Punktezahl für die Note 1 und die Punktezahl für die Note 4. Aus dem angedeuteten Aufbau der Tabelle geht hervor, dass die erreichte Punktezahl immer links von der errechneten Note steht. Es ist außerdem naheliegend, dass wir die umfangreiche Formel zur Notenberechnung nicht 60-mal eingeben wollen, sondern uns durch Kopieren das Leben einfacher machen werden. Bei solchen Kopieraktionen betrachtet EXCEL Bezüge in den Formeln in der Form `D4` als relativ. Dies bedeutet, wenn die Berechnungsformel in `E4` eine Zeile nach unten in `E5` kopiert wird, in dieser Formel aus dem Bezug `D4` automatisch den Bezug `D5` macht. In diesem Fall – in der Spalte `D` stehen die zugeordneten Punktezahlen – ist das auch völlig in Ordnung und hilfreich.

Für die Berechnung der Note sind aber auch die Punktezahlen für die Note 1 und die Note 4 nötig und diese Werte stehen in den Zellen `B4` und `B5`. Wenn wir nun diese Bezüge in dieser relativen Form in die Formel einfügen würden, dann würde daraus nach dem Kopieren eine Zeile nach unten `B5` und `B6` werden, was natürlich nicht korrekt und zielführend wäre. Deshalb muss mitgeteilt werden, dass eben diese Bezüge `B4` und `B5` nicht relativ sind, sondern fest bleiben müssen. Am einfachsten geschieht dies, wenn Sie sofort nach dem Erzeugen eines solchen Bezugs durch Klicken mit der Maus so lange die Taste F4 (Mac: cmd t) drücken, bis beispielsweise aus dem relativen Bezug `B4` der absolute Bezug `$B$4` geworden ist.

Bei jedem Drücken von F4 ändert sich der dargestellte Bezug, so dass man aus einem absoluten auch wieder einen relativen Bezug machen kann. Außerdem gibt es Zwischenstufen, wo nur der Zeilen- bzw. Spaltenbezug absolut ist. In unserem Fall würde ein absoluter Zeilenbezug (bspw. `B$4`) ausreichen, es schadet aber auch nicht, den gesamten Bezug absolut zu setzen.

1.1.5 Kopieren von Zellen



Die Inhalte der Spalte `F` stellen wir kurz zurück und erstellen die zweite Zeile unserer Notentabelle: Wenn wir eine absteigende Folge der Punktzahlen darstellen wollen, dann muss in `D5` nun die Punktzahl 59 erscheinen. Diese lassen wir über die Formel `=D4-1` errechnen. Die Formel in `E4` können wir schon nach unten kopieren, am einfachsten geschieht dies in EXCEL indem man die Zelle durch Anklicken auswählt und an dem rechteckigen Punkt rechts unten in der Zellumrandung nach unten zieht.

Damit ist die zweite Zeile der Tabelle erstellt und wir können beide Zellen `D4` und `E4` markieren und dann beide Zellen wie eben dargestellt durch Ziehen an dem rechteckigen Punkt belie-

big weit nach unten kopieren. Damit haben wir mit recht wenig Aufwand eine lineare Zuordnung von Punkten zu Noten entsprechend der Vorgaben (Punkte für die Noten 1 und 4) erhalten. Unter Umständen ist es nötig, die Zahlenangaben in der Spalte `E` einheitlich auf zwei Nachkommastellen zu reduzieren.

1.1.6 Wertetabellen

In Spalte `F` sind den Dezimalnoten die in der Schule gebräuchlichen Notenstufen 1, 1–, 1–2, 2+ usw. zugeordnet. Um dies realisieren zu können, benötigen wir eine kleine Tabelle, welche genau diese Zuordnung enthält.

Beim Öffnen einer neuen EXCEL-Arbeitsmappe erhalten Sie (je nach Version) ein bis mehrere leere Tabellenblätter, die durch Reiter am unteren Bildschirmrand ausgewählt werden können. Weitere Tabellenblätter können Sie problemlos durch Klicken auf das "+"-Zeichen erzeugen.

Die Tabellenblätter sind von Haus aus mit *Tabelle1*, *Tabelle2* usw. beschriftet. Durch Doppelklicken auf den Reiter können Sie das Blatt neu beschriften, nennen Sie Ihr aktuelles Blatt *Notentabelle*. Erzeugen Sie nötigenfalls ein zweites Blatt und nennen Sie dieses *Notenstufen*. Wechseln Sie auf dieses Blatt und legen Sie dort links oben beginnend die nebenstehende Zuordnung der Notenstufen zu den Dezimalnoten an. Zuvor müssen Sie jedoch die gesamte Spalte `B` durch Klicken auf den Spaltenkopf markieren und als Zellenformat (rechte Maustaste) die Option *Text* wählen.

	A	B
1	1	1
2	1,25	1-
3	1,5	1-2
4	1,75	2+
5	2	2

Wechseln Sie dann zurück auf das Tabellenblatt *Notentabelle*. Um dort in Spalte `F` die entsprechenden Schulnoten zuordnen zu können, greifen wir mit Hilfe der Funktion `VERWEIS()` auf die eben erstellte Zuordnung zu. In der Hilfe von EXCEL finden Sie die folgende Erläuterung (mit eingefügten Anmerkungen):

Durchsucht die erste Spalte einer Matrix und durchläuft die Zeile nach rechts, um den Wert einer Zelle zurückzugeben.

Syntax

SVERWEIS(**Suchkriterium;Matrix;Spaltenindex;Bereich_Verweis**)

- **Suchkriterium**: ist der Wert, nach dem Sie in der ersten Spalte der Matrix suchen.

(in unserem Fall die Dezimalnoten aus der Spalte E)

- **Matrix**: ist die Informationstabelle, in der Daten gesucht werden.

(in unserem Fall die Zuordnungstabelle auf dem Blatt Notenstufen)

- **Spaltenindex**: ist die Nummer der Spalte in der Mehrfachoperationsmatrix, aus der der übereinstimmende Wert zurückgegeben werden soll.

(in unserem Fall stehen die Schulnoten in der zweiten Spalte)

- **Bereich_Verweis**: gibt an, ob eine genaue Übereinstimmung gefunden werden soll: WAHR = aus der aufsteigend sortierten Reihenfolge der Werte wird der Wert zurückgegeben, der am dichtesten am gesuchten Wert liegt; FALSCH = es wird eine genaue Übereinstimmung gesucht.

(in unserem Fall WAHR, kann aber auch einfach weggelassen werden)

Die Funktion `SVERWEIS()` geben Sie in die Zelle `F4` ein, beginnend mit einem Gleichheitszeichen. Den ersten Parameter, das Suchkriterium, erhalten Sie, indem Sie auf `E4`, die Dezimalnote, klicken. Den Strichpunkt als Trennzeichen geben Sie von Hand ein. Die Matrix haben Sie auf dem Tabellenblatt Notenstufen erstellt. Wechseln Sie also zu diesem Blatt und markieren dort die gesamte Tabelle durch Drüberziehen mit der Maus. Den nächsten Strichpunkt geben Sie wieder von Hand ein, ebenso den letzten Wert 2 für den Spaltenindex. Den letzten Parameter lassen wir einfach weg. Die erzeugte Formel in F4 müsste nun folgendermaßen aussehen:

```
=SVERWEIS(E4;Notenstufen!$A$1:$B$22;2)
```

Achten Sie darauf, dass der Verweis auf die Matrix absolut angegeben wird!

Wenn Sie diese Formel einige wenige Zeilen nach unten kopieren werden Sie feststellen, dass die Zuordnung nicht ganz korrekt ist. Überlegen Sie, wie Sie diesen Fehler beheben können!

Sollten Sie sich entschließen, nicht nur für die höchsterreichbare Punktzahl die Note 1 zu vergeben sondern auch für ein paar Punkte weniger, so können ebenfalls Probleme bei der Zuordnung auftauchen. Finden Sie auch hierfür eine Lösung!

1.2 Euklidischer Algorithmus

1.2.1 Mathematische Grundlagen

Der euklidische Algorithmus dient dazu, den größten gemeinsamen Teiler (ggT) zweier Zahlen zu finden. Euklid gelangte schon vor rund 2300 Jahren zu der Erkenntnis, wonach das gemeinsame Maß zweier Strecken auch ein Maß für die Differenz der Strecken sein muss. Wenden wir diese Erkenntnis auf die beiden Zahlen 9 und 15 an:

Das *gemeinsame Maß* (der gemeinsame Teiler) von 9 und 15 muss auch ein Maß (ein Teiler) der Differenz von 9 und 15, also von 6 sein. Damit gilt ebenfalls: Das gemeinsame Maß von 9 und 6 muss auch ein Maß von 9 – 6 = 3 sein. Und weiter: Das gemeinsame Maß von 6 und 3 muss ein Maß von 6 – 3 = 3 sein. Damit liegt das „gemeinsame Maß“ fest, es ist die Zahl 3.

Notieren wir diese Überlegungen formal, so können wir schreiben:

15 – 9 = 6 und 9 – 6 = 3 und 6 – 3 = 3 und 3 – 3 = 0

Durch fortwährende Differenzbildung erhalten wir schließlich den ggT zweier Zahlen. Bestimmt man den ggT anderer Zahlen, kann diese fortgesetzte Differenzbildung allerdings recht langwierig werden:

150 – 27 = 123 123 – 27 = 96 96 – 27 = 69 69 – 27 = 42

42 – 27 = 15 27 – 15 = 12 15 – 12 = 3 12 – 3 = 9

9 – 3 = 6 6 – 3 = 3 3 – 3 = 0

Hier hilft die Erkenntnis, dass die Division eine fortgesetzte Subtraktion ist: Wie oft 27 in 150 enthalten ist, kann man herausbekommen, indem man 27 so oft von 150 subtrahiert bis es „nicht mehr geht“ – oder eben, indem man 150 durch 27 dividiert. Und zwar genau so, wie es die Grundschüler machen, mit der Angabe des Rests:

150 = 5 · 27 + 15

27 = 1 · 15 + 12

15 = 1 · 12 + 3

12 = 4 · 3 + 0

Genau dies ist der *euklidische Algorithmus:* Um den ggT zweier Zahlen zu finden dividiert man die größere durch die kleinere Zahl und ermittelt den auftretenden Rest. Danach verfährt man mit der kleineren Zahl und dem erhaltenen Rest genauso weiter, bis der Rest 0 auftritt. Der letzte, von Null verschiedene Rest ist der gesuchte ggT.

Die Lösung diophantischer Gleichungen ist eng mit dem Finden des größten gemeinsamen Teilers zweier Zahlen verknüpft. Daher finden Sie zu dieser weiterführenden Thematik unter dem Kapitel *Weiterführende Anwendungen zu EXCEL* die Themen *Mathematisches Golfspiel* und *Diophantische Gleichungen*.

1.2.2 Realisation in EXCEL

Dieser Algorithmus lässt sich aufgrund seiner einfachen und klaren Struktur recht einfach in EXCEL nachbilden. Benötigt werden hierfür die Funktionen `QUOTIENT()` und `REST()`. Die erstgenannte Funktion liefert das ganzzahlige Ergebnis einer Division, die zweite Funktion liefert den bei einer Ganzzahldivision auftretenden Rest.

Probieren Sie zunächst beide Funktionen aus, um ein Gefühl für deren Arbeitsweise zu bekommen!

Erstellen Sie dann nach dem untenstehenden Muster eine siebenspaltige Tabelle, welche den euklidischen Algorithmus durchführen soll (Zahl / Gleichheitszeichen / Quotient / Malpunkt

bzw. -stern / Zahl / Pluszeichen / Rest). Um die Rechenzeichen eingeben zu können, ist es ggf. notwendig, ihnen einen einfachen Apostroph (´) voranzustellen.

	A	B	C	D	E	F	G	H	I
1		424				76			
2									
3									
4									
5		**Euklid:**							
6		424	=	5	*	76	+	44	
7		76	=	1	*	44	+	32	
8		44	=	1	*	32	+	12	
9		32	=	2	*	12	+	8	
10		12	=	1	*	8	+	4	
11		8	=	2	*	4	+	0	
12									
13									

Die beiden Zahlen, deren ggT bestimmt werden sollen, werden in `B1` und `F1` eingegeben.

Der euklidische Algorithmus soll in Zeile `6` des Tabellenblatts beginnen. In den Spalten `B` und `F` übernehmen Sie per Verweis die oben eingegebenen Zahlen.

In Spalte `D` soll der Ganzzahlquotient errechnet werden und in Spalte `H` der entstehende Rest.

Die Breiten der Spalten können Sie anpassen, indem Sie in den Spaltenköpfen (dort wo `A`, `B`, `C`, ... steht) mit der Maus die vertikalen Trennlinien verziehen. Machen Sie die Spalten `B`, `D`, `F` und `H` breit genug, da EXCEL sonst große Zahlen nicht anzeigen kann.

In die zweite Zeile des Algorithmus (Zeile `7` im Tabellenblatt) werden der Divisor aus `F6` nach `B7` und der Rest aus `H6` nach `F7` per Verweis übernommen. Der Ganzzahlquotient in `D7` wird ebenso wieder errechnet wie der Rest in `H7`, die Rechenzeichen kann man einfach aus der obigen Zeile kopieren.

Markieren Sie nun diese zweite Zeile Ihres Algorithmus und kopieren Sie diese bis einschließlich in Zeile `15` des Tabellenblatts.

Überprüfen Sie anhand verschiedener Beispiele, ob der erstellte Algorithmus einwandfrei funktioniert und zuverlässig den ggT der beiden eingegebenen Zahlen liefert.

1.2.3 Bedingte Ausführung

In aller Regel werden nicht alle vorbereiteten Zeilen für die Bestimmung des ggT benötigt, die nicht benutzten Zeilen mit den Rechenzeichen und Fehlermeldungen bleiben stehen und wirken irritierend. Dieses Manko soll beseitigt werden, indem die Ausgabe solcher nicht benötigten Zellen unterdrückt wird!

Hierfür beobachten Sie nochmals genau die Ausgabe von EXCEL bei verschiedenen ggT-Berechnungen. Welches ist jeweils die letzte sinnvoll dargestellte Zeile? Was ist in jedem Fall deren Erkennungsmerkmal?

Das Erkennungsmerkmal der letzten, sinnvoll dargestellten Zeile ist der dort auftretende Rest 0. Man kann somit – noch umgangssprachlich – die folgende Bedingung formulieren:

Wenn in der vorhergehenden Zeile der Rest 0 auftritt, **dann** soll in der darauffolgenden Zeile in Spalte B keine Ausgabe mehr erfolgen, **sonst** soll dort der vorhergehende Divisor erscheinen.

Solche wenn–dann-sonst–Aussagen werden in der Logik als Subjunktion bezeichnet, sie sind umgangssprachlich sehr geläufig und auch jede Programmiersprache enthält solche *bedingten Anweisungen*. Ihre Struktur ist immer dieselbe, die Hilfe von EXCEL führt aus:

Syntax

WENN(**Wahrheitstest;Wert_wenn_wahr;Wert_wenn_falsch**)

- **Wahrheitstest**: ist ein beliebiger Wert oder Ausdruck, der WAHR oder FALSCH sein kann.

(in unserem Fall die Überprüfung, ob der Rest in der vorhergehenden Zeile 0 ist)

- **Wert_wenn_wahr**: ist das Resultat der Funktion, wenn der Wahrheitstest WAHR ergibt.

(in unserem Fall soll dann *nichts* ausgegeben werden)

- **Wert_wenn_falsch**: ist das Resultat der Funktion, wenn der Wahrheitstest FALSCH ergibt.

(in unserem Fall soll dann der Divisor aus der vorhergehenden Zeile ausgegeben werden)

In Feld B7 wird man somit die folgende Formel eingeben:

```
=WENN(H6=0;"";F6)
```

Nichts auszugeben kann man EXCEL anweisen, indem man leere Anführungszeichen angibt. Achten Sie dabei unbedingt darauf, dass Sie wirklich *leere Anführungszeichen* (`""`) eingeben und kein Leerzeichen dazwischen einfügen (`" "`)!

In den weiteren Spalten wird man überprüfen, ob in der `B`-Spalte *nichts* steht und dann ebenso *nichts* ausgeben oder eben das jeweilige Rechenzeichen oder die Berechnung des Quotienten oder des Rests durchführen, beispielsweise in der Spalte `C`:

```
=WENN(B7="";"";"=")
```

Vervollständigen Sie die restlichen Spalten entsprechend!

Haben Sie die zweite Zeile des Algorithmus derart komplettiert, so markieren Sie die gesamte Zeile und kopieren Sie diese bis einschließlich Zeile `15` des Tabellenblatts.

In manchen EXCEL-Versionen werden Sie feststellen, dass unsere Bemühungen noch nicht ganz zielführend waren: Wohl wird die auf die letzte Zeile des Algorithmus folgende Zeile nun komplett leer dargestellt, in den nächsten Zeilen erfolgen jedoch wieder unerwünschte Ausgaben. Woran liegt das? Denken Sie scharf nach und versuchen Sie, die Ursache dafür herauszufinden! Wenn Sie die Ursache gefunden haben, dann versuchen Sie, dieses Problem zu lösen! Dabei kommt es im ersten Moment nicht auf eine Problemlösung in EXCEL an, Sie sollen vielmehr das Problem und dessen mögliche Lösung zunächst mit Ihren eigenen Worten präzise schildern!

Ist Ihnen dies gelungen, so werden Sie sicher festgestellt haben, dass zur Problemlösung eine Funktion notwendig ist, mit der ein *logisches Oder* nachgebildet werden kann:

Diese Funktion prüft, ob eines der Argumente WAHR ist und gibt dann WAHR zurück. Nur wenn alle Argumente FALSCH sind, wird FALSCH zurückgegeben.

Syntax

ODER(**Wahrheitswert1**;Wahrheitswert2;...)

- **Wahrheitswert1**: Wahrheitswert1;Wahrheitswert2;... sind 1 bis 255 Bedingungen, die Sie überprüfen möchten und die jeweils entweder WAHR oder FALSCH sein können.
- **Wahrheitswert2**: Wahrheitswert1;Wahrheitswert2;... sind 1 bis 255 Bedingungen, die Sie überprüfen möchten und die jeweils entweder WAHR oder FALSCH sein können.

Versuchen Sie, mit Hilfe dieser ODER()-Funktion den aufgetretenen Fehler zu eliminieren. Welche Formel (in welcher Spalte?) müssen Sie hierfür nur ändern?

Welcher Teil muss dort nur verändert werden? Versuchen Sie dafür, die Problemlösung möglichst exakt mit Ihren eigenen Worten zu formulieren!

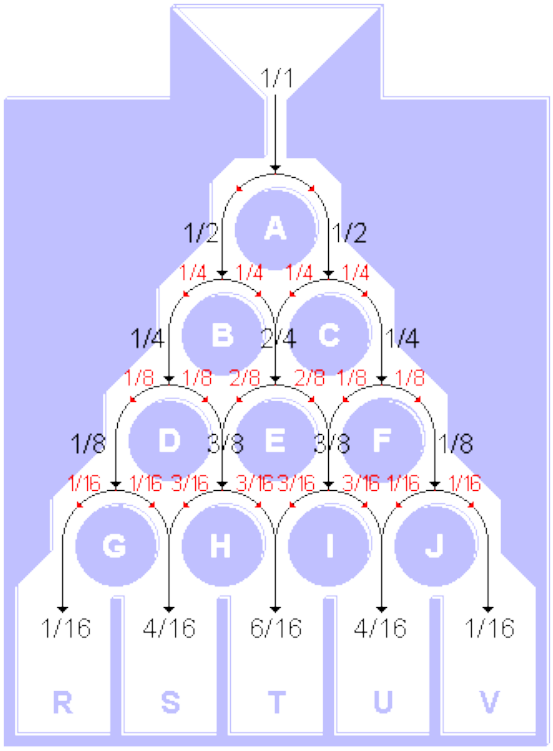
So noch Zeit bleibt, bearbeiten Sie die Aufgabenstellungen *Mathematisches Golfspiel* und *Diophantische Gleichungen* am Ende des ersten Kapitels.

1.3 Galtonbrett

Das *Galtonbrett* ist ein mechanisches Modell zur Demonstration und Veranschaulichung der Binominalverteilung und wurde nach seinem Erfinder Francis Galton (britischer Naturforscher und Schriftsteller, 1822–1911) benannt. Am Galtonbrett lässt sich die Binominalverteilung demonstrieren, dies ist eine spezielle Wahrscheinlichkeitsverteilung, die in vielen Zufallsexperimenten eine Rolle spielt. Sie finden im Internet unschwer einige Videos, in denen ein Galtonbrett in Funktion dargestellt ist.

1.3.1 Mathematische Grundlagen

Das Galtonbrett besteht aus einer regelmäßigen Anordnung von Hindernissen, an denen eine von oben eingeworfene Kugel jeweils nach links oder rechts abprallen kann. Nach dem Passieren der Hindernisse werden die Kugeln in Fächern aufgefangen, um dort gezählt zu werden. Jedes Auftreffen einer Kugel auf ein Hindernis ist ein *Bernoulli-Versuch*, bei dem es nur zwei mögliche Versuchsausgänge gibt. Das Werfen einer Münze ist beispielsweise ein Bernoulli-Versuch (Jakob I Bernoulli, 1655–1705). Beim Galtonbrett gibt es die beiden Versuchsausgänge „Kugel fällt rechts“ und „Kugel fällt links“. Wenn das Galtonbrett exakt gebaut ist und genau waagerecht steht, dann ist die Wahrscheinlichkeit, dass die Kugel nach rechts fällt genau 1/2. Die Wahrscheinlichkeit, dass die Kugel nach links fällt ist ebenfalls 1/2.



Beim Galtonbrett sind die Hindernisse in mehreren Reihen angeordnet, so dass eine Kugel nach dem Passieren eines Hindernisses auf ein Nächstes trifft. Hat das Galtonbrett beispielsweise 4 Reihen, so handelt es sich um die 4-malige Wiederholung eines Bernoulli-Versuchs, man spricht von einer Bernoulli-Kette der Länge 4.

Man kann einfach die Wahrscheinlichkeiten berechnen, mit denen die Kugel in ein bestimmtes Fach fällt: Am Hindernis *A* fällt die Hälfte der Kugeln nach rechts, die andere Hälfte fällt nach links. Damit kommt an den Hindernissen *B* und *C* jeweils die Hälfte der Kugeln an, von denen wiederum die Hälfte (der Hälfte, also ein jeweils ein Viertel) nach rechts und das andere Viertel nach links fällt. Links von *B* und rechts von *C* fallen also jeweils ein Viertel der Kugeln durch und in der Mitte zwischen *B* und *C* sind es zwei Viertel. In der nächsten Hindernisreihe werden diese Viertel wieder halbiert, also in Achtel aufgeteilt, wobei diese Achtel zwischen den Hindernissen wieder addiert werden. Man tut übrigens gut daran, die entstehenden Brüche nicht zu kürzen, da man so die Wahrscheinlichkeiten der Kugelwege besser verfolgen kann. Die sich so ergebenden Wahrscheinlichkeiten nach der letzten Aufteilung und Vereinigung an der untersten Hindernisreihe sind die Wahrscheinlichkeiten, mit denen die Kugeln in die Fächer fallen. Diese Wahrscheinlichkeiten sind auf der nachfolgenden Tabelle nochmals herausgearbeitet:

				$\frac{1}{1}$				
			$\frac{1}{2}$		$\frac{1}{2}$			
		$\frac{1}{4}$		$\frac{1}{4}+\frac{1}{4}$		$\frac{1}{4}$		
	$\frac{1}{8}$		$\frac{1}{8}+\frac{2}{8}$		$\frac{2}{8}+\frac{1}{8}$		$\frac{1}{8}$	
$\frac{1}{16}$		$\frac{1}{16}+\frac{3}{16}$		$\frac{3}{16}+\frac{3}{16}$		$\frac{3}{16}+\frac{1}{16}$		$\frac{1}{16}$

Bereits an diesem Beginn lassen sich wesentliche Strukturen herausarbeiten: Die Nenner der Brüche sind die Zweierpotenzen, die von Zeile zu Zeile fortschreiten. Die Zähler der Brüche kann man ganz einfach ermitteln, indem man die Zähler der jeweils rechts und links über der Zelle stehenden Zähler addiert.

Beschränken wir uns auf die Zähler, erhalten wir das Pascal'sche Dreieck, das ebenfalls mit ein paar Klicks in EXCEL erzeugt ist:

														1																
												1				1														
										1				2				1												
								1				3				3				1										
						1				4				6				4				1								
					1				5				10				10				5				1				
				1				6				15				20				15				6				1		
			1				7				21				35				35				21				7			
		1				8				28				56				70				56				28				8		
	1				9				36				84				126				126				84				36				9				1	
1				10				45				120				210				252				210				120				45				10				1

Die Zahlen im Pascal'schen Dreieck – und damit die Zähler unserer Wahrscheinlichkeitsbrüche – sind die Binominalkoeffizienten. Diese treten bekanntlich auf, wenn man ein Binom $(a + b)$ quadriert bzw. in höhere Potenzen versetzt.

Bezeichnet man die Zeilenzahl des Pascal'schen Dreiecks mit n und die Zeilenzahl mit k (und beginnt mit der Nummerierung jeweils bei 0) und notiert die jeweilige Stelle mit $\left(\begin{array}{c} n \\ k \end{array} \right)$ (gesprochen: n über k), dann gilt offensichtlich:

(1): $\left(\begin{array}{c} n \\ 0 \end{array} \right) = \left(\begin{array}{c} n \\ n \end{array} \right) = 1$, dies sind die Einsen an den Rändern des Dreiecks.

Da man den Wert an der Stelle $\left(\begin{array}{c} n \\ k \end{array} \right)$ aus der Summe der Werte an den beiden darüber liegenden Stellen $\left(\begin{array}{c} n-1 \\ k-1 \end{array} \right)$ und $\left(\begin{array}{c} n-1 \\ k \end{array} \right)$ bestimmen kann, gilt:

(2): $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$

Mit (1) und (2) haben wir eine rekursive Darstellung des Pascal'schen Dreiecks gefunden. Der Wert für $\binom{n}{k}$ lässt sich mit Hilfe der Beziehung

(3): $\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$

auch direkt berechnen. Unser Zahlenlotto (6 aus 49) hat übrigens $\binom{49}{6} = 13983816$ verschiedene Möglichkeiten, die sechs Zahlen anzukreuzen.

1.3.2 Simulation in EXCEL

Um die mathematischen Beziehungen besser nachvollziehen zu können, bietet es sich an, ein Galtonbrett selbst zu bauen – oder eben zu simulieren. Wir machen letzteres mit Hilfe einer Tabellenkalkulation. Bei der Erstellung der Tabelle treten *Zirkelbezüge* auf, deshalb funktioniert das Modell nur, wenn bei den BERECHNUNGSOPTIONEN die *manuelle Berechnung* eingetragen ist und der Haken bei ITERATION BESCHRÄNKEN gesetzt ist. Die MAXIMALE ITERATIONSZAHL und die MAXIMALE ÄNDERUNG muss dabei jeweils auf 1 gesetzt sein. Genau dies ist in *Numbers* m.W. nicht möglich, so dass mit diesem Programm zwar das Fallen der Kugeln simuliert werden kann, ein Aufsummieren wird aber genau so wenig möglich sein wie das Erstellen von Makros. In *Calc* gibt es das Problem, dass die fallenden Kugeln nicht korrekt aufsummiert werden.

EXCEL kennt die Funktion `ZUFALLSZAHL()`, die ohne Argumente aufgerufen wird und die bei jeder Neuberechnung eine Zufallszahl zwischen 0 (einschließlich) und 1 (ausschließlich) liefert. Probieren Sie deren Funktionsweise aus, eine Neuberechnung können Sie in EXCEL über die Taste F9 erzwingen.

Wir legen fest, dass eine erzeugte Zufallszahl kleiner als 0,5 „Kugel fällt nach links" bedeutet und eine größere Zufallszahl „Kugel fällt nach rechts" gewertet wird. Um flexibel zu bleiben, schreiben wir den Wert 0,5 nicht als Konstante in die Formeln, sondern in das Feld `A1`, das wir dann jeweils mit absoluten Bezügen referenzieren.

Diese Unterscheidung wird mit Hilfe der `WENN()`-Funktion vorgenommen. Unsere Bedingung ist die Überprüfung, ob die Zufallszahl kleiner als 0,5 ist. Dann soll die nach links fallende Kugel durch ein „<"-Zeichen dargestellt werden, sonst durch das „>"-Zeichen. Tragen Sie entsprechende Funktion in die Zelle `N2` ein.

Fällt die Kugel nach links, dann fällt sie ins Feld `M3`, fällt sie nach rechts, dann fällt sie in `O3`. In genau diese Felder müssen die nächsten Formeln eingetragen werden. Macht man dies, dann merkt man schnell, dass dann an den jeweiligen Entscheidungspunkten immer ein Versuchsausgang erzeugt wird. Dies entspricht jedoch nicht der Realität, da die Kugel nur dann auf `M3` fällt, wenn der Ausgang in `N2` das Ergebnis „<" hatte. Und die Entscheidung in `O3` steht nur an, wenn der Ausgang in M2 „>" ist.

Dies bedeutet, dass wir eine zweistufige Verzweigung benötigen. Diese muss in `M3` in Langschrift lauten:

WENN in `N2` „<" steht, DANN entscheide ob rechts oder links SONST setze einen „."

In EXCEL-Syntax:

```
=WENN(N2="<";WENN(ZUFALLSZAHL()<$A$1;"<";">");".")
```

In O3 machen Sie aus dem ersten „<“-Zeichen ein „>“.

In der nächsten vierten Zeile finden die Entscheidungen in den Zellen L4, N4 und P4 statt. Dabei können Sie M3 nach L4 kopieren und O3 nach P4.

Für die Zelle N4 müssen wir uns jedoch etwas einfallen lassen und die bisher generierte Formel erweitern: Die Entscheidung in N4 muss ja dann stattfinden, wenn der Ausgang in M3 ein „>“ war *oder* wenn der Ausgang in O3 ein „<“ war. Dies erreichen wir, indem wir die Bedingung der ersten Wenn-Verzweigung entsprechend durch ein logisches *Oder* erweitern und die beiden genannten Fälle zusammenfassen:

```
=WENN(ODER(M3=">";O3="<");WENN(ZUFALLSZAHL()<$A$1;"<";">");".")
```

Damit haben wir drei verschiedene Formeln kreiert: Eine für die Entscheidungszellen am linken Rand, eine für die Entscheidungszellen am rechten Rand und die eben erzeugte für alle Zellen innerhalb des Galtonbretts. Es ist jetzt nur eine Fleißaufgabe, diese Formeln korrekt in die darunterliegenden Zeilen zu kopieren. Wir kopieren insgesamt bis in die Tabellenzeile 13[1].

In der Zeile 14 stellen wir die herabgefallene Kugel durch einen „*“ dar. Dafür müssen wir in den Randzellen B und Z ermitteln, ob die Zelle C13 den Ausgang „<“ bzw. die Zelle Y13 den Ausgang „>“ hatte. Dann soll jeweils die „Kugel“ gezeichnet werden, sonst soll Nichts in dem Feld erscheinen. „Nichts“ kann man anzeigen lassen, indem man zwei leere Anführungszeichen ("") angibt. In den Zellen zwischen B und Z – nur jede zweite Zelle ist betroffen – müssen mit der ODER()-Funktion jeweils die beiden schräg darüber liegenden Zellen abgefragt werden.

```
=WENN(C13="<";"*";"")
```

```
=WENN(Y13=">";"*";"")
```

```
=WENN(ODER(O13=">";Q13="<");"*";"")
```

Schließlich wollen wir die in die einzelnen Fächer gefallenen Kugeln aufsummieren. Hierfür ist es als Zwischenschritt hilfreich, in Zeile 15 in die entsprechenden Felder eine 1 einzutragen, wenn im Feld darüber die Kugel „*“ erschienen ist, sonst soll eine 0 eingetragen werden.

```
=WENN(N14="*";1;0)
```

In Zeile 16 soll schließlich die Summation erfolgen. Dies machen wir, indem wir dort in die Felder B, D, F, H, ..., Z eine einfache Addition eintragen, in N beispielsweise:

```
=N16+N15
```

Diese Formel ist übrigens der Grund dafür, dass EXCEL einen Zirkelbezug anmahnt und deshalb wie oben erwähnt die Iterationsanzahl beschränkt werden muss. Jedes Mal, wenn man BLATT NEU BERECHNEN aufruft, wird eine neuer Kugelweg angezeigt, die Kugel in Zeile 14 passend positioniert und die Zähler in Zeile 16 korrekt hochgezählt.

1.3.3 Zurücksetzen des Modells

Soweit funktioniert unsere Simulation schon ganz perfekt, allerdings haben wir keine Möglichkeit, unsere Zähler in Zeile 16 wieder auf null zu setzen, um ein neues Experiment zu starten. Hier hilft ein Trick:

In Zelle AB16 schreiben wir die entsprechende Additionsformel:

```
=AB16+AB15
```

und weil weder in AB15 noch in AB16 Werte eingetragen sind, erhalten wir in AB16 das Ergebnis 0. Nun können wir diese Zelle AB16 markieren und kopieren (<Strg><C>) und dann in

[1] Sie können natürlich auch mehr oder weniger Zeilen erzeugen, die weiteren Zellbezüge im Skript gehen aber von der genannten Anzahl aus.

alle Summenzellen B16 bis Z16 einfügen (<Strg><V>), womit wieder alle Zähler auf 0 gesetzt werden.

Es ist allerdings schon etwas lästig, die Zelle AB16 mehrfach in die Summenzellen kopieren zu müssen, es ist zudem ein Vorgang, der immer gleich abläuft und der schließlich dadurch prädestiniert ist, ihn zu automatisieren. Für die Automatisierung gibt es in EXCEL die Möglichkeit *Makros* zu erzeugen:

Bei neueren EXCEL-Versionen müssen Sie zunächst im Menü DATEI / OPTIONEN / MENÜBAND ANPASSEN bei den HAUPTREGISTERKARTEN den Eintrag ENTWICKLER aktivieren. Hernach steht Ihnen die Registerkarte ENTWICKLERTOOLS zur Verfügung, in welcher Sie die Makroaufzeichnung starten können.

Optional wählen Sie in älteren Versionen die Menüoption EXTRAS / MAKRO / AUFZEICHNEN. In beiden Fällen erscheint ein Fenster, in welchem Sie einen Makronamen angeben, außerdem können Sie noch eine kurze Beschreibung anfügen, was das Makro leistet. Achtung: In dem Moment, in dem Sie auf OK klicken, beginnt die Aufzeichnung des Makros und alle Ihre Aktionen in EXCEL (Mausaktionen, Tastendrucke usw.) werden wie bei einem Rekorder aufgezeichnet!

Für die Aufzeichnung des Makros klicken Sie zuerst auf die Zelle AB16. Tun Sie dies auch unbedingt, falls die Markierung schon auf der betreffenden Zelle stehen sollte! Dann rufen Sie im Menü BEARBEITEN die Option KOPIEREN auf, klicken hernach auf die Zelle B16 und wählen dann im Menü BEARBEITEN die Option EINFÜGEN. Dann klicken Sie auf D16 und wählen wieder BEARBEITEN / EINFÜGEN, dann auf F16 und so fort, bis Sie die kopierte Zelle in allen Summenzellen eingefügt haben. Am Ende klicken Sie nochmals auf die Zelle AB16.

Jetzt ist die Aufzeichnung beendet, deshalb wählen Sie EXTRAS / MAKRO / AUFZEICHNUNG BEENDEN. Anstatt bei der Aufzeichnung des Makros über die Menüs zu gehen, können Sie natürlich genauso gut <Strg><C> und <Strg><V> verwenden.

Um das aufgezeichnete Makro ausführen zu lassen, wählen Sie EXTRAS / MAKRO / MAKROS. Im erscheinenden Fenster wählen Sie – sofern Sie mehrere Makros erstellt haben – das Gewünschte aus und klicken auf AUSFÜHREN.

Falls Ihnen das Aufrufen über die Menüs zu aufwändig ist, können Sie im letztgenannten Fenster über OPTIONEN auch eine Tastenkombination zuordnen, mit der das Makro aufgerufen werden kann. Sollte die Aufzeichnung aus irgendeinem Grund fehlgeschlagen sein, können Sie dort auch ein Makro wieder löschen.

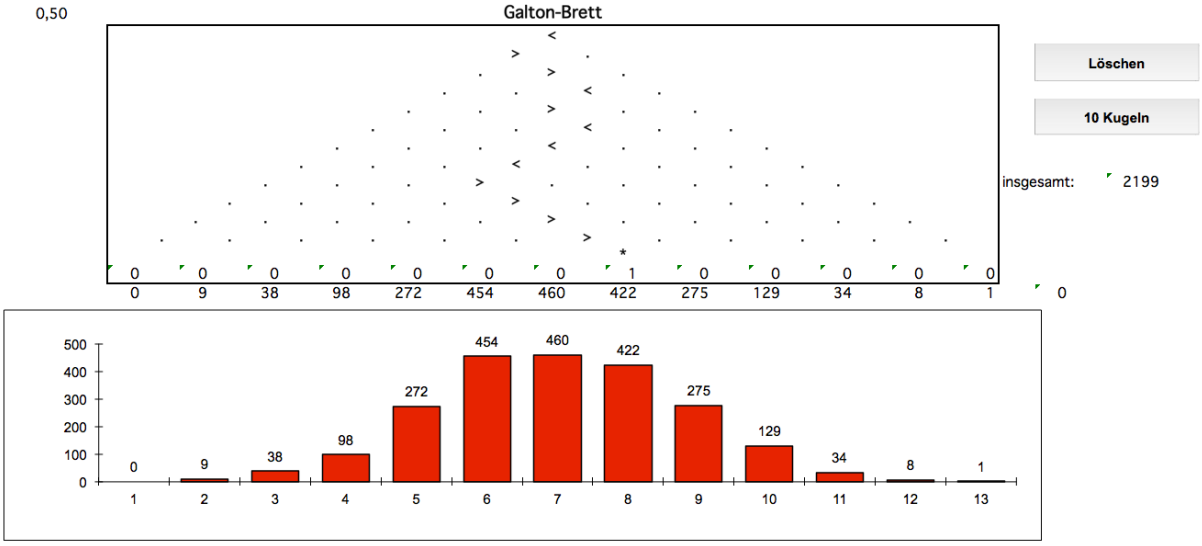
Noch benutzerfreundlicher wird der Makroaufruf, wenn Sie diesen an eine Schaltfläche binden: Erstellen Sie über EINFÜGEN ein TEXTFELD oder eine FORM und platzieren Sie dieses Objekt auf dem Tabellenblatt. Das Textfeld hat den Vorteil, dass es beschriftet werden kann, Sie können so z.B. „Löschen“ in das Feld schreiben. Wenn Sie einen Rechtsklick auf das Objekt machen, erhalten Sie im Kontextmenü die Option MAKRO ZUWEISEN, die Sie bitte auswählen. Aus der erscheinenden Liste wählen Sie Ihr Löschmakro aus und damit ist es an die Schaltfläche gebunden. Jedes Mal, wenn Sie auf die Schaltfläche klicken, werden die Summenspeicher auf null zurückgesetzt.

Da für Zufallsexperimente das *Gesetz der großen Zahl* gilt, ist vielleicht ein Makro samt Schaltfläche sinnvoll, das 10 oder mehr Kugeldurchläufe simuliert.

1.3.4 Graphische Darstellung

Schließlich wollen wir die generierten Ergebnisse noch graphisch darstellen lassen: Markieren Sie die Zellen mit den Summen und erstellen Sie mit diesen Werten ein Säulendiagramm direkt unterhalb der Ergebniszellen. Dieses Diagramm können Sie in dessen Breite so anpassen, dass die einzelnen Balken genau unter den zugehörigen Werten stehen.

Beim Abspeichern des fertig gestellten Arbeitsblatts kann es sein, dass Sie von EXCEL eine Meldung erhalten, dass die Datei mit den erstellten Makros nicht unter dem seitherigen Dateityp gespeichert werden kann. Wählen Sie deshalb im DATEI SPEICHERN-Dialog als Dateityp *EXCEL mit Makros* aus.



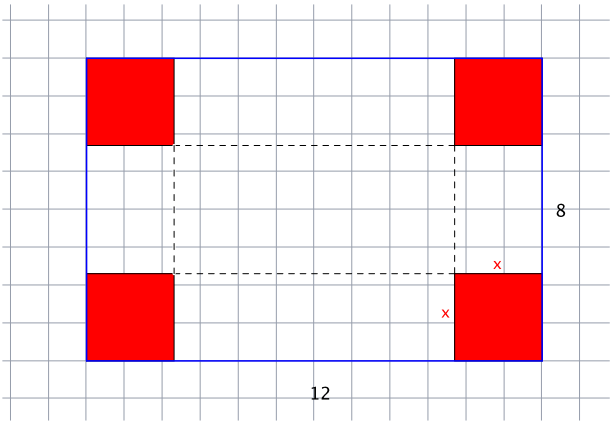
1.4 Optimieren mit EXCEL

Optimierungen sind sehr anwendungsnahe Probleme, welche mathematisch mit unterschiedlichen Mitteln gelöst werden können. Wir bearbeiten insgesamt drei unterschiedlich komplexe Beispiele.

1.4.1 Mathematische Grundlagen

1.4.1.1 Quaderoptimierung

Von einem rechteckigen Stück Blech mit den Maßen 12 und 8 werden an allen vier Ecken quadratische Stücke mit der Seitenlänge x entfernt. Die zwischen den herausgenommenen Quadraten stehen gebliebenen Blechstreifen werden rechtwinklig nach oben gekantet, so dass ein quaderförmiger und nach oben offener Behälter entsteht.



Die Funktion für das Volumen dieses quaderförmigen Behälters in Abhängigkeit von der Seitenlänge x der Quadrate lautet:

$V(x) = (12–2x)\cdot(8–2x)\cdot x = 4x^3 – 40x^2 + 96x$

Der entstehende quaderförmige Behälter soll unter den gegebenen Bedingungen das größte Volumen erhalten. Klassisch wird diese Aufgabe mit Hilfe der Differenzialrechnung gelöst, indem man das Maximum der gegebenen Funktion bestimmt. Hierzu bildet man die erste Ableitung der zugrundeliegenden Funktion

$V'(x) = 12x^2 – 80x + 96$, setzt diese gleich Null

$12x^2 – 80x + 96 = 0$

und erhält für x:

x_1=1.57, x_2=*5.10*

Die zweite Lösung liegt außerhalb unseres realiter gegebenen Definitionsbereichs. Für die Überprüfung auf ein Maximum bilden wir die zweite Ableitung

$V''(x) = 24x – 80$

und bestimmen deren Wert an der Stelle 1.57:

$24 \cdot 1.57 – 80 = –42.23$

Es handelt sich tatsächlich um ein Maximum.

1.4.1.2 Zylinderoptimierung

Konserven werden meist in zylinderförmigen Dosen verkauft und es fällt auf, dass Dosen von verschiedenen Firmen und mit unterschiedlichsten Inhalten nahezu gleiche Maße haben. Diese Maße sind das Ergebnis eines Optimierungsprozesses hinsichtlich des größten Volumens bei kleinster Oberfläche. Beide Größen (Volumen und Oberfläche sind bei zylindrischen Behältern ausschließlich vom *Radius* der Grund- und Deckfläche und der *Höhe* abhängig und es gilt:

$V(r, h) = \pi r^2 \cdot h$ und $O(r, h) = 2\pi r \cdot (r+h)$

Im Handel gängige Dosen haben ein Volumen von 850 ml. Welchen Radius muss eine Dose mit 850 ml Volumen haben, wenn sie 80mm hoch sein soll?

Dies errechnen wir leicht aus der Volumenformel:

$850\,000\ \text{mm}^3 = \pi r^2 \cdot 80\text{mm} \quad r = \sqrt{\frac{850\,000}{\pi \cdot 80}} \approx 58{,}16\ \text{mm}$

Schwieriger ist die Frage nach der minimalen Oberfläche bei gegebenen Volumen von 850 ml zu beantworten. Hierbei handelt es sich um eine Optimierung mit gegebener Nebenbedingung. Diese Nebenbedingung – die Festlegung des Volumens auf 850 000 mm³ – lautet formal:

$850\,000\ \text{mm}^3 = \pi r^2 \cdot h$

Diese Gleichung lösen wir vorteilhaft[2] nach *h* auf:

$$h = \frac{850\,000 mm^3}{\pi \cdot r^2}$$

und setzen diese Nebenbedingung nun in die zu optimierende Gleichung ein:

$O(r, h) = 2\pi r \cdot (r+h)$

$$O(r) = 2\pi r \cdot \left(r + \frac{850\,000 mm^3}{\pi \cdot r^2} \right) = 2\pi r^2 + \frac{2\pi r \cdot 850\,000 mm^3}{\pi \cdot r^2} = 2\pi r^2 + \frac{2 \cdot 850\,000 mm^3}{r}$$

und damit schließlich:

$$O(r) = 2\pi r^2 + 1\,700\,000 \cdot r^{-1}$$

Für diese Funktion suchen wir ein Minimum. Wir bilden daher die erste Ableitung

$O'(r) = 4\pi r - 1\,700\,000 \cdot r^{-2}$

und setzen diese gleich Null:

$$0 = 4\pi r - \frac{1\,700\,000}{r^2} \quad \Leftrightarrow \quad \frac{1\,700\,000}{r^2} = 4\pi r \quad \Leftrightarrow \quad r^3 = \frac{1\,700\,000}{4\pi}$$

Daraus folgt für *r*:

$r \approx 51{,}33$ mm

und für *h* errechnen wir laut der zugehörigen obigen Gleichung

$$h = \frac{850\,000 mm^3}{\pi \cdot (51{,}33 mm)^2} \approx 102{,}69 mm$$

Eine optimale Dose hat somit ein quadratisches Verhältnis.

1.4.1.3 Das Problem des Handlungsreisenden

Das *Problem des Handlungsreisenden* (engl: *travelling salesman problem – tsp*) beschreibt ursprünglich die Aufgabe, einen optimalen, d.h. den kürzest möglichen Weg für eine Rundreise durch mehrere Städte zu finden. Ein analoges Problem hat ein Paketausfahrer, der in einer Stadt mehrere Pakete zustellen und danach wieder ins Depot zurückkehren soll. Genau dieselben Probleme treten beim Design von Leiterplatten oder von Computerchips auf, auch die Sequenzierung des Genoms basiert letztlich auf der gleichen Aufgabenstellung.

So einfach und intuitiv die Problemlage verständlich ist, so schwierig bis unmöglich ist dessen Lösung in der Realität. Es gibt schlichtweg keinen intelligenten Algorithmus, der dieses Problem in akzeptabler Zeit löst!

Natürlich ist die Problematik einfach zu lösen, wenn man nur ganz wenige Orte auf seiner Rundreise besuchen muss. Dann rechnet man alle Möglichkeiten durch und kommt schnell zu einem optimalen Weg. Werden es jedoch mehr Orte, dann kommt es zur einer kombinatorischen Explosion. Unter der Maßgabe, dass jeder Ort mit jedem verbunden ist und ein schneller Computer 1 Million Wege pro Sekunde untersuchen kann, gibt es die folgenden ernüchternden Zusammenhänge:

---

[2] damit vermeiden wir die Bestimmung von Wurzeln

Orte	Wege	Rechenzeit
10	181.440	0,18 sec
11	1.814.400	1,81 sec
12	19.958.400	19,96 sec
13	239.500.800	4,00 min
14	3.113.510.400	51,90 min
15	43.589.145.600	12,11 Std
16	6,54 E+11	7,57 Tage
17	1,05 E+13	4,03 Mon
18	1,78 E+14	5,71 Jahre
19	3,20 E+15	102,92 Jahre
20	6,08 E+16	1955,46 Jahre

Eine optimale Bestimmung einer etwas umfangreicheren Route mit Hilfe einer Überprüfung aller Wege scheidet somit aus. Natürlich gibt es pfiffige Algorithmen, mit denen man eine recht gute Lösung in akzeptabler Zeit erreichen kann, aber von diesen Algorithmen weiß man auch, dass sie maximal das 1,4-fache des optimalen Rundwegs erzeugen. Wir greifen diese Thematik hier auf, indem wir EXCEL eine optimale Rundreise durch zehn gegebene Orte bestimmen lassen.

1.4.2 Realisation in EXCEL

Um die oben diskutierten Probleme in EXCEL lösen zu können, bedienen wir uns des *Solvers*. Dieser Programmteil ist wohl schon in EXCEL eingebaut, er muss jedoch vor seiner Verwendung freigeschaltet werden. Diese Freischaltung verläuft auf den unterschiedlichen Betriebssystemen und in den jeweiligen Programmversionen leider unterschiedlich,.

Auf *Windows-Rechnern* wählt man DATEI / OPTIONEN und im erscheinenden Fenster EXEL-OPTIONEN in der Leiste links unten den Eintrag ADD-INS. Aus den dargebotenen Add-Ins wählt man den Solver und klickt auf LOS. In dem dann erscheinenden Fenster klickt man den Eintrag SOLVER und hernach OK an.

Auf dem Mac geht man über die Menüoption EXTRAS / EXCEL ADD-INS , wählt den SOLVER und klickt auf OK.

Danach steht in beiden Fällen der SOLVER im Menüband DATEN ganz rechts zur Verfügung.

1.4.2.1 Quaderoptimierung

	A	B
1	Länge der Blechplatte:	12
2	Breite der Blechplatte:	8
3	Länge des Ausschnitts x:	2
4		
5	Volumen des Behälters:	64

Zunächst tragen wir jedoch die benötigten Werte in ein Tabellenblatt ein. Das Volumen des Behälters soll aus den gelb hinterlegten Werten nach der oben angegebenen Formel berechnet werden.

Überprüfen Sie zuerst die Korrektheit Ihrer Formel und versuchen Sie dann, durch Variation der Länge des Ausschnitts *x* ein möglichst großes Volumen zu erzeugen.

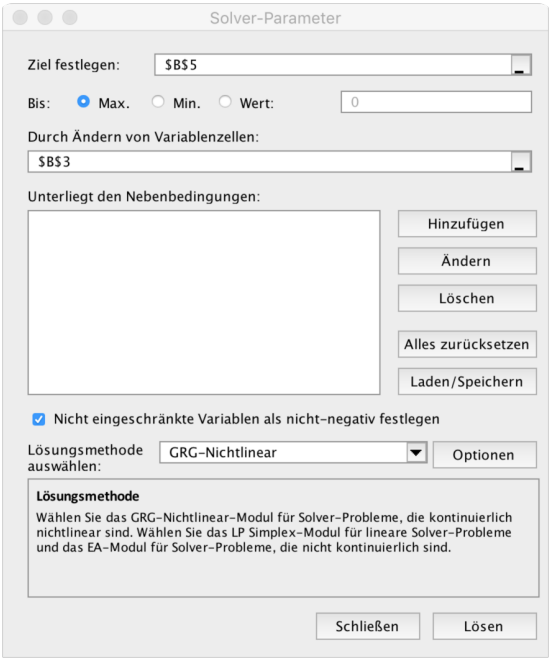
Übertragen Sie nun diese Aufgabe auf den Solver. Es erscheint das nebenstehende Fenster, in dem Sie nur einige wenige Eingaben machen müssen:

Als Ziel legen Sie die zu optimierende Zelle fest. Bei uns ist dies die Zelle `B5`, welche das Volumen des Quaders enthält. Den Verweis können Sie wie gewohnt durch Klicken erzeugen.

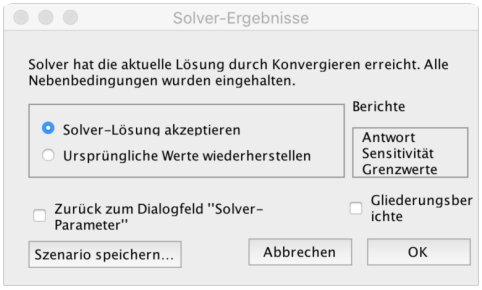
Wir wollen ein maximales Volumen erhalten, deshalb klicken Sie in der nächsten Zeile die Option MAX. an.

Schließlich muss der Solver noch wissen, welche Zellen er im Hinblick auf eine Maximierung des Volumens verändern darf. Dies ist die Zelle B3, in welcher die Größe des *x*-Ausschnitts eingetragen ist.

Weitere Angaben müssen wir für dieses einfache Beispiel nicht machen, klicken Sie deshalb abschließend auf LÖSEN.



Kurz darauf öffnet sich ein weiteres Fenster, in welchem der Solver meldet, dass eine Lösung gefunden werden konnte. Gleichzeitig wurde diese Lösung mitsamt der beteiligen und veränderten Parameter in das Tabellenblatt eingetragen. Sie haben die Möglichkeit, dieses Ergebnis zu akzeptieren oder die ursprünglichen Werte wieder herzustellen.

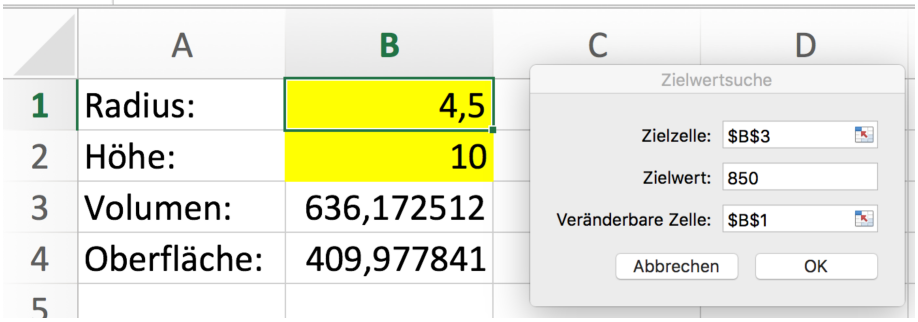


1.4.2.2 Zylinderoptimierung

Für die Optimierung des Zylinders tragen Sie ebenfalls zuerst die Eingangswerte und die daraus errechneten Größen in ein neues Tabellenblatt ein. Volumen und Oberfläche werden über die oben dargestellten Formeln errechnet. Überprüfen Sie die korrekte Eingabe der Formeln anhand der nebenstehenden Werte und versuchen Sie dann durch manuelle Veränderung der Werte, ein Volumen von 850 ml bei minimaler Oberfläche zu erreichen! Sie werden sehen, dass dies in ein langwieriges Verfahren ausartet! Beschränken Sie sich daher zunächst darauf, das geforderte Volumen in Abhängigkeit vom Radius zu erreichen, d.h., Sie verändern so lange den Radius, bis Sie ein Volumen von 850 ml erreicht haben.

	A	B
1	Radius:	4,5
2	Höhe:	10
3	Volumen:	636,172512
4	Oberfläche:	409,977841

Diese Aufgabe können Sie relativ einfach und schnell lösen lassen. Man nennt so etwas eine *Zielwertsuche* und auch hierbei ist Ihnen EXCEL behilflich. Setzen Sie zuvor jedoch den Radius auf 4,5. Rufen Sie dann DATEN / WAS-WÄRE-WENN-ANALYSE / ZIELWERTSUCHE auf und geben Sie in das erscheinende Fenster als Zielzelle die Zelle mit dem errechneten Volumen B3 ein, als Zielwert die geforderten 850 und als veränderbare Zelle die Radiuszelle B1.

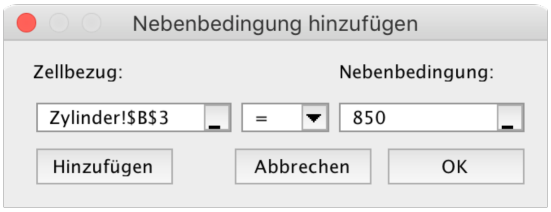


Klicken auf OK startet die Zielwertsuche und umgehend präsentiert EXCEL das ermittelte Ergebnis.

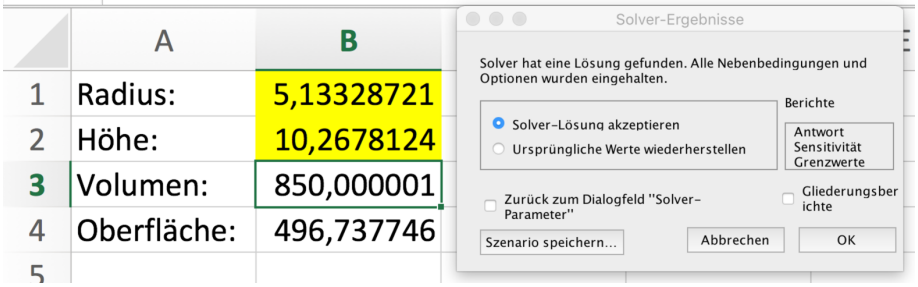
Die eigentliche Lösung der Aufgabe, ein Volumen von 850 ml bei minimaler Oberfläche zu finden, erfordert wieder den *Solver*. Da wir aktuell schon eine Lösung mit 850 ml gefunden und eingetragen haben, sollten Sie sowohl den Radius als auch die Höhe wieder abändern.

Im Solver-Fenster tragen Sie als Ziel den Bezug der Zelle mit dem Oberflächenwert ein, dieser soll minimal werden durch Verändern der Zellen `B1` und `B2`.

Nun müssen wir allerdings noch die zweite Bedingung formulieren, dass nämlich das Volumen 850 ml betragen soll. Ein Klick auf die Schaltfläche HINZUFÜGEN bei den Nebenbedingungen öffnet ein weiteres Fenster, in welchem diese Nebenbedingung(en) formuliert werden können.



Das Volumen steht in Zelle `B3` und der dort enthaltene Wert soll gleich 850 sein. Geben Sie diese Nebenbedingungen in das Fenster ein und klicken Sie abschließend auf OK. Sie gelangen dann zum Solver-Fenster zurück, in welchem nun die formulierte Nebenbedingung eingetragen ist. Ein Klick auf LÖSEN startet die Suche des korrekten Ergebnisses, mit dem sich EXCEL umgehend meldet:



Welche optimalen Maße hat eine Konservendose mit 425 ml Inhalt?

1.4.2.3 Problem des Handlungsreisenden

Wir wollen EXCEL das überschaubare Problem einer optimalen Rundreise durch die zehn Städte Frankfurt, Mannheim, Würzburg, Erfurt, Karlsruhe, Stuttgart, Nürnberg, Kassel, Augsburg und München bearbeiten lassen. Hierfür benötigen wir eine Entfernungstabelle zwischen diesen Orten:

	A	B	C	D	E	F	G	H	I	J	K	L
1			Ff	Ma	Wü	Er	Ka	St	Nü	Ks	Au	Mü
2			1	2	3	4	5	6	7	8	9	10
3	Ff	1	0	88	120	256	143	210	228	192	360	393
4	Ma	2	88	0	177	329	65	132	238	264	285	341
5	Wü	3	120	177	0	194	188	147	119	211	245	284
6	Er	4	256	329	194	0	387	340	231	157	426	396
7	Ka	5	143	65	188	387	0	78	248	324	227	297
8	St	6	210	132	147	340	78	0	207	367	161	231
9	Nü	7	228	238	119	231	248	207	0	308	143	166
10	Ks	8	192	264	211	157	324	367	308	0	442	475
11	Au	9	360	285	245	426	227	161	143	442	0	80
12	Mü	10	393	341	284	396	297	231	166	475	80	0

Diese Entfernungstabelle finden Sie in der EXCEL-Datei `TSP_Vorlage.xlsx` in Moodle. Unterhalb dieser Entfernungsmatrix legen Sie einen möglichen Rundweg von Hand an, indem Sie die Ortsnummern von 1 bis 10 von Hand eingeben. Am Ende geht es wieder zum Startpunkt zurück, realisieren Sie dies unbedingt, indem Sie in Zelle `M14` einen Verweis auf Zelle `C14` eintragen:

11	Au	9	360	285	245	426	227	161	143	442	0	80
12	Mü	10	393	341	284	396	297	231	166	475	80	0
13												
14			1	2	3	4	5	6	7	8	9	10 =C14

Nun müssen in Zeile `15` die jeweiligen Entfernungen eingetragen werden. In `C15` soll die Entfernung zwischen den Orten 1 (Frankfurt) und 2 (Mannheim) und in `D15` die Entfernung zwischen den Orten 2 (Mannheim) und 3 (Würzburg) eingetragen werden usw. Natürlich tragen wir die Entfernungen nicht von Hand ein, sondern verwenden hierfür die EXCEL-Funktion INDEX():

`index(<matrix>;<zeile>;<spalte>)`

Die anzugebende Matrix ist die Matrix mit den Entfernungsangaben – und zwar nur der Bereich mit den Entfernungen und nicht die Kopfzeilen und –spalten mit den Städtenamen und –nummern. Stellen Sie diesen Bezug absolut dar!

Der Zeilenindex ist die Nummer des Startorts (also `C14`) und der Spaltenindex ist die Nummer des Zielorts (also `D14`).

9	Nü	7	228	238	119	231	248	207	0	308	143	166	
10	Ks	8	192	264	211	157	324	367	308	0	442	475	
11	Au	9	360	285	245	426	227	161	143	442	0	80	
12	Mü	10	393	341	284	396	297	231	166	475	80	0	
13													
14			1	2	3	4	5	6	7	8	9	10	1
15			=INDEX($C$3:$L$12;C14;D14)										

Nach dem Abschluss der Eingabe muss im Feld `C15` die Entfernung von Frankfurt nach Mannheim korrekt mit 88 km erscheinen. Die eben eingegebene Formel wird dann in den Bereich von `D15` bis `L15` kopiert. Überprüfen Sie, dass dort jeweils die korrekten Entfernungen eingetragen werden. Die Summe der Entfernungen lassen Sie in `M15` eintragen, wenn alles korrekt ist, müsste dort die Zahl 2354 erscheinen.

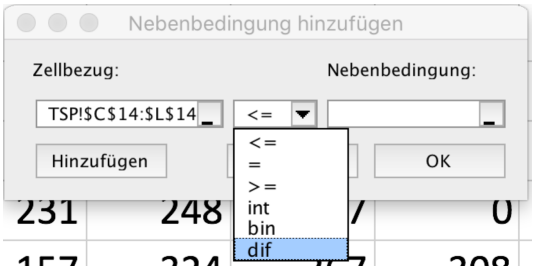
Versuchen Sie dann aufgrund Ihrer geographischen Kenntnisse einen optimaleren Rundweg zu finden, indem Sie eine neue Reihenfolge der Städtenummern in den Bereich *C14* bis *L14* eintragen. Vergleichen Sie Ihr Ergebnis mit demjenigen Ihrer Kommilitonen! Worauf müssen Sie achten, wenn es ein Rundweg werden soll, bei dem jeder Ort genau einmal besucht wird?

Schließlich übertragen wir diese Aufgabe dem *Solver*. Starten Sie den *Solver* und geben Sie die benötigten Werte ein:

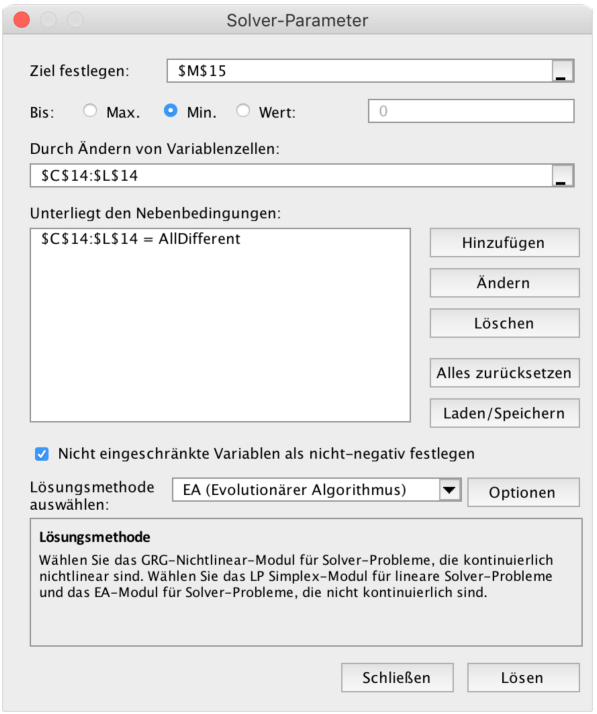
Das Ziel ist die Zelle mit der Wegsumme `M15`.

Diese soll minimal werden.

Geändert werden kann hierfür der Bereich von `C14` bis `L14`, in welchem die Abfolge der Städtenummern eingetragen ist.



Als Nebenbedingung haben Sie sicher herausgefunden, dass alle Städte nur einmal besucht werden dürfen. Dies bedeutet, dass die Werte mit den Städtenummern im Bereich `C14` bis `L14` alle voneinander verschieden sein müssen! Diese Bedingung tragen Sie ein, indem Sie in das Nebenbedingungsfenster DIF auswählen. Den Zellbezug können Sie durch Markieren festlegen.



Als Lösungsmethode wählen Sie den *Evolutionären Algorithmus* und tragen als OPTION im erscheinenden Fenster eine Höchstzeit von 20 Sekunden ein. Haben Sie alles korrekt ausgefüllt, dann sollte Ihr Solver-Fenster die nebenstehenden Eintragungen haben.

Sind alle Eintragungen korrekt gemacht, klicken Sie auf LÖSEN. Nach 20 Sekunden informiert EXCEL, dass die Höchstzeit für die Lösung erreicht wurde. Klicken Sie auf STOPP und überprüfen Sie das erreichte Ergebnis.

Um weitere Versuche anzustellen, beispielsweise mit längeren oder kürzeren Lösungszeiten, sollten Sie zuvor die Besuchsreihenfolge wieder auf 1, 2, 3, ..., 10 setzen. Dies können Sie natürlich immer manuell machen oder dafür ein Makro kreieren.

1.5 Weitere Anwendungen in EXCEL

1.5.1 Erstellen von Wahrheitswertetafeln

EXCEL kennt die logischen Funktionen NICHT, UND, ODER und XODER sowie die logischen Konstanten WAHR und FALSCH. Die Syntax der logischen Funktionen ist immer dieselbe, hier gezeigt am Beispiel der Funktion `UND()`:

UND(Wahrheitswert1; Wahrheitswert2; Wahrheitswert3; ...)

wobei bis zu 256 Wahrheitswerte angegeben werden können.

Konkret sieht der Aufruf beispielsweise folgendermaßen aus:

```
=UND(A2;B2)
```

wobei dann die referenzierten Zellen `A2` und `B2` die Wahrheitswerte WAHR oder FALSCH enthalten müssen. Das Ergebnis einer logischen Funktion ist immer wieder einer der beiden Wahrheitswerte WAHR oder FALSCH.

Damit lassen sich mit nur geringem Aufwand Wahrheitswertetafeln erstellen:

	A	B	C	D	E
1	Wahrheitswertetafeln mit EXCEL				
2					
3	A	B	NICHT A	A UND B	A ODER B
4	WAHR	WAHR	FALSCH	WAHR	WAHR
5	WAHR	FALSCH	FALSCH	FALSCH	WAHR
6	FALSCH	WAHR	WAHR	FALSCH	WAHR
7	FALSCH	FALSCH	WAHR	FALSCH	FALSCH

Dazu schreibt man in die Spalten `A` und `B` die möglichen Kombinationen der logischen Variablen `A` und `B` und kann dann in den weiteren Spalten die Ergebnisse logischer Verknüpfungen darstellen lassen.

Neben den logischen Funktionen NICHT, UND und ODER enthält EXCEL auch die logische Verknüpfung XODER, so dass auch das *ausschließliche Oder* dargestellt werden kann. Falls dies in anderen Tabellenkalkulationen nicht der Fall ist, kann man das *ausschließliche Oder* durch den nachfolgenden äquivalenten Ausdruck ersetzen:

$(A \wedge \neg B) \vee (B \wedge \neg A)$

in der EXCEL-Syntax:

```
=ODER(UND(A4;NICHT(B4));UND(B4;NICHT(A4)))
```

Auch die Subjunktion und die Bijunktion lassen sich leicht ersetzen. Für die Subjunktion $A \rightarrow B$ gilt:

$\neg A \vee B$, bzw.:

```
=ODER(NICHT(A4);B4)
```

und für die Bijunktion: $A \leftrightarrow B$:

$(\neg A \wedge \neg B) \vee (A \wedge B)$, bzw.

```
=ODER(UND(NICHT(A4);NICHT(B4));UND(A4;B4))
```

1.5.2 Mathematisches Golfspiel

Vorbereitend auf die Thematik zu diophantischen Gleichungen sollen Sie einen *mathematischen Golfplatz* auf einem Tabellenblatt erstellen. Die zugrundeliegende mathematische Fragestellung sei an einem Beispiel erläutert:

Sie haben zwei Golfschläger; mit dem einen können Sie immer 9 (Meter) vorwärts oder zurückspielen, mit dem zweiten kommen Sie immer nur 15 (Meter) weiter oder zurück. Welche Zahlen können Sie beim Start von der Zahl 0 aus mit diesen Schlägern erreichen?

Natürlich können Sie mit den einzelnen Schlägern die (auch negativen) Vielfachen von 9 und 15 erreichen. Sie können aber auch beide Schläger beliebig kombinieren und beispielsweise 3 Schläge mit dem 15er-Schläger voraus und 2 Schläge mit dem 9er-Schläger zurückspielen – dann wären Sie bei der Zahl 27 angelangt. Auf einem Tabellenblatt soll nun zumindest ein Ausschnitt derjenigen Zahlen dargestellt werden, welche Sie mit zwei gegebenen Schlägern erreichen können. Für die Übersichtlichkeit vereinbaren wir, dass wir mit dem 9er-Schläger horizontal und mit dem 15-er Schläger vertikal spielen wollen. Das Tabellenblatt soll hinterher folgendermaßen aussehen:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI
1	Startzahl	0																																	
2	Schläger 1	9															-9	←		→	9														
3	Schläger 2	15																																	
4				165	174	183	192	201	210	219	228	237	246	255	264	273	282	291	300	309	318	327	336	345	354	363	372	381	390	399	408	417	426	435	
5	erreichbarer Wert:	3		150	159	168	177	186	195	204	213	222	231	240	249	258	267	276	285	294	303	312	321	330	339	348	357	366	375	384	393	402	411	420	
6				135	144	153	162	171	180	189	198	207	216	225	234	243	252	261	270	279	288	297	306	315	324	333	342	351	360	369	378	387	396	405	
7				120	129	138	147	156	165	174	183	192	201	210	219	228	237	246	255	264	273	282	291	300	309	318	327	336	345	354	363	372	381	390	
8				105	114	123	132	141	150	159	168	177	186	195	204	213	222	231	240	249	258	267	276	285	294	303	312	321	330	339	348	357	366	375	
9				90	99	108	117	126	135	144	153	162	171	180	189	198	207	216	225	234	243	252	261	270	279	288	297	306	315	324	333	342	351	360	
10				75	84	93	102	111	120	129	138	147	156	165	174	183	192	201	210	219	228	237	246	255	264	273	282	291	300	309	318	327	336	345	
11				60	69	78	87	96	105	114	123	132	141	150	159	168	177	186	195	204	213	222	231	240	249	258	267	276	285	294	303	312	321	330	
12				45	54	63	72	81	90	99	108	117	126	135	144	153	162	171	180	189	198	207	216	225	234	243	252	261	270	279	288	297	306	315	
13				30	39	48	57	66	75	84	93	102	111	120	129	138	147	156	165	174	183	192	201	210	219	228	237	246	255	264	273	282	291	300	
14				15	24	33	42	51	60	69	78	87	96	105	114	123	132	141	150	159	168	177	186	195	204	213	222	231	240	249	258	267	276	285	
15				0	9	18	27	36	45	54	63	72	81	90	99	108	117	126	135	144	153	162	171	180	189	198	207	216	225	234	243	252	261	270	
16				-15	-6	3	12	21	30	39	48	57	66	75	84	93	102	111	120	129	138	147	156	165	174	183	192	201	210	219	228	237	246	255	
17				-30	-21	-12	-3	6	15	24	33	42	51	60	69	78	87	96	105	114	123	132	141	150	159	168	177	186	195	204	213	222	231	240	
18				-45	-36	-27	-18	-9	0	9	18	27	36	45	54	63	72	81	90	99	108	117	126	135	144	153	162	171	180	189	198	207	216	225	
19				-60	-51	-42	-33	-24	-15	-6	3	12	21	30	39	48	57	66	75	84	93	102	111	120	129	138	147	156	165	174	183	192	201	210	
20				-75	-66	-57	-48	-39	-30	-21	-12	-3	6	15	24	33	42	51	60	69	78	87	96	105	114	123	132	141	150	159	168	177	186	195	
21				-90	-81	-72	-63	-54	-45	-36	-27	-18	-9	0	9	18	27	36	45	54	63	72	81	90	99	108	117	126	135	144	153	162	171	180	
22		15		-105	-96	-87	-78	-69	-60	-51	-42	-33	-24	-15	-6	3	12	21	30	39	48	57	66	75	84	93	102	111	120	129	138	147	156	165	
23		↑		-120	-111	-102	-93	-84	-75	-66	-57	-48	-39	-30	-21	-12	-3	6	15	24	33	42	51	60	69	78	87	96	105	114	123	132	141	150	
24				-135	-126	-117	-108	-99	-90	-81	-72	-63	-54	-45	-36	-27	-18	-9	0	9	18	27	36	45	54	63	72	81	90	99	108	117	126	135	
25		↓		-150	-141	-132	-123	-114	-105	-96	-87	-78	-69	-60	-51	-42	-33	-24	-15	-6	3	12	21	30	39	48	57	66	75	84	93	102	111	120	
26		-15		-165	-156	-147	-138	-129	-120	-111	-102	-93	-84	-75	-66	-57	-48	-39	-30	-21	-12	-3	6	15	24	33	42	51	60	69	78	87	96	105	
27				-180	-171	-162	-153	-144	-135	-126	-117	-108	-99	-90	-81	-72	-63	-54	-45	-36	-27	-18	-9	0	9	18	27	36	45	54	63	72	81	90	
28				-195	-186	-177	-168	-159	-150	-141	-132	-123	-114	-105	-96	-87	-78	-69	-60	-51	-42	-33	-24	-15	-6	3	12	21	30	39	48	57	66	75	
29				-210	-201	-192	-183	-174	-165	-156	-147	-138	-129	-120	-111	-102	-93	-84	-75	-66	-57	-48	-39	-30	-21	-12	-3	6	15	24	33	42	51	60	
30				-225	-216	-207	-198	-189	-180	-171	-162	-153	-144	-135	-126	-117	-108	-99	-90	-81	-72	-63	-54	-45	-36	-27	-18	-9	0	9	18	27	36	45	
31				-240	-231	-222	-213	-204	-195	-186	-177	-168	-159	-150	-141	-132	-123	-114	-105	-96	-87	-78	-69	-60	-51	-42	-33	-24	-15	-6	3	12	21	30	
32				-255	-246	-237	-228	-219	-210	-201	-192	-183	-174	-165	-156	-147	-138	-129	-120	-111	-102	-93	-84	-75	-66	-57	-48	-39	-30	-21	-12	-3	6	15	
33				-270	-261	-252	-243	-234	-225	-216	-207	-198	-189	-180	-171	-162	-153	-144	-135	-126	-117	-108	-99	-90	-81	-72	-63	-54	-45	-36	-27	-18	-9	0	
34				-285	-276	-267	-258	-249	-240	-231	-222	-213	-204	-195	-186	-177	-168	-159	-150	-141	-132	-123	-114	-105	-96	-87	-78	-69	-60	-51	-42	-33	-24	-15	
35				-300	-291	-282	-273	-264	-255	-246	-237	-228	-219	-210	-201	-192	-183	-174	-165	-156	-147	-138	-129	-120	-111	-102	-93	-84	-75	-66	-57	-48	-39	-30	
36				-315	-306	-297	-288	-279	-270	-261	-252	-243	-234	-225	-216	-207	-198	-189	-180	-171	-162	-153	-144	-135	-126	-117	-108	-99	-90	-81	-72	-63	-54	-45	
37				-330	-321	-312	-303	-294	-285	-276	-267	-258	-249	-240	-231	-222	-213	-204	-195	-186	-177	-168	-159	-150	-141	-132	-123	-114	-105	-96	-87	-78	-69	-60	
38				-345	-336	-327	-318	-309	-300	-291	-282	-273	-264	-255	-246	-237	-228	-219	-210	-201	-192	-183	-174	-165	-156	-147	-138	-129	-120	-111	-102	-93	-84	-75	
39				-360	-351	-342	-333	-324	-315	-306	-297	-288	-279	-270	-261	-252	-243	-234	-225	-216	-207	-198	-189	-180	-171	-162	-153	-144	-135	-126	-117	-108	-99	-90	
40				-375	-366	-357	-348	-339	-330	-321	-312	-303	-294	-285	-276	-267	-258	-249	-240	-231	-222	-213	-204	-195	-186	-177	-168	-159	-150	-141	-132	-123	-114	-105	
41				-390	-381	-372	-363	-354	-345	-336	-327	-318	-309	-300	-291	-282	-273	-264	-255	-246	-237	-228	-219	-210	-201	-192	-183	-174	-165	-156	-147	-138	-129	-120	
42				-405	-396	-387	-378	-369	-360	-351	-342	-333	-324	-315	-306	-297	-288	-279	-270	-261	-252	-243	-234	-225	-216	-207	-198	-189	-180	-171	-162	-153	-144	-135	
43				-420	-411	-402	-393	-384	-375	-366	-357	-348	-339	-330	-321	-312	-303	-294	-285	-276	-267	-258	-249	-240	-231	-222	-213	-204	-195	-186	-177	-168	-159	-150	
44				-435	-426	-417	-408	-399	-390	-381	-372	-363	-354	-345	-336	-327	-318	-309	-300	-291	-282	-273	-264	-255	-246	-237	-228	-219	-210	-201	-192	-183	-174	-165	
45																																			

Die Startzahl, von welcher aus gespielt wird soll natürlich variabel bleiben, ebenso die erreichbaren Weiten der beiden Schläger. Diese Vorgaben sollen in `B1` bis `B3` eingegeben werden.

Wenn Sie sich am obigen Beispiel orientieren möchten, dann platzieren Sie die Startzahl per Verweis in das Feld `S24`.

Weiter nach rechts in Zeile `24` gelangen Sie, indem Sie zur Zelle links immer den Wert des 1. Schlägers addieren. Überlegen Sie, welche Verweise hier relativ und welche absolut sein müssen!

Von der Startzahl aus nach links kommen Sie, indem Sie jeweils den Wert des 1. Schlägers subtrahieren.

Entsprechend verfahren Sie in der Spalte `S`: Ausgehend von der Startzahl addieren Sie nach oben den Wert des Schlägers 2, nach unten subtrahieren Sie diesen Wert jeweils von der vorausgehenden Zelle.

Den Hintergrund dieser Vielfachenreihen färben Sie in einer hellen Farbe ein.

Um die anderen Zellen errechnen zu lassen, gibt es mehrere Möglichkeiten. Sie können beispielsweise ausgehend von der vertikalen Vielfachenreihe des Schlägers 2 ausgehend nach rechts den Wert des 1. Schlägers addieren und nach links subtrahieren. Somit können Sie das gesamte „Spielfeld“ einfach und schnell durch Kopieren erstellen.

Sie sehen dann alle (im Spielfeld gelegenen) Zahlen, die Sie mit einer gegebenen Schlägerkombination erreichen können. Überprüfen Sie Ihren Golfplatz, indem Sie andere Schlägerweiten eingeben und das von EXCEL generierte Ergebnis auf Plausibilität überprüfen.

Durch die Vielzahl der dargestellten Zahlen wird unser Golfplatz sehr unübersichtlich. Wir wollen daher noch eine Möglichkeit schaffen, schnell erkennen zu können, ob eine bestimmte Zahl mit den gewählten Schlägern erreicht werden kann. Dieser erreichbare Wert soll in Zelle `B5` eingegeben werden.

Um alle Zellen der Tabelle hervorzuheben, welche den in `B5` angegebenen Wert haben, verwenden Sie die *bedingte Formatierung*. Die *bedingte Formatierung* finden Sie im Menü oder im Menüband; ihre Anwendung ist selbsterklärend – experimentieren Sie einfach. Ein Hinweis dazu: Sie sollten vor dem Aufruf der bedingten Formatierung die gesamte Zahlentabelle markieren.

1.5.3 Diophantische Gleichungen

Ein Bauer bringt Schafe und Enten auf den Markt. Als er gefragt wird, wie viele Tiere es denn jeweils sind, antwortet er nur „Zusammen haben sie 28 Beine.“

Damit kann die Gleichung $e \cdot 2 + s \cdot 4 = 28$ aufgestellt werden.

Für eine konkrete Lösung der Aufgabe bräuchten wir noch eine weitere Information, beispielsweise, um wie viele Tiere es sich insgesamt handelt. Mit etwas Nachdenken kann man aber auch ohne solche Zusatzinformationen problemlos durch Probieren alle möglichen Lösungen aufzählen:

Enten	0	2	4	6	8	10	12	14
Schafe	7	6	5	4	3	2	1	0

Wenn wir davon ausgehen wollen, dass der Bauer lebendige und wohlbehaltene Tiere zum Markt bringt, brauchen wir nur ganzzahlige Lösungen zu suchen. In diesem konkreten Fall sind gar nur Lösungen im Bereich N der natürlichen Zahlen sinnvoll.

Losgelöst vom gegebenen Kontext und erweitert auf den Bereich der ganzen Zahlen hat die Gleichung $e \cdot 2 + s \cdot 4 = 28$ sogar unendlich viele Lösungen:

e	…	-6	-4	-2	…	16	18	20	…
s	…	10	9	8	…	-1	-2	-3	…

1.5.3.1 Mathematische Grundlagen

Gleichungen der Form $x \cdot a + y \cdot b = c$, für die man nur ganzzahlige Lösungen x und y zulässt, nennt man *diophantische Gleichungen*.

DIOPHANT lebte im ägyptischen Alexandria, welches Jahrhunderte lang das wissenschaftliche Zentrum der antiken Welt war. Seine Lebensdaten schwanken zwischen 100 v. Chr. und 350 n. Chr. Über das eigentliche Leben DIOPHANTs weiß man so gut wie nichts. Bekannt sind lediglich seine Werke. DIOPHANT befasste sich mit der Lösung von algebraischen Gleichungen mit mehreren Unbekannten. Heute noch nennt man deshalb algebraische Gleichungen, für die ganzzahlige Lösungen gesucht werden, **diophantische Gleichungen**.

“Für eine Reisegruppe von 66 Personen stehen Kleinbusse mit jeweils 9 oder 15 Sitzplätzen zur Verfügung.“

Durch Probieren findet man relativ leicht, dass die Reisegruppe 4 Neunsitzer und 2 Fünfzehnsitzer chartern muss, um alle Personen Kosten sparend transportieren zu können.

Löst man sich vom Aufgabenkontext und geht über zur allgemeinen diophantischen Gleichung $x \cdot 9 + y \cdot 15 = 66$, so findet man noch viele weitere Lösungen, beispielsweise gilt auch $44 \cdot 9 – 22 \cdot 15 = 66$; ebenso $49 \cdot 9 – 25 \cdot 15 = 66$.

Offensichtlich scheinen solche diophantischen Gleichungen – losgelöst von irgendwelchem Kontext – sehr viele (unendlich viele?) Lösungen zu haben. Ist dies immer so?

„Karl hat en gros eingekauft. Socken zu 6 € je Paar, und T-Shirts zu 15 € je Stück.
Er sagt, er habe insgesamt 227 € ausgegeben. Nun bekommt er Ärger mit seiner Frau…“

Den Ärger bekommt er zu Recht, da für diese diophantische Gleichung auch nach vielem Probieren keine Lösung gefunden werden kann!

Fassen wir unsere seitherigen Erkenntnisse über diophantische Gleichungen zusammen, so können wir festhalten, dass dies Gleichungen in Z (der Menge der ganzen Zahlen) sind und sie entweder (unendlich?) viele oder gar keine Lösungen haben.

Versuchen wir nun zu ergründen, in welchen Fällen wir Lösungen finden und in welchen Fällen keine Lösungen existieren. Greifen wir dazu – um ein erstes „Gefühl“ zu bekommen – nochmals auf unsere beiden Einstiegsaufgaben zurück:

Wenn der Bauer auf die Frage nach der Anzahl seiner Schafe und Enten geantwortet hätte, dass die Anzahl der Beine insgesamt 27 beträgt, wäre – normale, gesunde Tiere vorausgesetzt – sofort offensichtlich geworden, dass es keine Lösung geben kann.

Verändert man die Anzahl der Reisenden in der zweiten Aufgabe um eine Person nach unten oder oben, so bliebe im ersten Fall ein Platz unbesetzt und im 2. Fall bräuchte man ein weiteres Fahrzeug, in dem dann nur eine Person mitfährt. Wie viele Personen mehr (oder weniger) müssten es denn sein, damit die Gleichung wieder lösbar ist?

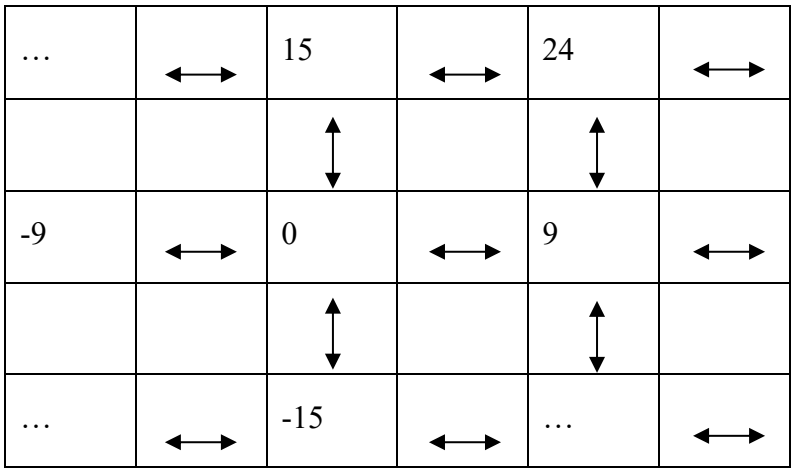
Wenn es drei Personen weniger sind (also 63 Personen), dann ist die Gleichung wieder lösbar: Man würde dann die 2 Fünfzehnsitzer durch 3 Neunsitzer ersetzen:
$7 \cdot 9 + 0 \cdot 15 = 63$

Wenn es drei Personen mehr sind, dann ist die Gleichung ebenfalls wieder lösbar: In dem Fall könnte man 3 Neunsitzer durch 2 Fünfzehnsitzer ersetzen: $1 \cdot 9 + 4 \cdot 15 = 69$

Überlegen Sie nun, welche weiteren Personenzahlen möglich sind und versuchen Sie, daraus schon eine Antwort auf die Frage nach der prinzipiellen Lösbarkeit diophantischer Gleichungen abzuleiten!

Offensichtlich hängen doch die Intervalle, in denen man die Personenzahl verändern kann, von der Differenz der Sitzzahlen beider Bustypen ab. Dabei ist nicht die Differenz als solche gesucht – die wäre im konkreten Fall nämlich 6 – sondern man muss fragen: Welche kleinste Differenz kann denn mit den Vielfachen der Zahlen 9 und 15 erzeugt werden?

Zur Beantwortung dieser Frage spielen wir eine Art „mathematisches Golf“: Wir haben für unser Golfspiel genau zwei Schläger zur Verfügung, mit dem einen Schläger können wir den Ball immer genau 9 Einheiten weiter vor- oder zurücktreiben und mit dem anderen Schläger genau 15 Einheiten vorwärts oder rückwärts. Startpunkt ist die Null, welche Zahlen auf dem Zahlenstrahl können wir erreichen?



Für die Übersichtlichkeit ist es am besten, wenn wir die Schläge mit dem Schläger 9 in horizontaler und die Schläge mit dem Schläger 15 in vertikaler Richtung notieren. Spielen Sie analog „mathematisches Golf " mit den Schlägern 18 und 26 (18, 48; 38, 51)!

Wir stellen fest, dass wir mit den Schlägern 9 und 15 alle Vielfachen von 3 erreichen können, mit den Schlägen 18 und 26 alle Vielfachen von 2 und mit den Schlägern 18 und 48 alle Vielfachen von 6. Mit den Schlägern 38 und 51 können wir sogar jede Zahl erreichen. Suchen Sie sich selbst noch andere „Schlägerpaare" aus und ermitteln Sie jeweils die damit erreichbaren Zahlen. Vielleicht finden Sie schon bald die Antwort auf unsere Frage, welche diophantischen Gleichungen lösbar sind und welche nicht.

Diese Überlegungen können man vereinfachen, indem man einen solchen „Golfplatz" in einer Tabellenkalkulation anlegt. Man kann dann einfach und zügig viele beliebige Kombinationen durchprobieren und dadurch schnell zu den nötigen Einsichten gelangen!

Laden Sie die EXCEL-Datei `mathe_golf.xlsx` von Moodle und experimentieren Sie mit verschiedenen Zahlen. Untersuchen Sie auch den Aufbau dieses Tabellenblatts!

Damit kommen Sie schnell auf die aus der Zahlentheorie seit Euklid bekannte Erkenntnis, wonach das gemeinsame Maß zweier Strecken auch ein Maß für die Differenz der Strecken sein muss. Damit wissen wir, dass wir zur Lösung dieser Frage – „Welche Zahlen lassen sich beim mathematischen Golf erreichen?" – den ggT der beiden Zahlen ermitteln müssen. Es lassen sich dann dieser ggT und alle seine Vielfachen erreichen!

Wenden wir diesen Satz auf unser erstes Schlägerpaar 9 und 15 an:

Das *gemeinsame Maß* (der gemeinsame Teiler) von 9 und 15 muss auch ein Maß (ein Teiler) der Differenz von 9 und 15, also von 6 sein. Damit gilt auch: Das gemeinsame Maß von 9 und 6 muss auch ein Maß von 9 – 6 = 3 sein. Und weiter: Das gemeinsame Maß von 6 und 3 muss ein Maß von 6 – 3 = 3 sein. Damit liegt das „gemeinsame Maß" fest, es ist die Zahl 3.

Notieren wir diese Überlegungen formal, so können wir schreiben:

15 – 9 = 6 und 9 – 6 = 3 und 6 – 3 = 3 und 3 – 3 = 0

Durch fortwährende Differenzbildung erhalten wir schließlich den ggT zweier Zahlen. Wählt man andere Zahlen, kann diese fortgesetzte Differenzbildung allerdings recht langwierig werden:

150 – 27 = 123 123 – 27 = 96 96 – 27 = 69 69 – 27 = 42

42 – 27 = 15 27 – 15 = 12 15 – 12 = 3 12 – 3 = 9

9 – 3 = 6 6 – 3 = 3 3 – 3 = 0

Hier hilft die Erkenntnis, dass die Division eine fortgesetzte Subtraktion ist: Wie oft 27 in 150 enthalten ist, kann man herausbekommen, indem man 27 so oft von 150 subtrahiert bis es

„nicht mehr geht“ – oder eben, indem man 150 durch 27 dividiert. Und zwar genau so, wie es die Grundschüler machen, mit der Angabe des Rests:

$150 = 5 \cdot 27 + 15$

$27 = 1 \cdot 15 + 12$

$15 = 1 \cdot 12 + 3$

$12 = 4 \cdot 3 + 0$

Wir haben damit den *euklidischen Algorithmus* entwickelt und so nicht nur eine Antwort auf unsere Frage nach der Lösbarkeit von diophantischen Gleichungen, sondern auch bereits einen Ausblick auf die Ermittlung von Lösungen erhalten:

Die „linke Seite“ $x \cdot a + y \cdot b$ einer diophantischen Gleichung beschreibt genau unser „mathematisches Golfspiel“. Wir haben mit unserem „mathematischen Golf“ andererseits herausgefunden, dass wir damit den ggT von a und b und alle seine Vielfachen erreichen können. Anders ausgedrückt muss dann doch die „rechte Seite“ c einer diophantischen Gleichung ein Vielfaches des ggT von a und b sein, wenn die Gleichung lösbar sein soll.

Satz über die Lösbarkeit von diophantischen Gleichungen:

Eine diophantische Gleichung $x \cdot a + y \cdot b = c$ ist genau dann lösbar, wenn der ggT von a und b ein Teiler von c ist: $ggT(a, b) \mid c$

Zukünftig wissen wir also schon im Voraus, ob es sich überhaupt lohnt, bei einer diophantischen Gleichung nach Lösungen zu suchen. Karls Frau war dies auch bekannt, deshalb konnte sie sofort feststellen, dass ihr Mann unmöglich für Socken zu 6 € je Paar und T-Shirts zu 15 € je Stück insgesamt 227 € ausgegeben haben konnte!

Der euklidische Algorithmus hat uns nicht nur geholfen, die Frage nach der Lösbarkeit zu klären, er ist uns vielmehr auch bei der Ermittlung konkreter Lösungen behilflich. Wir wollen die Lösungsstrategie anhand einiger aufeinander aufbauender Beispiele entwickeln:

$x \cdot 48 + y \cdot 36 = 12$

Wenn wir mit Euklid den ggT bestimmen, dann erhalten wir:

$48 = 36 \cdot 1 + 12$

$36 = 12 \cdot 3 + 0$

Der ggT erscheint bereits in der ersten Zeile, diese kann umgeformt werden zu:

$12 = 1 \cdot 48 - 1 \cdot 36$

Als Lösungen erhalten wir $x = 1$ und $y = -1$. In diesem Fall entsprach der ggT von a und b gleich der rechten Seite c der diophantischen Gleichung.

Wir verändern jetzt unsere Gleichung ein wenig:

$x \cdot 48 + y \cdot 36 = 24$

Der euklidische Algorithmus bleibt derselbe:

$48 = 36 \cdot 1 + 12$, wir formen wieder um zu $12 = 1 \cdot 48 - 1 \cdot 36$

Da der ggT (hier 12) ein Teiler von c (hier 24) sein muss, brauchen wir die erhaltene Gleichung nur zu multiplizieren und erhalten:

$24 = 2 \cdot 48 - 2 \cdot 36$

Unsere Lösungen sind nun $x = 2$ und $y = -2$.

Lösen Sie selbst die diophantischen Gleichungen:

$x \cdot 8 + y \cdot 6 = 14$ $\quad$ $x \cdot 50 + y \cdot 49 = 78$ $\quad$ $x \cdot 14 + y \cdot 49 = 21$

Bisher hatte der euklidische Algorithmus bereits im ersten Schritt den ggT ermittelt. Somit konnte immer der jeweilige ggT von a und b durch einfache algebraische Umformungen als

„Linearkombination“ von *a* und *b* dargestellt werden. Der Begriff „Linearkombination“ stammt aus der linearen Algebra und wird dort im Zusammenhang mit Vektoren verwendet. Bei unserem „mathematischen Golf“ hatten wir solche Linearkombinationen konkret ermittelt: Jede „Schlagweite“ beim Golf stellte einen „Vektor“ dar und durch Kombinieren unterschiedlicher Vielfacher solcher Vektoren haben wir untersucht, welche Zahlen erreicht werden können. Diese Kombination unterschiedlicher Vielfacher kommt im Term $x \cdot a + y \cdot b$ zum Ausdruck.

Was ist nun, wenn der euklidische Algorithmus mehr als 2 Schritte umfasst? In solchen Fällen verwendet man vorteilhaft den *Berlekamp*-Algorithmus, der uns die sonst nötige umständliche Rechnerei deutlich vereinfacht:

1. Überlegung: Beim euklidischen Algorithmus kommt es ja immer auf die Reste an. In seiner ersten, ursprünglichen Form des Algorithmus haben wir diese Reste durch fortgesetztes Subtrahieren erhalten. Sucht man beispielsweise den ggT von 98 und 44, so drückt die erste Zeile $98 = 2 \cdot 44 + 10$ des zugehörigen Algorithmus nichts anderes aus, als dass wir von 98 zweimal 44 subtrahiert haben, um dann den Rest 10 zu erhalten. Genau dieselbe Aussage wird auch durch die Linearkombination $10 = 98 - 2 \cdot 44$ ausgedrückt. Wir subtrahieren also in jeder Zeile immer die kleinere Zahl von der größeren Zahl; wie oft subtrahiert werden kann, gibt uns jeweils der Ganzzahlquotient *q* an.

2. Kniff: Beim *Berlekamp*-Algorithmus schreibt man neben jede Zeile des Euklidischen Algorithmus jeweils die Linearkombinationen der Reste. Was hindert uns, solche Linearkombinationen auch für die beiden Ausgangszahlen zu notieren und zwar in zwei „vorausgehende“ Zeilen der Erweiterung (blau dargestellte Zeilen):

Euklid:		Berlekamp:							
		98	=	1	· 98 +	0	·	44	(1)
		44	=	0	· 98 +	1	·	44	(2)
98	= **2** · 44 + 10	10	=	1	· 98 +	–2	·	44	(3)
44	= **4** · 10 + 4	4	=	–4	· 98 +	9	·	44	(4)
10	= **2** · 4 + 2	2	=	9	· 98 +	–20	·	44	(5)

Die Linearkombination des ersten Restes 10 ergibt sich dann aus den beiden darüberstehenden (blauen) Gleichungen (1) und (2), indem man die Gleichung (2) von der Gleichung (1) zweimal subtrahiert. Dass zweimal subtrahiert werden muss, sagt uns der entsprechende Ganzzahlquotient aus der ersten Zeile des euklidischen Algorithmus. Beim Subtrahieren der rechten Gleichungsseiten müssen wir bei diesem spaltenweisen Arbeiten nun nur noch auf die Koeffizienten achten!

Die Linearkombination für den Rest 4 in der nächsten Zeile (Gleichung (4)) erhalten wir, indem wir die dritte Gleichung viermal von der zweiten Gleichung subtrahieren und die letzte Zeile schließlich, indem wir die vierte Gleichung zweimal von der dritten subtrahieren. Der kombinierte Euklid-Berlekamp-Algorithmus liefert uns somit den ggT zweier Zahlen als Linearkombination eben dieser Zahlen.

Lösen Sie nun übungshalber die folgenden diophantischen Gleichungen mit Hilfe des BERLEKAMP-Algorithmus:

$x \cdot 70 + y \cdot 40 = 20$	Lösung: $-2 \cdot 70 + 4 \cdot 40 = 20$
$x \cdot 68 + y \cdot 40 = 20$	Lösung: $15 \cdot 68 - 25 \cdot 40 = 20$
$x \cdot 68 + y \cdot 38 = 20$	Lösung: $-50 \cdot 68 + 90 \cdot 38 = 20$
$x \cdot 68 + y \cdot 42 = 20$	Lösung: $-80 \cdot 68 + 130 \cdot 42 = 20$

Wir haben für diese diophantischen Gleichungen jeweils nur ein Lösungspaar angegeben, das mit Hilfe des BERLEKAMP-Algorithmus gefunden wurde. Andererseits wissen wir aus unseren seitherigen Überlegungen, dass es für diophantische Gleichungen mehrere Lösungen geben kann. Führen wir unseren Algorithmus nochmals an zwei Beispielen konkret durch und lösen zunächst die diophantische Gleichung

$x \cdot 213 + y \cdot 45 = 24$

Euklid:															
						213	=	1	· 213	+	0	· 45			
						45	=	0	· 213	+	1	· 45			
213	=	4	· 45	+	33	33	=	1	· 213	+	–4	· 45			
45	=	1	· 33	+	12	12	=	–1	· 213	+	5	· 45			
33	=	2	· 12	+	9	9	=	3	· 213	+	–14	· 45			
12	=	1	· 9	+	3	3	=	–4	· 213	+	19	· 45			

Am Ende des *Berlekamp*-Algorithmus erhalten Sie eine Linearkombination von 213 und 45 für deren ggT 3. Diese Gleichung muss nun noch mit 8 erweitert werden, um das Lösungspaar $x_0 = –32$ und $y_0 = 152$ für die gestellte diophantische Gleichung zu erhalten.

Lösen wir eine weitere Gleichung $x \cdot 153 + y \cdot 45 = 24$

Euklid:															
						153	=	1	· 153	+	0	· 45			
						45	=	0	· 153	+	1	· 45			
153	=	3	· 45	+	18	18	=	1	· 153	+	–3	· 45			
45	=	2	· 18	+	9	9	=	–2	· 153	+	7	· 45			

Wieder können wir den ggT von 153 und 54 als Linearkombination dieser beiden Vektoren („Golfschläger“) angeben. Dies leistet der BERLEKAMP-Algorithmus in jedem Fall. Damit ist aber noch nichts über die Lösbarkeit der diophantischen Gleichung ausgesagt, denn um ein Lösungspaar zu erhalten, muss die gefundene Linearkombination des ggT noch auf die eigentliche Gleichung erweitert werden. Dies kann in diesem Fall nicht durchgeführt werden, weil der ggT von 9 kein Teiler der Zahl 24 ist. Wir haben also nochmals und wiederholend den schon festgestellten Zusammenhang bestätigt, wonach eine diophantische Gleichung $x \cdot a + y \cdot b = c$ nur dann lösbar ist, wenn der ggT(a, b) ein Teiler von c ist.

Bis hierher sollten Sie alles soweit verstanden haben, dass Sie ...

- den Berlekamp-Algorithmus – wie oben mehrfach dargestellt – selbst von Hand ggf. mit Hilfe des bereits in EXCEL erstellten Euklidischen Algorithmus durchführen können.
- wissen, dass Sie am Ende des Berlekamp-Algorithmus die Linearkombination des ggT der beiden Zahlen a und b vorliegen haben.
- schließlich wissen, dass Sie diese Linearkombination ggf. erweitern müssen, um eine Lösung der Diophantischen Gleichung $ax + by = c$ zu erhalten.

Dieses Wissen brauchen Sie, um den Euklid-Berlekamp-Algorithmus in EXCEL so realisieren zu können, dass Sie ein Lösungspaar für eine gegebene Diophantische Gleichung erhalten. Blättern Sie vor zu der Stelle, an der es mit eben dieser Realisation in EXCEL weitergeht. Die hier nachfolgenden Erläuterungen zeigen, wie man ausgehend von einem gefundenen Lösungspaar die weiteren Lösungen finden kann. Sie können hierher zurückkommen, wenn Sie die eben angesprochene Aufgabe (Finden eines Lösungspaars) gelöst haben und sich tiefer mit der Thematik der Diophantischen Gleichungen beschäftigen wollen. Dies kann durchaus auch erst in einem weiteren Semester sein, wenn Sie die Veranstaltung Zahlentheorie hören!

Im Folgenden wollen wir uns damit beschäftigen, wie für eine grundsätzlich lösbare diophantische Gleichung, für die auch schon ein Lösungspaar vorliegt, weitere Lösungen gefunden werden können.

Finden Sie im Kopf alle Lösungen der Gleichung $x \cdot 4 + y \cdot 6 = 36$ mit positiven x und y und stellen Sie alle Lösungen in einer Tabelle dar, wie wir dies anfangs mit der Anzahl der Enten und Schafe gemacht haben:

x	…	-3	0	3	6	9	12	…
y	…	8	6	4	2	0	–2	…

Ein Lösungspaar der Gleichung $x \cdot 5 + y \cdot 3 = 13$ ist $x = 2$ und $y = 1$. Stellen Sie andere, benachbarte Lösungspaare ebenso in einer Tabelle dar:

x	…	-4	-1	2	5	8	11	…
y	…	11	6	1	–4	–9	–14	…

Wir stellen fest, dass wir in diesen Fällen x und y jeweils in gleichen „Schrittweiten“ vergrößern bzw. verkleinern: Im ersten Beispiel lassen wir x von links nach rechts um 3 wachsen und gleichzeitig y um 2 kleiner werden. Im zweiten Beispiel wächst x ebenfalls um 3, während y um 5 kleiner wird. Können Sie – insbesondere im 2. Beispiel – schon einen Zusammenhang erkennen?

Für einen ersten Überblick ist es hilfreich, uns wieder des mathematischen Golfspiels zu bedienen. Wir haben ja eine Tabelle erstellt, die in einem recht großen Ausschnitt alle Zahlen darstellt, die mit zwei vorgewählten Golfschlägern erreicht werden können. Außerdem wird das Auftreten einer ebenfalls frei wählbaren Zielzahl farbig markiert, so dass hierbei die zugrunde liegenden Zusammenhänge augenfällig werden. Experimentieren Sie mit verschiedenen „Schlägern“ und lassen Sie sich jeweils unterschiedliche Zielzahlen farbig markieren! Welche Zusammenhänge erkennen Sie?

Sehr schnell wird ein offensichtlicher Zusammenhang deutlich: Die „Orte“ einer gewählten Zielzahl liegen alle auf einer Geraden, wie dies hier exemplarisch für die Schläger 3 und 5 und die Zielzahl 13 aufgezeigt ist.

Dies ist nicht weiter verwunderlich, wir hatten ja eben schon festgestellt, dass sich die Werte der Lösungspaare jeweils in gleichen Schritten ändern. Von einem markierten Wert zum anderen kommen wir, indem wir jeweils 3 Schritte in horizontaler und 5 Schritte in vertikaler Richtung laufen. So ist aber eben die Geradensteigung definiert, alle markierten Werte liegen somit auf einer Geraden:

8	13	18	23	28	33	38	43	48	53	58	63	68	73
5	10	15	20	25	30	35	40	45	50	55	60	65	70
2	7	12	17	22	27	32	37	42	47	52	57	62	67
-1	4	9	14	19	24	29	34	39	44	49	54	59	64
-4	1	6	11	16	21	26	31	36	41	46	51	56	61
-7	-2	3	8	13	18	23	28	33	38	43	48	53	58
-10	-5	0	5	10	15	20	25	30	35	40	45	50	55
-13	-8	-3	2	7	12	17	22	27	32	37	42	47	52
-16	-11	-6	-1	4	9	14	19	24	29	34	39	44	49
-19	-14	-9	-4	1	6	11	16	21	26	31	36	41	46

-22	-17	-12	-7	-2	3	8	13	18	23	28	33	38	43
-25	-20	-15	-10	-5	0	5	10	15	20	25	30	35	40
-28	-23	-18	-13	-8	-3	2	7	12	17	22	27	32	37
-31	-26	-21	-16	-11	-6	-1	4	9	14	19	24	29	34
-34	-29	-24	-19	-14	-9	-4	1	6	11	16	21	26	31
-37	-32	-27	-22	-17	-12	-7	-2	3	8	13	18	23	28
-40	-35	-30	-25	-20	-15	-10	-5	0	5	10	15	20	25
-43	-38	-33	-28	-23	-18	-13	-8	-3	2	7	12	17	22
-46	-41	-36	-31	-26	-21	-16	-11	-6	-1	4	9	14	19
-49	-44	-39	-34	-29	-24	-19	-14	-9	-4	1	6	11	16
-52	-47	-42	-37	-32	-27	-22	-17	-12	-7	-2	3	8	13

Deutlich wird bereits, dass andere Zielzahlen ebenfalls auf einer Geraden mit einer bestimmten Steigung liegen!

Dieser einfache Zusammenhang ermöglicht es uns, schnell auf die Zusammenhänge zu schließen. Probieren wir zunächst weitere Golfschläger aus und untersuchen, auf welchen Wegen man von einer gegebenen Zielzahl zu allen weiteren gelangen kann:

Schläger x	2	5	9	4	7	3
Schläger y	7	9	5	1	3	7
Steigung horizontal	7	9	5	1	3	7
Steigung vertikal	2	5	9	4	7	3

Gehen wir nun für eine systematische Betrachtung auf unser zweites Eingangsbeispiel zurück: Wir erhalten offensichtlich ein weiteres Lösungspaar der Gleichung $x \cdot 5 + y \cdot 3 = 13$, wenn wir ausgehend vom „ersten“ Lösungspaar $x_0 = 2$ und $y_0 = 1$ zu x_0 3 addieren und von y_0 5 subtrahieren; 3 und 5 sind aber genau die Vektoren („Golfschläger“) unserer diophantischen Gleichung.

Überprüfen wir unsere Vermutung allgemein: Wenn einerseits x_0 und y_0 als auch x_1 und y_1 Lösungspaare der diophantischen Gleichung $x \cdot a + y \cdot b = c$ sind, so muss doch gelten:

$$x_1 \cdot a + y_1 \cdot b = c$$
$$x_0 \cdot a + y_0 \cdot b = c$$
$$(x_1 - x_0) \cdot a + (y_1 - y_0) \cdot b = 0$$

Eine Lösung der letzten Gleichung erhalten wir, wenn wir

$(x_1 - x_0) = b$ und $(y_1 - y_0) = -a$

setzen, da $(b) \cdot a + (-a) \cdot b = 0$

Ausgehend von einem Lösungspaar x_0 und y_0 erhalten wir ein weiteres Paar

$x_1 = x_0 + b$ und $y_1 = y_0 - a$

Ein weiteres Paar x_2 und y_2 findet man auf dieselbe Weise:

$x_2 = x_1 + b = x_0 + 2 \cdot b$ und $y_2 = y_1 - a = y_0 - 2 \cdot a$

Auf diese Weise kann man letztlich alle Lösungspaare aus einem Lösungspaar x_0 und y_0 folgendermaßen gewinnen:

$x_k = x_0 + k \cdot b$ und $y_k = y_0 - k \cdot a$

Diese Überlegung gilt nun nicht nur für unser zweites Eingangsbeispiel, sondern auch für das erste mit der Gleichung $x \cdot 4 + y \cdot 6 = 36$. Allerdings gibt es in diesem ersten Beispiel zusätzlich auch noch einige „Zwischenlösungen“. Untersuchen wir hier ebenfalls zunächst einige weitere Beispiele mit Hilfe des mathematischen Golfspiels:

Schläger x	4	24	27	25
Schläger y	6	16	21	20
Steigung horizontal	3	2	7	4
Steigung vertikal	2	3	9	5

Wir sehen, dass unsere zuvor aufgestellte Vermutung grundsätzlich auch für diese Lösungen gilt. Dies bedeutet, dass unser „Steigungsdreieck“ auch hier angewendet werden kann. Nun haben jedoch die Schrittweiten in horizontaler als auch in vertikaler Richtung – welche das „Steigungsdreieck“ definieren – auch noch gemeinsame Teiler. Durch diese gemeinsamen Teiler kann man die horizontale als auch die vertikale Schrittweite dividieren und ein geometrisch ähnliches Steigungsdreieck zu erhalten:

38	42	46	50	54	58	62	66	70	74	78	82	86	90	94
32	36	40	44	48	52	56	60	64	68	72	76	80	84	88
26	30	34	38	42	46	50	54	58	62	66	70	74	78	82
20	24	28	32	36	40	44	48	52	56	60	64	68	72	76
14	18	22	26	30	34	38	42	46	50	54	58	62	66	70
8	12	16	20	24	28	32	36	40	44	48	52	56	60	64
2	6	10	14	18	22	26	30	34	38	42	46	50	54	58
-4	0	4	8	12	16	20	24	28	32	36	40	44	48	52
-10	-6	-2	2	6	10	14	18	22	26	30	34	38	42	46
-16	-12	-8	-4	0	4	8	12	16	20	24	28	32	36	40

Von einer Zielzahl zu einer anderen kommt man bei der oben dargestellten diophantischen Gleichung $x \cdot 4 + y \cdot 6 = 36$ auch, indem man 6 Schritte in x- und 4 Schritte in y-Richtung läuft. Dieses Steigungsdreieck lässt sich aber durch dividieren seiner Kathetenlängen durch 2 in ein dazu ähnliches Überführen.

Um das **kleinst**mögliche Steigungsdreieck zu erhalten, muss man durch den **größten** gemeinsamen Teiler dividieren!

Unsere „Lösungsformel“ leiten wir wieder aus der Gleichung $(x_1 - x_0) \cdot a + (y_1 - y_0) \cdot b = 0$ ab, indem wir diese durch $g = \text{ggT}(a, b)$ dividieren. Da wir jeweils durch den ggT dividieren, bleiben alle Terme ganzzahlig:

$$\left(x_1 - x_0\right) \cdot \frac{a}{g} + \left(y_1 - y_0\right) \cdot \frac{b}{g} = 0$$

Eine Lösung dieser Gleichung erhalten wir wie oben, indem wir

$x_1 - x_0 = \frac{b}{g}$ und $y_1 - y_0 = -\frac{a}{g}$ ersetzen. Damit ergibt sich für

$x_1 = x_0 + \frac{b}{g}$ und $y_1 = y_0 - \frac{a}{g}$.

Alle weiteren Lösungspaare x_k und y_k finden wir also mit den Formeln:

$x_k = x_0 + k \cdot \frac{b}{g}$ und $y_k = y_0 - k \cdot \frac{a}{g}$.

Diese Beziehungen für x_k und y_k gelten selbstverständlich auch für den zuerst dargestellten Sonderfall teilerfremder Vektoren: Für teilerfremde Vektoren ist deren ggT = 1 und damit gehen die eben dargestellten Beziehungen in die des zuvor gefundenen Sonderfalls über.

Damit können wir – ausgehend von einem „ersten“ gefundenen Lösungspaar x_0 und y_0 – unendlich viele weitere Lösungspaare x_k und y_k angeben. Man beachte, dass hierbei k ebenfalls der Menge der ganzen Zahlen entstammt und damit auch negativ sein kann.

Suchen wir anhand dieser Erkenntnisse nun für eine weitere diophantische Gleichung ein erstes Lösungspaar x_0, y_0 und dann auch noch weitere Lösungspaare.

$x \cdot 323 + y \cdot 119 = 34$

					323	= 1	· 323	+ 0	· 119
Euklid:					119	= 0	· 323	+ 1	· 119
323	= 2	· 119	+ 85		85	= 1	· 323	+ –2	· 119
119	= 1	· 85	+ 34		34	= –1	· 323	+ 3	· 119
85	= 2	· 34	+ 17		17	= 3	· 323	+ –8	· 119

Durch Erweitern der gefundenen Linearkombination erhalten wir ein erstes Lösungspaar $x_0 = 6$ und $y_0 = –16$. Mit Hilfe der obigen Formeln finden wir weitere Lösungspaare:

$x_k = 6 + k \cdot \frac{119}{17}$ und $y_k = -16 - k \cdot \frac{323}{17}$ bzw. $x_k = 6 + k \cdot 7$ und $y_k = -16 - k \cdot 19$.

Diese lauten beispielsweise:

k	x_k	y_k
…	…	…
-4	-22	60
-3	-15	41
-2	-8	22
-1	-1	3
0	6	-16
1	13	-35
2	20	-54
3	27	-73
…	…	…

Geben Sie für die im vorigen Abschnitt bearbeiteten diophantischen Gleichungen

$x \cdot 70 + y \cdot 40 = 20$	Lösung: –2 , 4 weitere:	(–6,11), (–10, 18), (2, –3), (6, –10)
$x \cdot 68 + y \cdot 40 = 20$	Lösung: 15, –25	(5, –8), (–5, 9), (25, –42), (35, –59)
$x \cdot 68 + y \cdot 38 = 20$	Lösung: –50, 90	(–69, 124), (–88, 158), (–31, 56), (–12, 22)
$x \cdot 68 + y \cdot 42 = 20$	Lösung: –80, 130	(–59, 130, (–38, 62), (–17, 28), (4, –6)

ebenfalls weitere Lösungspaare an.

Die Vorgehensweise für das Lösen einer diophantischen Gleichung soll abschließend nochmals kurz wiederholt und zusammengestellt werden:

1. Feststellen der Lösbarkeit überhaupt:

Eine diophantische Gleichung $x \cdot a + y \cdot b = c$ ist genau dann lösbar, wenn der ggT von a und b ein Teiler von c ist: $ggT(a, b) \mid c$

2. Bestimmen der Linearkombination des ggT von a und b mit Hilfe des EUKLID-BERLEKAMP-Algorithmus. Damit erhalten wir eine Gleichung der Form:

$x' \cdot a + y' \cdot b = g$ mit: $g = ggT(a, b)$

3. Erweitern der erhaltenen Linearkombination

Da nach der Voraussetzung für lösbare diophantische Gleichungen g ein Teiler von c sein muss, kann die erhaltene Linearkombination immer mit $\frac{c}{g}$ erweitert werden und man erhält eine Lösung der diophantischen Gleichung:

$x_0 \cdot a + y_0 \cdot b = c$

4. Ermitteln weiterer Lösungen

Weitere Lösungspaare x_k und y_k erhält man über die Beziehungen:

$x_k = x_0 + k \cdot \frac{b}{g}$ und $y_k = y_0 - k \cdot \frac{a}{g}$. Dabei ist k ∈ Z.

5. Plausibilitätsüberprüfung

Für eine aus einem Sachkontext ermittelte diophantische Gleichung muss ggf. über weitere Randbedingungen die Lösung eingeschränkt werden. So sind bei der einführenden Aufgabe mit den Tieren selbstverständlich nur positive Werte für die Anzahl der Enten und Schafe sinnvoll.

1.5.3.2 Realisation in EXCEL

Im Folgenden wird es darum gehen, den bereits von Ihnen in ECXEL erstellten Euklidischen Algorithmus mit dem Berlekamp-Algorithmus so zu erweitern, dass ein Lösungspaar einer gegebenen Diophantischen Gleichung ermittelt wird.

Laden Sie daher zunächst Ihre Version des Euklidischen Algorithmus und erweitern Sie die erste Zeile nach dem untenstehenden Muster so, dass die gegebene Diophantische Gleichung eingegeben werden kann.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1		98	*	x	+	44	*	y	=	12								
2																		
3										**Berlekamp:**								
4										98	=	1	x	98	+	0	x	44
5		**Euklid:**								44	=	0	x	98	+	1	x	44
6		98	=	2	x	44	+	10		10	=	1	x	98	+	-2	x	44
7		44	=	4	x	10	+	4		4	=	-4	x	98	+	9	x	44
8	1	10	=	2	x	4	+	2		2	=	9	x	98	+	-20	x	44
9		4	=	2	x	2	+	0		0	=	-22	x	98	+	49	x	44

In den Zeilen `4` und `5` des Tabellenblatts sollen in den Spalten `J` bis `R` die Linearkombinationen der Zahlen `a` und `b` (im Beispiel 98 und 44) erscheinen. Die Zahlen 98 und 44 schreiben Sie als Bezüge auf `B1` und `F1`, die Faktoren 0 und 1 werden als Konstante eingetragen.

Nun muss die Linearkombination des ersten Rests 10 bestimmt werden. Den in `J6` nochmals erscheinenden Rest übernehmen Sie einfach aus `H6`.

Die Zahlen 98 und 44 übernehmen Sie entweder aus der darüberstehenden Zeile oder (mit festen Bezügen) aus den gelben Eingabefeldern.

Berechnet werden müssen lediglich die Koeffizienten in den Spalten `L` und `P`. Aus den fachlichen Grundlagen wissen Sie, dass die dritte Berlekamp-Gleichung entsteht, indem man die zweite Gleichung so oft von der ersten Gleichung subtrahiert, wie es der Ganzzahlquotient der zugehörigen Zeile des euklidischen Algorithmus angibt. Dieser Ganzzahlquotient steht in Spal-

te D. Mit diesen Angaben sollten Sie es schaffen, die benötigten einfachen Formeln in die Felder L6 und P6 einzutragen.

Haben Sie die gesamte dritte Zeile des Berlekamp-Algorithmus aufgebaut, so markieren Sie die zugehörigen Zellen und kopieren diese bis einschließlich Zeile 15 nach unten. Sind Ihre Formeln korrekt, so sollten Sie genau die oben dargestellten Werte bekommen.

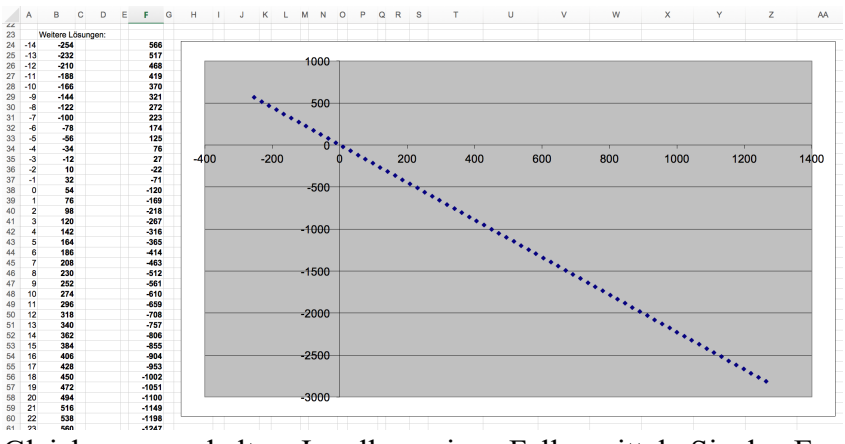
Die ab Zeile 10 erscheinenden obskuren Meldungen können Sie später(!) mit Hilfe des in der letzten Veranstaltung dargestellten Verfahrens ausblenden. Zunächst wollen wir uns darum kümmern, ein Lösungspaar der gegebenen Diophantischen Gleichung zu ermitteln.

Es ist leicht zu *sehen*, dass in Zeile 8 Euklid den ggT von 98 und 44 liefert und Berlekamp diesen ggT als Linearkombination von 98 und 44 darstellt. Dass wir dies *sehen* nützt für die Weiterarbeit nichts, wir müssen genau diese Zeile, in welcher Euklid den ggT liefert, eindeutig markieren. Hierfür wird die seither noch freie Spalte A verwendet: Dort soll genau in der Zeile, in welcher der ggT erscheint, eine 1 erscheinen, in allen anderen Zeilen soll *nichts* erscheinen (s.o.). Es liegt nahe, für diese Aufgabe die WENN()-Funktion zu verwenden: WENN in dieser Zeile der ggT steht, DANN schreibe eine 1 SONST nichts. Jetzt müssen Sie nur noch überlegen, wie der ggT erkannt werden kann und diese Bedingung in die WENN()-Funktion eintragen. Die gesamte Formel kopieren Sie dann in der A-Spalte von Zeile 6 bis einschließlich Zeile 14. Haben Sie eine Antwort darauf, warum diese Formel nur bis Zeile 14 kopiert wird?

Um ein Lösungspaar der gegebenen Diophantischen Gleichung darstellen zu können, ist es wohl nicht notwendig aber doch hilfreich, die Darstellung des ggT als Linearkombination in Zeile 18 nochmals aufzuführen:

	A	B	C	D	E	F	G	H	I	J
16										
17		Darstellung des ggT als Linearkombination:								
18		**9**	x	98	+	**-20**	x	44	=	2
19										
20		Ein spezielles Lösungspaar der Ausgangsgleichung:								
21		**54**	x	98	+	**-120**	x	44	=	12

Die Zahlen 98 und 44 dort erscheinen zu lassen ist einfach, doch woher bekommen wir deren Koeffizienten 9 und –20? Nun, diese stehen doch genau in der mit der 1 markierten Zeile, so dass wir mit der ebenfalls schon bekannten SVERWEIS()-Funktion darauf zugreifen können. Das Suchkriterium ist die 1, als Matrix geben Sie den Bereich von A6 bis R14 an, den Zugriff auf den entsprechenden Koeffizienten erhalten Sie über den Spaltenindex, indem Sie die Spalten von A bis Spalte L bzw. P zählen.



Wenn Sie nun diese Gleichung (ggT als Linearkombination) mit der eigentlich zu lösenden Diophantischen Gleichung in Zeile 1 vergleichen, dann sehen Sie unschwer, dass Sie (im gegebenen Beispiel) die Gleichung der Linearkombination des ggT nur mit 6 erweitern müssen, um eine Lösung der Diophantischen Gleichung zu erhalten. Im allgemeinen Fall ermitteln Sie den Erweiterungsfaktor, indem Sie die Zahl c der Diophantischen Gleichung furch den ermittelten ggT dividieren. Damit können

Sie in Zeile 21 eine spezielle Lösung der gegebenen Diophantischen Gleichung darstellen lassen.

Sind Sie soweit gediehen können Sei darangehen, die obskuren Meldungen der letzten Zeilen des Berlekamp-Algorithmus zu eliminieren.

Haben Sie auch dies erledigt, so könnten Sie überlegen wie es Ihnen gelingt, eine Tabelle weiterer Lösungspaare in EXCEL darzustellen und vielleicht auch eine Graphik erstellen, in welcher diese Lösungspaare dargestellt sind.

2 Dynamische Geometriesysteme

Es wundert kaum, dass es für den Computer eine große Anzahl an Programmen zum Rechnen und Berechnen gibt. Auch die Anwendung solcher Programme in Schule und Hochschule ist nur konsequent. Einen wirklichen Meilenstein für die Anwendung des Computers in der Mathematikausbildung stellen jedoch dynamische Geometriesysteme dar, die erst seit vergleichsweise kurzer Zeit erhältlich sind. Und während die genannten Rechenprogramme ausschließlich für kommerzielle Anwendungen in der Industrie und der Wirtschaft entwickelt wurden, nehmen Geometrieprogramme ihren fast ausschließlichen Selbstzweck in der Mathematikausbildung ein. Mit ihnen ist es erstmals möglich, geometrische Relationen dynamisch herauszuarbeiten und die Geometrie damit aus den seitherigen starren Papier-und-Bleistift-Konstruktionen herauszulösen. Damit können funktionale Zusammenhänge ungleich besser und augenfälliger dargestellt und Abhängigkeiten eindrucksvoll demonstriert werden.

Inzwischen gibt es eine Vielzahl an dynamischen Geometriesystemen, die alle ihre Daseinsberechtigung im Hinblick auf einen didaktischen Einsatz an der Schule haben. Dass ich selbst Cinderella verwende, hat drei Gründe:

Der erste Grund liegt in dem enormen Funktionsumfang von Cinderella begründet. Cinderella bietet nicht nur dynamisches Konstruieren in der euklidischen Ebene, sondern darüber hinaus auch Konstruktionen in der hyperbolischen und elliptischen Geometrie sowie die Möglichkeit von physikalischen Simulationen. Mit Makros und einer mächtigen Scriptsprache gelingt es problemlos, Cinderella an seine persönlichen Vorlieben und Bedürfnisse anzupassen.

Cinderella ist kostenlos und läuft auf allen gängigen Plattformen: Es ist in der Windowswelt ebenso zuhause wie auf Macs und auf Unix-Rechnern. Damit ist auch ein Arbeiten in heterogenen Umgebungen problemlos möglich.

Zu guter Letzt ist eine persönliche Verbundenheit zum Entwickler Ulli Kortenkamp sehr hilfreich bei all den kleinen Fallstricken, die im Umgang mit Softwaresystemen lauern. Außerdem sorgt sie für eine schnelle Behebung von entdeckten Bugs.

Wir werden in dieser Veranstaltung die grundlegenden Eigenschaften und Möglichkeiten eines dynamischen Geometriesystems am Beispiel Cinderella kennen lernen. Diese Erfahrungen werden Sie später nötigenfalls leicht auf andere Systeme übertragen können.

Sie sollten vorbereitend Cinderella auf Ihrem Rechner installiert und schon einige eigene Versuche damit angestellt haben. Hinweise zur Installation und eine Anleitung für Ihre ersten Schritte finden Sie unter

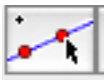
http://www.albrecht57.de/index.php/ph-gd/ph-mathematik/computereinsatz

2.1 Besondere Punkte im Dreieck

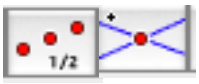
Als „besondere Punkte“ des Dreiecks wollen wir hier den Höhenschnittpunkt *H*, den Schnittpunkt der Mittelsenkrechten *M* sowie den Schwerpunkt *S* und den Schnittpunkt der Winkelhalbierenden *W* verstehen. Zunächst rekapitulieren Sie die Inhalte der Mittelstufengeometrie – oder lesen Sie diese nach – und geben an, wie der Schwerpunkt *S* eines Dreiecks konstruiert wird und erklären, wozu *M* und *W* nützlich sind.

2.1.1 Konstruktion der besonderen Punkte

Setzen Sie auf der Zeichenfläche von Cinderella drei Punkte *A*, *B* und *C* ab und definieren Sie das Dreieck als Polygon.



Um die besonderen Punkte konstruieren zu können, müssen Sie außerdem die drei Trägergeraden der Dreiecksseiten durch je zwei Eckpunkte legen.



Erstellen Sie die Seitenmitten aller Dreiecksseiten und verbinden Sie jede Seitenmitte mit der gegenüberliegenden Ecke durch eine Gerade. Bringen Sie zwei Geraden zum Schnitt und nennen Sie den entstandenen Schnittpunkt *S*.

Errichten Sie auf den Seitenmitten die Senkrechten zur jeweiligen Seite, bringen zwei dieser Senkrechten zum Schnitt und nennen den erhaltenen Schnittpunkt *M*.

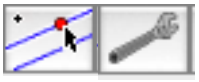
Fällen Sie die Lote von den Dreieckspunkten auf die jeweils gegenüberliegende Seite, schneiden Sie zwei dieser Lote und nennen Sie den Schnittpunkt *H*.

Lassen Sie schließlich zwei Winkelhalbierende zeichnen, schneiden Sie diese und nennen Sie den Schnittpunkt *W*.

Machen Sie die Hilfsgeraden (Winkelhalbierende, Seitenhalbierende, Höhen und Mittelsenkrechten) unsichtbar, um den Blick für das Wesentliche zu behalten.

2.1.2 Mögliche Orte der besonderen Punkte

Die genaue Lage der besonderen Punkte hängt von der Form des Dreiecks ab. Welche der besonderen Punkte liegen immer innerhalb des Dreiecks und welche können auch außerhalb liegen? Stellen Sie erst Vermutungen an und überprüfen Sie diese hernach durch Verändern der Eckpunkte des Dreiecks. Versuchen Sie auch, Ihre Vermutungen zu begründen!



Erstellen Sie oberhalb von *AB* eine zu *AB* parallele Gerade und befestigen Sie den Punkt *C* auf dieser Geraden, so dass *C* nur parallel zu *AB* verschoben werden kann. Achtung: Legen Sie die Parallele zu *AB* **nicht** durch *C* sondern außerhalb von *C* und befestigen hinterher *C* auf der Parallelen!

Auf welchen Bahnen bewegen sich *H*, *M*, *S* und *W*, wenn *C* parallel zu *AB* verschoben wird? Stellen Sie erst Vermutungen an und überprüfen Sie diese erst hernach durch Verschieben von *C*.

Cinderella bietet die Möglichkeit, diese Bahnkurven darzustellen. Der Fachterminus lautet *Ortskurve* – sie ist die Menge aller Orte, die ein bestimmter Punkt annehmen kann, wenn das gesamte System definiert verändert wird.

Lassen Sie zuerst die Ortskurve von *H* aufzeichnen, wenn *C* auf der Parallelen zu *AB* verschoben wird. Nach der Auswahl der zugehörigen Befehlsschaltfläche müssen Sie zuerst das bewegende Element (den Punkt *C*) anklicken und danach denjenigen Punkt, dessen Ortskurve Sie erzeugen wollen, in unserem Fall den Punkt *H*.

Löschen Sie die erzeugte Ortskurve jeweils wieder, bevor Sie auch die zu *M*, *S* und *W* gehörigen Ortskurven zeichnen lassen.

Lösen Sie *C* von der Parallelen durch *AB* und löschen Sie diese Parallele. Zeichnen Sie stattdessen einen Kreis und befestigen Sie *C* auf der Kreislinie. Achtung:

Sie sollen den Kreis **nicht** um *C* zeichnen, sondern völlig unabhängig von *C*. Befestigen Sie erst danach *C* auf der Kreislinie!

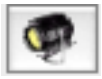
Auf welchen Bahnen verlaufen *H*, *M*, *S* und *W*, wenn *C* auf der Kreislinie bewegt wird? Stellen Sie zunächst wieder Vermutungen an und überprüfen Sie diese, indem Sie Ortskurven zeichnen lassen. Verändern Sie dabei auch die Lage des Dreiecks und die Lage sowie die Größe des Kreises – es entstehen interessante Ortslinien!

2.1.3 Die Eulergerade

Entfernen Sie *C* vom Kreis und löschen Sie diesen. Bewegen Sie *C* nun freihändig und achten Sie dabei auf die Lage der Punkte *H*, *M* und *S*. Welchen Eindruck haben Sie? Legen Sie durch *H* und *M* eine Gerade und beobachten Sie genau!

Satz von der *Eulergeraden*: In jedem Dreieck liegen der Höhenschnittpunkt, der Schwerpunkt und der Schnittpunkt der Mittelsenkrechten auf einer Geraden.

Verbinden Sie die Seitenmittelpunkte über ein Polygon, wählen Sie als Füllfarbe Gelb und stellen Sie die Füllsichtbarkeit auf einen etwas niedrigen Wert. Das gelbe kleine Dreieck ist das *Mittendreieck* des großen Dreiecks ABC.



Machen Sie die Mittelsenkrechten des Dreiecks *ABC* wieder sichtbar.

Färben Sie diese Höhen schwarz und stellen Sie diese etwas dicker dar. In welcher Beziehung stehen diese dicken schwarzen Geraden zum Mittendreieck? Was folgt daraus für den Schnittpunkt *M* der Mittelsenkrechten im großen Dreieck und dem Schnittpunkt *H* der Höhen im Mittendreieck?

$$M_{ABC} = H_{\text{Mittendreieck}} \qquad (1)$$

Machen Sie die Seitenhalbierenden des Dreiecks *ABC* wieder sichtbar. Färben Sie diese Seitenhalbierenden rot ein und stellen Sie diese etwas dicker dar. Durch welche konkrete Abbildung gelingt es, das Dreieck *ABC* (und jedes beliebige Dreieck) auf das zugehörige Mittendreieck abzubilden?

$$Z(S, -1/2)$$

Die zentrische Streckung $Z(S, -1/2)$ bildet jedes beliebige Dreieck auf sein Mittendreieck ab. (Umgekehrt wird durch $Z(S,-2)$ ein Mittendreieck auf sein zugehöriges Umdreieck abgebildet. Schauen Sie genau hin!

Bei der zentrischen Streckung wird der Schnittpunkt der Mittelsenkrechten des großen Dreiecks M_{ABC} auf den Schnittpunkt der Mittelsenkrechten des Mittendreiecks $M_{\text{Mittendreieck}}$ abgebildet. Dies folgt u.a. aus der Streckenverhältnistreue der zentrischen Streckung:

$$M_{ABC} \xrightarrow{Z\left(S,-\frac{1}{2}\right)} M_{Mittendreieck} \qquad (2)$$

Ersetzt man M_{ABC} durch $H_{\text{Mittendreieck}}$ (s. (1)), so erhält man:

$$H_{Mittendreieck} \xrightarrow{Z\left(S,-\frac{1}{2}\right)} M_{Mittendreieck}$$

$H_{\text{Mittendreieck}} \rightarrow M_{\text{Mittendreieck}}$

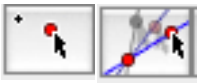
Der Höhenschnittpunkt *H* eines Dreiecks wird durch die angegebene zentrische Streckung um *S* auf den Schnittpunkt *M* der Mittelsenkrechten abgebildet.

Nach den Gesetzen der Abbildung *Z* sind Zentrum, Urbildpunkt und Bildpunkt kollinear, somit gilt: $H_{\text{Mittendreieck}}$, *S* und $M_{\text{Mittendreieck}}$ sind kollinear! Und da jedes beliebige Dreieck ein Mittendreieck ist, gilt der Satz für die Eulergerade für alle beliebigen Dreiecke.

2.2 Der Feuerbach- oder Neun-Punkte-Kreis

Der Feuerbachkreis des Dreiecks *ABC* mit Höhenschnittpunkt *H*, ist der Kreis, der durch die Höhenfußpunkte H_a, H_b, H_c, die Seitenmittelpunkte M_a, M_b, M_c und durch die Mitten der Strecken *HA*, *HB* und *HC* verläuft. Diese letztgenannten Punkte nennt man auch Euler-Punkte, sie werden daher mit E_a, E_b und E_c bezeichnet. Um den Beweis dieses Satzes wollen wir uns hier nicht kümmern, sondern diesen Kreis und seine Eigenschaften als Anlass nehmen, grundlegende Konstruktionsaufgaben mit Cinderella zu erledigen.[3]

2.2.1 Konstruktion des Kreises

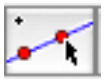


Wählen Sie die Schaltfläche zum Erzeugen von Punkten und setzen Sie drei Punkte durch Klicken auf die Arbeitsfläche ab, diese werden fortfolgend mit *A*, *B* und *C* benannt. Sollten Sie mit der Anordnung nicht zufrieden sein, so können Sie diese nach dem Auswählen der Bewegen-Schaltfläche jederzeit ändern.

Dass es sich um ein Dreieck handelt können Sie hervorheben, indem Sie die drei Punkte als Polygon definieren. Die Füllfarbe und die Füllsichtbarkeit können Sie ebenfalls im Eigenschaftsfenster verändern.

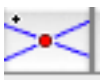
2.2.1.1 Höhenfußpunkte

Für die Konstruktion des Neunpunktekreises benötigen wir die Höhen des Dreiecks und die Höhenfußpunkte. Letztere liegen bei einem stumpfwinkligen Dreieck außerhalb der Seitenkanten. Damit wir die Höhenschnittpunkte auch in solchen Fällen verlässlich erhalten, dürfen wir die Höhenfußpunkte nicht als Schnittpunkte der Senkrechten durch einen Dreieckspunkt auf die gegenüberliegende Seite definieren, sondern müssen zuvor durch die Dreieckspunkte noch Geraden legen. Diese Geraden stellen wir mit Hilfe des Eigenschaftsfensters gestrichelt dar.



Die Höhen erzeugen wir danach als Senkrechte auf die gegenüberliegende Trägergerade durch jeweils einen Dreieckspunkt.

Die Höhenfußpunkte erhalten wir als die Schnittpunkte der Höhen mit den Trägergeraden. Es mutet anfangs etwas fremd an, aber Schnittpunkte müssen in dynamischen Geometriesystemen explizit als solche erzeugt werden.



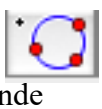
Für die Namen der Höhenfußpunkte werden automatisch Namen vergeben; im Eigenschaftsfenster ändern Sie diese in *Ha*, *Hb* und *Hc* ab. Erzeugen Sie schließlich noch den Höhenschnittpunkt *H* als Schnittpunkt zweier Höhen. Die Höhen können Sie – falls gewünscht – im Eigenschaftsfenster unsichtbar schalten.

2.2.1.2 Seitenmitten

Der Neunpunktekreis verläuft weiter durch die Seitenmitten des Dreiecks. Erzeugen Sie diese Mittelunkte mit der zugehörigen Schaltfläche und bezeichnen Sie diese mit *Ma*, *Mb* und *Mc*.

2.2.1.3 Eulerpunkte

Schließlich benötigen wir noch die Streckenmitten der Strecken *HA*, *HB* und *HC*, die so genannten Eulerpunkte *Ea*, *Eb* und *Ec*. Erzeugen und benennen Sie diese entsprechend.



Nun sind die 9 Punkte konstruiert, so dass Sie den Feuerbachkreis zeichnen können. Ein Kreis ist bereits eindeutig durch drei Punkte festgelegt, klicken Sie daher die zugehörige Schaltfläche an und wählen drei der neun erzeugten Punkte aus. Der erscheinende Kreis verläuft offensichtlich auch durch die anderen sechs Punkte.

---

[3] Den Beweis zum Kreis und seinen Eigenschaften finden Sie unter http://www.math.uni-bremen.de/didaktik/ma/ralbers/Materialien/Dreieckszentren/X5_Feuerbach/X5_Feuerbach.html

2.2.2 Eigenschaften des Kreises

2.2.2.1 Satz von Feuerbach

Von Feuerbach selbst stammt der nach ihm benannte „große Satz“: *Der Feuerbachkreis eines Dreiecks ABC berührt stets den Inkreis und die drei Ankreise und zwar den Inkreis innerlich und die drei Ankreise äußerlich.*

Bekannt ist sicher, dass der Inkreismittelpunkt der Schnittpunkt der Winkelhalbierenden des Dreiecks ist. Da sich alle drei Winkelhalbierenden in einem Punkt schneiden, reicht die Konstruktion zweier Winkelhalbierender aus.

Den Radius des Inkreises findet man, indem man das Lot von seinem Mittelpunkt auf eine Seite fällt.

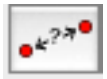
Neu dürfte sein, dass ein Dreieck auch *Ankreise* hat. Ein Ankreis eines Dreiecks berührt eine Seite des Dreiecks und die Trägergeraden der beiden anderen Seiten. Sein Mittelpunkt ist der Schnittpunkt der Winkelhalbierenden der beiden zugeordneten Außenwinkel. Sein Radius ist der Abstand seines Mittelpunkts zu einer der Trägergeraden.

Achten Sie bei diesen nun doch recht umfangreichen Konstruktionen darauf, dass zur Konstruktion benötigte Hilfslinien – wie beispielsweise die Winkelhalbierenden der Außenwinkel – nach der Konstruktion des Mittelpunkts unsichtbar geschaltet werden. Sie verlieren sonst leicht den Überblick!

2.2.2.2 Radiusvergleich

Der Radius des Umkreises ist doppelt so groß wie derjenige des Feuerbachkreises. Diesen Satz werden wir verifizieren, indem wir beide Radien messen. Dazu muss zuerst der Umkreis des Dreiecks konstruiert werden. Sein Mittelpunkt ist der Schnittpunkt der Mittelsenkrechten des Dreiecks. Die Mittelpunkte der Dreiecksseiten sind schon vorhanden, es müssen somit nur noch die Senkrechten zu den Seiten durch diese Mittelpunkte konstruiert werden.

Da der Neunpunktekreis durch drei seiner Punkte gezeichnet worden ist, müssen wir auch noch dessen Mittelpunkt konstruieren. Dies machen wir klassisch über den Schnittpunkt der Mittelsenkrechten zweier Sehnen.



Nun können wir messen: Vom Mittelpunkt des Umkreises zu einem Eckpunkt des Dreiecks und vom Mittelpunkt des Feuerbachkreises zu einem seiner definierenden neun Punkte.

Tatsächlich ermittelt Cinderella das angesprochene Streckenverhältnis unabhängig von der Form des Dreiecks – dies ist jedoch kein mathematischer Beweis! Dass es sich tatsächlich um die doppelte Streckenlänge handelt, können wir von Cinderella anzeigen lassen: Dazu wählen wir die Schaltfläche BERECHNUNG EINES WERTS und klicken in einen freien Bereich des Zeichnungsfensters. In das sich öffnende Fenster schreiben wir das Streckenverhältnis, indem wir zuerst auf den angezeigten Abstand MC klicken (Cinderella schreibt `dist0`), dann den Bruchstrich „/“ eingeben und schließlich auf den angezeigten Abstand der Entfernung FM_a (Cinderella schreibt `dist1`). Die Eingabe beenden Sie mit dem Klicken auf ERGEBNIS. Das Fenster verschwindet und an seiner Stelle steht nun der Wert 2, der sich auch nicht verändert, wenn die Form des Dreiecks variiert wird.

2.2.2.3 Halbierungseigenschaft

Der Feuerbachkreis halbiert die Strecke zwischen Höhenschnittpunkt H und jedem beliebigen Punkt des Umkreises. Die Gültigkeit dieses Satzes können wir ebenfalls durch einfaches Messen überprüfen, allerdings damit nicht beweisen! Dazu befestigen wir einen Punkt *U* auf dem Umkreis und zeichnen die Stecke *HU*. Diese Strecke schneiden wir im Punkt *N* mit dem Feuerbachkreis. Von beiden Strecken *HN* und *HU* lassen wir uns deren Längen anzeigen und können schließlich wie oben das Verhältnis *HN* zu *HU* berechnen lassen.

Nun stehen zwei Zahlenwerte auf der Zeichenfläche und zu deren Unterscheidung sollten Sie über die Schaltfläche BESCHRIFTUNG ÄNDERN/ANBRINGEN einen jeweils erläuternden Text anbringen.

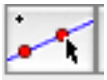
2.2.2.4 Eulergerade

Die letzte Eigenschaft bringt noch die *Eulergerade* ins Spiel: *Der Mittelpunkt F des Feuerbach-Kreises liegt stets auf der Eulergeraden und ist Mittelpunkt der Strecke HM (H Höhenschnittpunkt, M Mittelpunkt des Umkreises).*

Die letztgenannte Eigenschaft können wir durch Längenmessungen und eine Verhältnisbildung wie schon zweimal durchgeführt einfach überprüfen.

Unter der Eulergeraden eines Dreiecks *ABC* versteht man die Gerade, die durch den Schwerpunkt *S*, den Umkreismittelpunkt *M* und durch den Höhenschnittpunkt *H* des Dreiecks geht. Um die Eulergerade zeichnen zu können, benötigen wir noch den Schwerpunkt des Dreiecks als Schnittpunkt der Seitenhalbierenden.

Wir legen schließlich die Eulergerade durch *H* und *M* und sehen, dass auch *S* und der Mittelpunkt *F* des Feuerbachkreises auf dieser Geraden liegen.



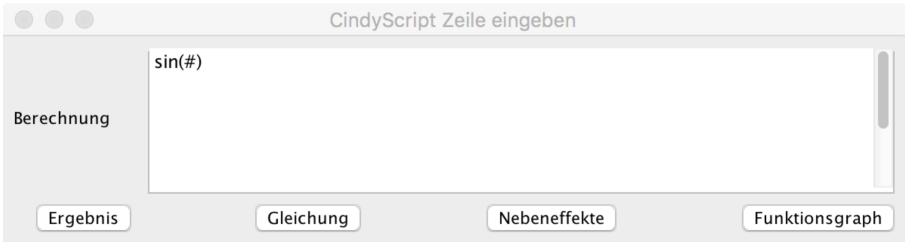
2.3 Darstellung von Funktionsgraphen

Neben den klassischen Konstruktionsaufgaben kann Cinderella Funktionen plotten, das Newton-Verfahren veranschaulichen und gar eine Kurvendiskussion durchführen.

2.3.1 Cinderella als Funktionsplotter

Um Cinderella als Funktionsplotter zu verwenden, ist es sinnvoll, zunächst die Koordinatenachsen und vielleicht auch das Karogitter anzeigen zu lassen.

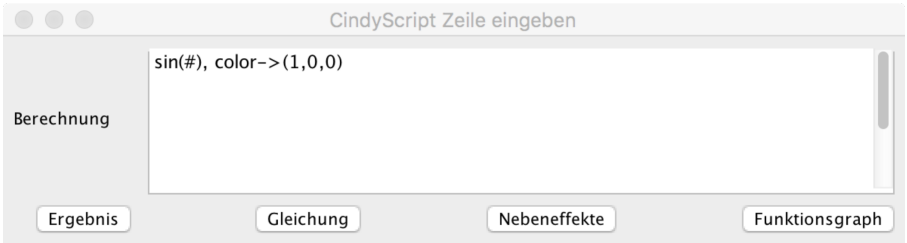
Den Funktionsterm gibt man über die Schaltfläche BERECHNUNG EINES WERTES ein. Es erscheint ein Fenster, in welches man den Funktionsterm eingibt. Eine Besonderheit dabei ist, dass die Funktionsvariable (also typischerweise das *x*) als # geschrieben werden muss. Möchte man eine Sinuskurve erhalten so schreibt man ...



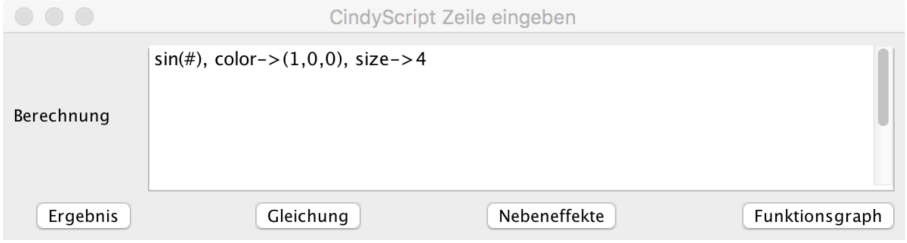
und klickt hernach auf die Schaltfläche FUNKTIONSGRAPH.

Den Cosinus erhält man über `cos(#)`, den Tangens mit `tan(#)` und die Parabel wird nach der Eingabe von `#^2` gezeichnet.

So schnell und einfach damit Funktionsgraphen gezeichnet werden können, so unübersichtlich wird die Sache, wenn man mehrere Graphen plotten lässt. Das Eigenschaftsfenster hilft uns hier nicht weiter, da ein Graph nicht selektierbar ist. Man kann allerdings den jeweiligen Befehl zum Plotten eines Graphen modifizieren: Klickt man mit ausgewählter Schaltfläche BERECHNUNG EINES WERTES den Funktionsterm auf dem Arbeitsblatt erneut an, so öffnet sich das Eingabefenster wieder und man kann den Befehl modifizieren. Abgetrennt durch ein Komma kann das Schlüsselwort *color* gefolgt von einem Pfeil (–>) und dem gewünschten Farbwert angegeben werden:



Die Farbe wird über das RGB-Modell durch ein Zahlentripel angegeben, bei dem die erste Zahl für den Rotanteil, die zweite für den Grünanteil und die dritte für den Blauanteil steht. Die Zahlen selbst können Werte zwischen Null und Eins annehmen. Der obige Befehl erzeugt somit eine rote Sinuslinie. Eine weitere hilfreiche Eigenschaft, welche man dem Graphen anhängen kann, ist dessen Strichstärke:



Alle Eigenschaften, welche bei der Darstellung von Graphen modifiziert werden können, finden Sie unter HILFE / ONLINE-DOKUMENTATION. Dort schauen Sie im REFERENCE GUIDE / CINDYSCRIPT / DRAWING / FUNCTION PLOTTING nach. Damit ist es in Cinderella einfach und schnell möglich, mehrere Graphen plotten zu lassen und damit einen raschen Überblick über Funktionen zu erhalten.

2.3.2 Newton-Verfahren

2.3.2.1 Darstellung des Verfahrens

Sie kennen sicher die Mitternachtsformel zur Lösung quadratischer Gleichungen bzw. zur Nullstellensuche quadratischer Funktionen. Auch für Funktionen dritten und vierten Grades gibt es noch solche geschlossenen Lösungsformeln – wobei diese deutlich komplexer aufgebaut sind als die Mitternachtsformel. Für Polynom-Funktionen fünften und höheren Grades gibt es solche Lösungsformeln jedoch nicht mehr. Sobald in Gleichungen trigonometrische Funktionen enthalten sind, lassen sich diese auch nicht symbolisch lösen, man muss vielmehr auf numerische Näherungsverfahren ausweichen. Ein solches Näherungsverfahren ist das Newton'sche Näherungsverfahren, dessen Funktionsweise wir nachfolgend in Cinderella demonstrieren wollen.

Beim Newton'schen Näherungsverfahren wird die Funktion in einem Ausgangspunkt nahe einer Nullstelle linearisiert, indem man die Tangente des Graphen bestimmt und die Nullstelle der Tangente als Näherung für die Nullstelle der Funktion verwendet. Die erhaltene Näherung dient als Ausgangspunkt für einen weiteren Verbesserungsschritt.

Zur Darstellung des Verfahrens gehen wir aus von der Funktion

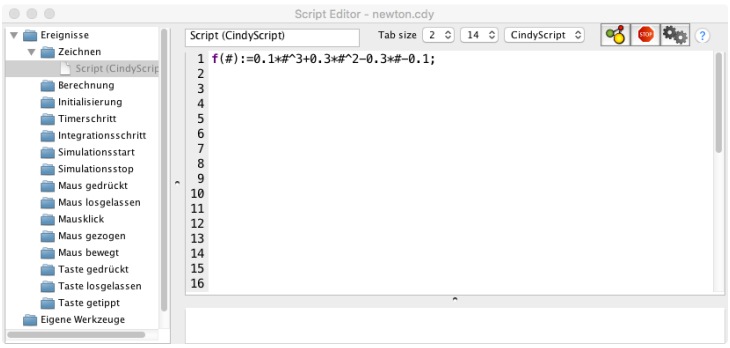
$$f(x) = 0.1x^3 + 0.3x^2 - 0.3x - 0.1$$

und erzeugen zunächst den zugehörigen Funktionsgraphen:

Wer aufmerksam die Eingabefenster für die Funktionsterme betrachtet hat, dem ist in der Überschrift aufgefallen, dass es sich bei den getätigten Eingaben und Befehle der Cinderella-internen Programmiersprache *CindyScript* handelt. In der Tat steht damit ein mächtiges Werkzeug zur Verfügung, mit dem sich in Cinderella auch sehr komplexe Aufgaben erledigen lassen. Für einen orientierenden Überblick lohnt der Blick in die Onlinehilfe (HILFE / ONLINE DOKUMENTATION), dort das Kapitel *CindyScript* anwählen und einfach durch die dortigen Themen querlesen.

2.3.2.2 Arbeiten mit dem Editor

Da wir nun nicht nur einzelne Script-Zeilen erstellen und eingeben wollen, öffnen wir in einer neuen Cinderella-Datei über SCRIPTING / SCRIPTE ERSTELLEN den Script-Editor. Dieser Editor ist dreigeteilt: In der linken Spalte sind die Ereignisse aufgeführt, bei denen ein zugeordnetes Script aufgerufen wird, wir verwenden ausschließlich das Ereignis ZEICHNEN. Der große Bereich rechts oben ist das eigentliche Editorfenster und darunter ist ein Bereich, in welchem Fehlermeldungen angezeigt werden. Wählen Sie im Ereignisfenster das Ereignis ZEICHNEN aus, dann tragen Sie im Editorfenster die obige Funktion ein:



Achtung: Versuchen Sie **nicht**, die angegebenen Befehle hier aus dem Skript zu kopieren und in den Editor einzufügen. Dabei können unsichtbare Steuerzeichen mit übertragen werden, die von Cindy-Script nicht verarbeitet werden können. Sie erhalten dann zwar keine Fehlermeldung – aber es erscheint in Cinderella auch nicht das Gewünschte! Tippen Sie die Befehle aus dem Skript stattdessen von Hand sorgfältig ab!

Ebenso müssen Sie unbedingt darauf achten, als Dezimaltrennzeichen statt des gewohnten Kommas immer einen Punkt einzugeben, also 3 *PUNKT* 1415 statt 3 *KOMMA* 1415!

Bei der Eingabe wird automatisch im Ereignisordner ZEICHNEN ein neues Script angelegt. Bitte achten Sie darauf, die Koeffizienten explizit mit dem Multiplikationszeichen „*“ einzubinden (also `0.3*#` statt nur `0.3#`) und nach dem Potenzierungszeichen „^“ müssen Sie wahrscheinlich zunächst die Leertaste drücken.

Mit der bloßen Eingabe der Funktion passiert noch gar nichts, um den zugehörigen Graphen plotten zu lassen, bedarf es eines *plot*-Befehls. Um in der nächsten Zeile einen weiteren Befehl eingeben zu können, müssen Sie zunächst den vorhergehenden Befehl mit einem Strichpunkt beenden. In die neue Zeile kommt dann der Befehl

```
plot(f(#),size->3, color->(1,0,1));
```

Die eingegebenen Befehle bringt man über die Zahnrad-Schaltfläche des Editorfensters zur Ausführung. Sie sollten hernach den Graphen sehen. Schalten Sie zudem die Anzeige der Koordinatenachsen und des Karogitters ein.

Setzen Sie nun drei Punkte *A*, *B* und *C* auf das Zeichenblatt. Als nächste Befehle im Script folgen:

```
A.y=0;
B.y=0;
C.y=0;
```

Damit setzen wir die *y*-Koordinaten der Punkte auf 0. Sobald Sie die Zahnradschaltfläche drücken, springen die drei Punkte auf die *x*-Achse und lassen sich auch nur entlang dieser verschieben. *B* und *C* schieben sie vorläufig etwas auf die Seite.

Der Punkt *A* soll unsere erste grobe Näherung für den Nulldurchgang unserer Funktion sein. Für die *x*-Koordinate des Punkts *A* soll der zugehörige Funktionswert bestimmt und diese Stelle auf dem Graphen mit einem Punkt markiert werden.

```
draw([A.x,f(A.x)],color->(1,0,0));
```

Wenn Sie nun den Punkt *A* auf der *x*-Achse verschieben, wandert dieser Punkt auf dem Graphen mit und an eben diesem Punkt am Graphen soll nun die Ableitungstangente gezeichnet werden.

Zunächst zeichnen wir jedoch noch die Verbindungslinie von *A* zum zugehörigen Graphenpunkt:

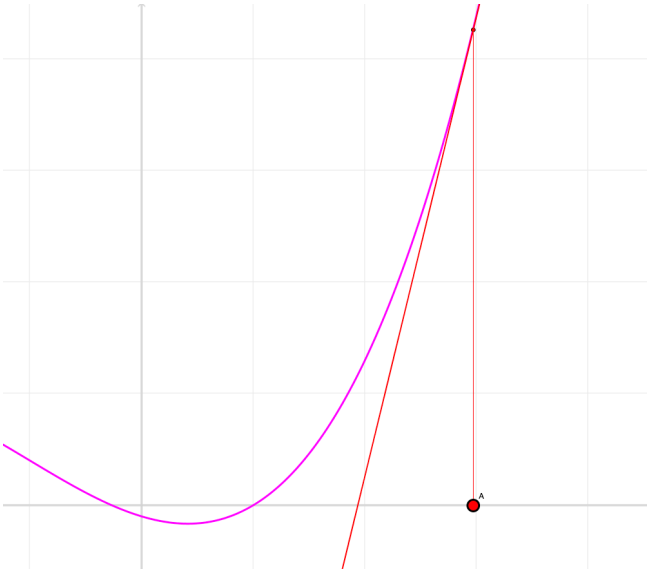
```
draw([[A.x,A.y],[A.x,f(A.x)]], color->(1,0,0));
```

Schließlich erzeugen wir die Tangente:

```
draw(tangent(f(#),A.x), color->(1,0,0), size->2);
```

Wenn Sie nun *A* verschieben, dann läuft nicht nur der Funktionswert auf dem Graphen, sondern auch dessen Tangente mit. Damit können Sie in jedem beliebigen Punkt die Tangente und damit die Ableitung graphisch darstellen.

Für die Durchführung des Newton-Verfahrens setzen Sie *A* etwa auf die *x*-Koordinate 3. Diese Stelle $x = 3$ sei unsere erste Schätzung für die Nullstelle der Funktion.



Die durch den Funktionswert von *f*(3) verlaufende Tangente kann rechnerisch ermittelt werden, indem man die Ableitung an der Stelle $x = 3$ berechnet. Damit und über die Punkt-Steigungs-Form kann die Tangentengleichung bestimmt und deren Nullpunkt berechnet werden.

Wir beschränken uns hier allerdings ausschließlich auf eine graphische Darstellung des Newton-Verfahrens und sehen, dass die Tangente die *x*-Achse bei etwa 2 passiert. Diese Stelle nehmen wir jetzt als Ausgangspunkt für die zweite Näherung: Schieben Sie den Punkt *B* auf diese Stelle, erzeugen Sie den zugehörigen Punkt auf dem Graphen mitsamt der Verbindungslinie und lassen sie die Tangente zeichnen. Verwenden Sie dabei eine andere Farbe!

Sie müssen den interessierenden Ausschnitt nun schon deutlich vergrößern, um das Annähern der Tangentenschnittpunkte an die tatsächliche Nullstelle sichtbar zu machen. Erzeugen Sie schließlich noch die Tangente für den Punkt *C*.

Deutlich wird das rasche Konvergieren des Newton-Verfahrens zur tatsächlichen Nullstelle. Untersuchen Sie auch die anderen Nullstellen der Funktion mit Hilfe dieses Verfahrens. Wo sind die Grenzen des Verfahrens?

2.3.2.3 Automatisieren des Näherungsverfahrens

So wie Sie bisher das Newton-Verfahren realisiert haben, müssen Sie den Punkt *B* manuell in den Schnittpunkt der roten Tangenten mit der *x*-Achse schieben und ebenso den Punkt *C* in den Schnittpunkt der blauen Geraden mit der *x*-Achse. Wir wollen daher noch eine Lösung finden, mit der dies automatisch geschieht:

Nach wie vor soll dabei der Punk *A* (der Startwert unseres Näherungsverfahren) frei auf der *x*-Achse positioniert werden können. Der Punkt *B* – d.h. seine *x*-Koordinate – soll allerdings automatisch auf den Schnittpunkt der von *A* erzeugten roten Tangente mit der *x*-Achse positioniert werden.

Hierfür müssen wir mathematisch die Nullstelle der (roten) Tangente bestimmen: Wir kennen mit `A.x` und `f(A.x)` den Tangentenberührpunkt und wir können uns von Cinderella die Tangentensteigung in eben diesem Punkt berechnen lassen:

```
m1=d(f(#),A.x);
```

Die Funktion `d()` (d : derivate, Ableitung) ermittelt die Ableitung der Funktion `f(#)` an der Stelle `A.x` und weist diesen Wert der Variablen `m1` zu. Damit kennen wir von der Tangente einen Punkt sowie die Steigung in diesem Punkt und können aus der Punkt-Steigungsform die gewohnte Form der Geradengleichung herleiten:

$$m = \frac{y - y_0}{x - x_0}$$

$$\Rightarrow y = mx - mx_0 + y_0$$

Von dieser Geraden suchen wir die Nullstelle:

$$0 = mx - mx_0 + y_0$$

$$\Rightarrow x = x_0 - \frac{y_0}{m}$$

Auf genau diese Nullstelle muss der Punkt *B* gesetzt werden:

```
B.x=A.x-f(A.x)/m1;
```

Fügen Sie beide Anweisungen direkt hinter dem Zeichenbefehl für die erste Tangente in Ihr Skript ein und klicken Sie auf die Zahnradschaltfläche: Der Punkt *B* muss dann auf den Schnittpunkt der roten Tangente mit der *x*-Achse springen und dort verbleiben.

Kopieren Sie beide Anweisungen und fügen Sie diese nach dem Zeichenbefehl für die zweite Tangente ebenfalls ein. Die Punktbezeichnungen müssen Sie anpassen, danach sollte der Punkt C auf den Schnittpunkt der zweiten, blauen Tangente mit der *x*-Achse springen.

Wenn Sie schließlich noch den Schnittpunkt der grünen Tangente ermitteln und einzeichnen lassen, dann wird damit die durch unser dreimaliges Newtonverfahren ermittelte Nullstelle angezeigt.

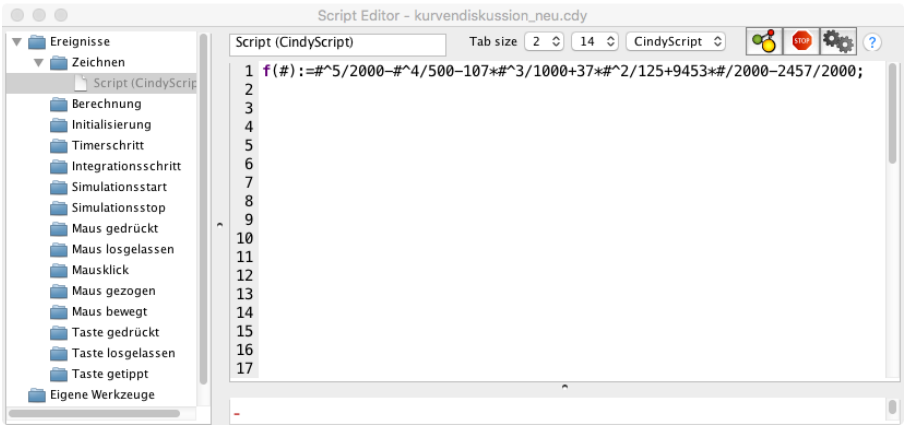
2.3.3 Kurvendiskussion mit Cinderella

Einige weitere hilfreiche Konzepte von *CindyScript* sollen im Folgenden anhand der Thematik einer Kurvendiskussion dargestellt werden.

Nachfolgend wollen wir die Funktion

$$f(x)=\frac{x^5}{2000}-\frac{x^4}{500}-\frac{107x^3}{1000}+\frac{37x^2}{125}+\frac{9453x}{2000}-\frac{2457}{500}$$

darstellen und untersuchen. Öffnen Sie hierfür ein neues, leeres Zeichenblatt. Da wir nicht nur einzelne Script-Zeilen erstellen und eingeben wollen, öffnen wir wieder über SCRIPTING / SCRIPTE ERSTELLEN den Script-Editor. Wählen Sie im Ereignisfenster das Ereignis ZEICHNEN aus, dann tragen Sie im Editorfenster die obige Funktion ein:



Mit der bloßen Eingabe der Funktion passiert nichts, um den zugehörigen Graphen plotten zu lassen, bedarf es eines plot-Befehls. In die neue Zeile kommt der Befehl

```
plot(f(#),size->6);
```

Beachten Sie, dass hierbei die Funktionsvariable *x* durch die Laufvariable # ersetzt werden muss! Damit auf dem Zeichenblatt tatsächlich die Funktion erscheint, müssen Sie dafür sorgen, dass ihr Script ausgeführt wird. Dies macht man durch Klicken auf die Zahnradschaltfläche rechts oben. Ggf. müssen Sie Ihr Zeichenblatt neu skalieren, um den interessanten Teil der Funktion komplett zur Anzeige zu bringen.

2.3.3.1 Funktionaler Zusammenhang

Als Erstes wollen wir die Funktion punktweise untersuchen und eine zahlenmäßige Zuordnung der *x*- und *y*-Werte schaffen. Setzen Sie dazu auf dem Zeichenblatt zwei Punkte *A* und *B* irgendwo ab.

In den Scripteditor geben Sie als nächste Zeile ein:

```
A.y=0;
```

Dies bedeutet, dass die y-Koordinate des Punkts *A* auf 0 gesetzt wird. Sobald Sie die Zahnradschaltfläche drücken, springt der Punkt *A* auf die *x*-Achse und kann auch nur noch entlang dieser bewegt werden. Damit wurde dieser Punkt *A* zum Selektor, um einen ganz bestimmten *x*-Wert anzuwählen, für den wir unsere Funktion näher untersuchen wollen. Der Punkt *B* soll genau an dem mit *A* gewählten *x*-Wert auf dem Graphen positioniert werden. Der nächste Befehl

```
B.x=A.x;
```

sorgt dafür, dass sich *B* parallel mit *A* in *x*-Richtung mitbewegt. Um *B* nun noch genau auf den Graphen zu setzen, müssen wir den Funktionswert an der Stelle *A.x* (oder *B.x*) errechnen und *B.y* auf eben diesen Funktionswert setzen:

```
B.y=f(A.x);
```

Jetzt läuft *B* parallel entlang zu *A* immer auf dem Funktionsgraphen mit.

Wir besorgen uns einen dritten Punkt *C* und sorgen dafür, dass dieser Punkt *C* immer den zugehörigen Funktionswert auf der *y*-Achse anzeigt:

```
C.x=0;
C.y=B.y
```

Stellen Sie *B* und *C* kleiner dar, *A* und *C* färben Sie schwarz und *B* blau.

Wenn wir jetzt noch die Verbindungslinien von *A* nach B und von dort nach *C* zeichnen lassen, dann haben wir den funktionalen Zusammenhang optisch dargestellt:

```
draw(A,B);
draw(B,C);
```

Den zahlenmäßigen Zusammenhang erhalten wir, indem wir den jeweiligen *x*- und *y*-Wert angeben lassen. Wir machen dies, indem wir zwei kleine Tabellen erstellen, eine mit den Zeilenüberschriften *x* und *y* und eine Zweite mit den zugehörigen Werten. Durch eine entsprechende Positionierung erscheinen dann beide Tabellen wie eine Einzige:

```
head=("x:","y:");
table=(B.x,B.y);
drawtable((15,0),head, width->30, align->"mid");
drawtable((17,0),table, width->100, align->"right");
```

Das erste Zahlenpaar im Befehl `drawtable` gibt die Position der zu erzeugenden Tabelle an.

2.3.3.2 Die Ableitung

Bei Funktionen interessiert nicht nur der Zusammenhang zwischen *x*- und *y*-Werten sondern auch deren Änderungsverhalten. Dieses geht der Frage nach, ob die Funktionswerte rechts und links von einem bestimmten x_0-Wert steigen oder fallen. Eine Antwort darauf liefert der Differenzialquotient

$$\frac{f(x+\Delta x)-f(x)}{\Delta x} \text{ für } \Delta x \text{ gegen } 0$$

und damit gleichzeitig die Steigung der Tangente an den Graphen an der Stelle x_0. Dieser grundsätzliche Zusammenhang ist häufig durch das bloße und meist sinnleere Ableiten in der Oberstufe verloren gegangen und wir tun gut daran, uns diese grundlegende Problemstellung der Analysis wieder ins Gedächtnis zu rufen und uns vor Augen zu stellen. Dies geht in Cinderella ungemein einfach, da *CindyScript* einen Befehl zum Erzeugen einer Tangente an einen Funktionsgraphen enthält:

```
draw(tangent(f(#),A.x),color->(0,1,0), size->3);
```

Gezeichnet wird die Tangente an die Funktion *f* an der Stelle `A.x`. Damit können wir die Tangente durch Verschieben des Punktes *A* über den Graphen wandern lassen.

Wenn man nun jedem Punkt `A.x` die Steigung der Tangente an diesem Ort zuordnet, dann erhält man die (erste) Ableitungsfunktion. Diese Ableitung kann man in Cinderella einfach erzeugen und anzeigen lassen:

```
g(x):=d(f(#),x);
plot(g(#),color->(0,1,0),size->3);
```

Die erste Zeile erzeugt die Ableitungsfunktion $g(x)$ und die zweite Zeile plottet den zugehörigen Graphen.

2.3.3.3 Extrempunkte

Wandert man mit dem Punkt *A* durch den angezeigten Bereich der Funktion, so wird der Zusammenhang zwischen der Tangentensteigung und der ersten Ableitung augenfällig: An den Hoch- und Tiefpunkten verläuft die Tangente waagerecht mit der Steigung Null, dementsprechend hat der Graph der ersten Ableitung an eben diesen Stellen jeweils einen Nulldurchgang. Genau diesen Zusammenhang macht man sich bei der Suche nach Extrempunkten zu Nutze wenn man die erste Ableitung gleich Null setzt.

Damit ist zwar klar, dass an den jeweiligen Stellen Extrempunkte vorliegen, man weiß aber noch nicht, ob es sich dabei um einen Hoch- oder Tiefpunkt handelt. Dies kann man rechnerisch herausfinden, indem man Funktionswerte knapp rechts und links von festgestellten Extrempunkten berechnet und diese mit dem Funktionswert am Extrempunkt vergleicht. Alternativ kann man auch genau beobachten, was unsere Tangente beim Durchfahren von solchen Extrempunkten macht: Lassen wir die Tangente von links nach rechts wandern, so wird deutlich, dass sie vor der Annäherung an Hochpunkte von einer positiven auf eine negative Steigung wechselt. Die Ableitungsfunktion wechselt somit vom Positiven ins Negative und hat damit eine negative Steigung! Bei einer Annäherung an einen Tiefpunkt ist es gerade umgekehrt: Die Tangentensteigung wechselt von einem negativen zu einem positiven Wert, die Ableitungsfunktion wechselt vom Negativen ins Positive, hat dort also eine positive Steigung.

Für die Ableitung der Ableitung lassen wir uns zunächst die Tangente an den Graphen der Ableitungsfunktion zeichnen. Dafür benötigen wir einen weiteren Punkt *D*, welcher auf dem Ableitungsgraphen mitlaufen soll und durch den die Tangente gezeichnet wird.

```
D.x=A.x;
D.y=g(A.x);
draw(tangent(g(#),A.x),color->(1,0,0), size->1);
```

Die Steigung der Ableitungsfunktion erhalten wir, indem wir diese ebenfalls ableiten. In Cinderella geht das ganz einfach:

```
h(x):=d(g(#),x);
plot(h(#),color->(1,0,0));
```

Die rote Tangente gibt nun die Steigung der ersten Ableitung an und es sofort ersichtlich, dass diese an den Hochpunkten der Funktion eine negative und an den Tiefpunkten eine positive Steigung hat. Entsprechend verläuft der rote Graph der zweiten Ableitung an Hochpunkten im negativen und an Tiefpunkten im positiven Bereich. Genau dies ist der Grund, warum Sie zur Bestimmung von Hoch- und Tiefpunkten auch noch die zweite Ableitung bestimmen und deren Wert an den gefundenen Extrempunkten berechnen mussten!

2.3.3.4 Wendepunkte

Verfolgen wir nun das Verhalten der grünen Tangente an den Funktionsgraphen von dessen links gelegenen ersten Hochpunkt bis zum nächsten Tiefpunkt: Im Hochpunkt hat die Tangente die Steigung 0, der Graph der Ableitung folglich eine Nullstelle. Wandert man weiter nach rechts, so dreht die Tangente nach rechts und zeigt immer steiler nach unten; konsequenter Weise läuft die Ableitungsfunktion nach ihrer Nullstelle ins Negative. Bald kommt jedoch der Punkt, wo die Rechtsdrehung der Tangente aufhört und sich diese wieder nach links dreht. Diesen Punkt nennt man bekanntlich Wendepunkt. Je näher man dem Tiefpunkt kommt, um so mehr muss die seitherige negative Steigung sich wieder der Waagerechten und damit der Steigung Null annähern. Irgendwo zwischen Hoch- und Tiefpunkt muss daher die Tangente die betragsmäßig größte Steigung gehabt haben. Die Ableitungsfunktion hat an eben dieser Stelle

einen Tiefpunkt. Im Umkehrschluss markiert dieser Tiefpunkt der ersten Ableitung den Wendepunkt des Funktionsgraphen.

An der betrachteten Stelle hat der Funktionsgraph von einer „Rechtskurve“ in eine „Linkskurve“ gewechselt. Untersucht man einen Bereich zwischen Tief- und Hochpunkt, so wechselt der Funktionsgraph von einer „Linkskurve“ zu einer „Rechtskurve“, am zugehörigen Wendepunkt hat die erste Ableitung ein Maximum.

Um Wendepunkte feststellen zu können, muss man die Extrempunkte der Ableitungsfunktion suchen. An den Extrempunkten der ersten Ableitung muss logischer Weise die zweite Ableitung gleich Null sein. An unseren Funktionen in Cinderella kann dieser Zusammenhang problemlos nachvollzogen werden.

Schließlich möchte man noch herausfinden, ob es sich um einen „rechts-links-Wendepunkt“ oder einen „links-rechts-Wendepunkt“ handelt.

Hierzu muss man den Verlauf der zweiten Ableitung betrachten. Wechselt dieser – wie in unserem Beispiel – vom negativen in den positiven Bereich, dann liegt ein rechts-links-Wechsel vor, sonst ein links-rechts-Wechsel.

Ob nun die zweite Ableitung an ihren Nulldurchgängen vom Positiven ins Negative wechselt oder umgekehrt, kann man schließlich über deren Ableitung – die nun dritte Ableitung – feststellen. Dabei gilt der folgende Zusammenhang:

Art des Wendepunkts	zweite Ableitung	dritte Ableitung
rechts–links	negativ–positiv	positiv
links–rechts	positiv–negativ	negativ

Versucht man allerdings nach dem oben gezeigten Muster $k(x)$ als dritte Ableitung aus $h(x)$ zu erzeugen und plotten zu lassen, dann kommt Cinderella schwer ins Straucheln: Die Antwortzeiten werden ungemein lang und die dritte Ableitung wird – wenn überhaupt – nur sehr unscharf dargestellt. Dies rührt daher, dass Cinderella nicht symbolisch differenziert (wie Sie das in der Schule gelernt haben), sondern numerisch vorgeht, indem für jede Stelle der Differenzenquotient berechnet wird. Dies scheint zweimal hintereinander zu funktionieren – aber eben kein drittes Mal. Man kann sich behelfen, indem man die dritte Ableitung von Hand bildet und den Funktionsterm direkt eingibt.

Bei umfangreichen Skripten können Kommentare hilfreich sein: Alles was nach zwei Schrägstrichen (//) steht, gilt als Kommentar und wird hellgrau dargestellt.

2.3.4 Affine Transformationen

Wahrscheinlich kennen Sie die Scheitelpunktsform einer Parabel

$$y = a \cdot (x - c)^2 + d$$

und wissen, dass der Scheitelpunkt an der Koordinate $[c,d]$ zu finden ist. Und sicherlich haben Sie auch noch im Hinterkopf, dass a der Formfaktor der Parabel ist. Was im Unterricht der Sekundarstufe für die Normalparabel – den Graphen der Funktion $f(x) = x^2$ – beispielhaft und ausschnittweise gezeigt wird, nennt man in der Mathematik eine *affine Transformation*. Diese lässt sich recht anschaulich durch das Maschinenmodell erläutern. Danach ist eine Funktion eine Maschine, welche eine Eingabezahl entgegennimmt und daraus nach festgelegten Regeln eine Ausgabezahl produziert. Die Quadratfunktion ist nach diesem Bild eine Maschine, die für jede Eingabezahl deren Quadrat ausgibt.

Nun kann man auf eine solche Maschine auf mehreren Weisen einwirken: Man kann eine Eingabezahl noch vor der Eingabe mit einer bestimmten Zahl multiplizieren oder addieren und man kann die Ausgabezahl ebenfalls mit einer Zahl multiplizieren oder addieren. Es gibt somit insgesamt vier Möglichkeiten des Einwirkens auf eine beliebige Funktion $f(x)$ und man erhält damit jeweils eine neue Funktion:

Eingabezahl verändern:

$f(b \cdot x)$	multiplikativ
$f(x+c)$	additiv

Ausgabezahl verändern:

$a \cdot f(x)$	multiplikativ
$f(x)+d$	additiv

Diese vier Möglichkeiten kann man beliebig kombinieren und erhält damit eine affin transformierte Funktion

$$g(x) = a \cdot f(b \cdot x+c)+d$$

Aus dem oben über die Scheitelpunktsform Gesagten kann man unschwer ableiten, dass die Parameter a, b, c und d einer solchen Transformation den Funktionsgraphen in Lage und Form beeinflussen. Welche Auswirkungen diese 4 Parameter im Einzelnen haben, werden wir uns dynamisch darstellen lassen.

Die Darstellung der Ausgangsfunktion erzeugen wir über ein Script in Cinderella:

```
f(#):=1/2*#^3-1/2*#^2-4*#+6;
plot(f(#),color->(1,0,0));
```

Dann benötigen wir 4 Punkte A, B, C und D, die wir als Schieberegler für die 4 Parameter a, b, c und d einer affinen Transformation verwenden wollen. Wohin wir die vier Punkte auf dem Zeichenblatt absetzen ist völlig egal, weil wir deren Lage ohnehin mit CindyScript beeinflussen werden:

```
A.y=-1;
B.y=-2;
C.y=-3;
D.y=-4;
```

Damit haben wir vier Schieberegler, mit denen wir nun die affine Transformation beeinflussen können. Dies geschieht, indem wir eine Funktion $g(x)$ definieren, welche die affin transformierte Funktion $f(x)$ beinhaltet:

```
g(#):=A.x*f(B.x*#+C.x)+D.x;
plot(g(#),color->(0,0,1));
```

Mit der Funktion `drawtext()` kann schließlich noch der mit den Reglern eingestellte Wert sichtbar gemacht werden:

```
drawtext((5,-1),A.x,size->18);
drawtext((5,-2),B.x,size->18);
drawtext((5,-3),C.x,size->18);
drawtext((5,-4),D.x,size->18);
```

2.4 Abbildungen

Abbildungen und *Funktionen* sind in der Mathematik synonyme Begriffe. Während man den Terminus *Funktion* in aller Regel für Zuordnungen zwischen Zahlen(mengen) verwendet, spricht man bei Zuordnungen zwischen Mengen von Punkten in der Regel von *Abbildungen*. So werden bei jeder Fotografie die Punkte des uns umgebenden dreidimensionalen Raums auf die Punkte der Bildebene abgebildet. Während Ihrer Schulzeit haben Sie sich – zumindest rudimentär – mit Kongruenzabbildungen beschäftigt.

2.4.1 Kongruenzabbildungen

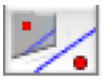
Abbildungen sind Zuordnungen der Menge der Punkte der (Zeichen-)Ebene auf sich selbst. Bei solchen Abbildungen werden grundsätzlich immer alle Punkte der Ebene abgebildet – dies ist jedoch nur in der Theorie möglich. In der Realität behilft man sich, nur endlich viele Punkte abzubilden. Beschränkt man sich auf die Darstellung der Abbildung von drei Punkten, dann hat man ein Dreieck abgebildet, zeigt man die Abbildung anhand von vier Punkten, so wurde dementsprechend ein Viereck abgebildet. Man sollte aber im Hinterkopf behalten, dass immer alle Punkte abgebildet werden, die ausschließliche Betrachtung von nur drei oder vier Punkten kann zu Problemen und Fehlern führen!

Ihren Namen haben Kongruenzabbildungen erhalten, weil sie sehr viele Invarianten haben. Bildet man zwei Punkte mit Hilfe einer Kongruenzabbildung ab, so ist der Abstand zwischen den zwei Urbildpunkten derselbe wie derjenige zwischen den beiden Bildpunkten. Man sagt, die Abbildung ist streckentreu. Daraus folgt dann automatisch, dass Kongruenzabbildungen auch flächeninhalts- und winkelmaßtreu sind. Bildet man also ein Dreieck oder Viereck ab, so wird das Bilddreieck dieselben Seitenlängen, Winken und denselben Flächeninhalt haben, beide Drei- bzw. Vierecke sind somit kongruent.

Als Kongruenzabbildungen haben Sie wahrscheinlich die Achsenspiegelung, die Verschiebung, die Drehung und – als Sonderfall einer Drehung um 180° – die Punktspiegelung kennen gelernt.

2.4.1.1 Achsenspiegelung

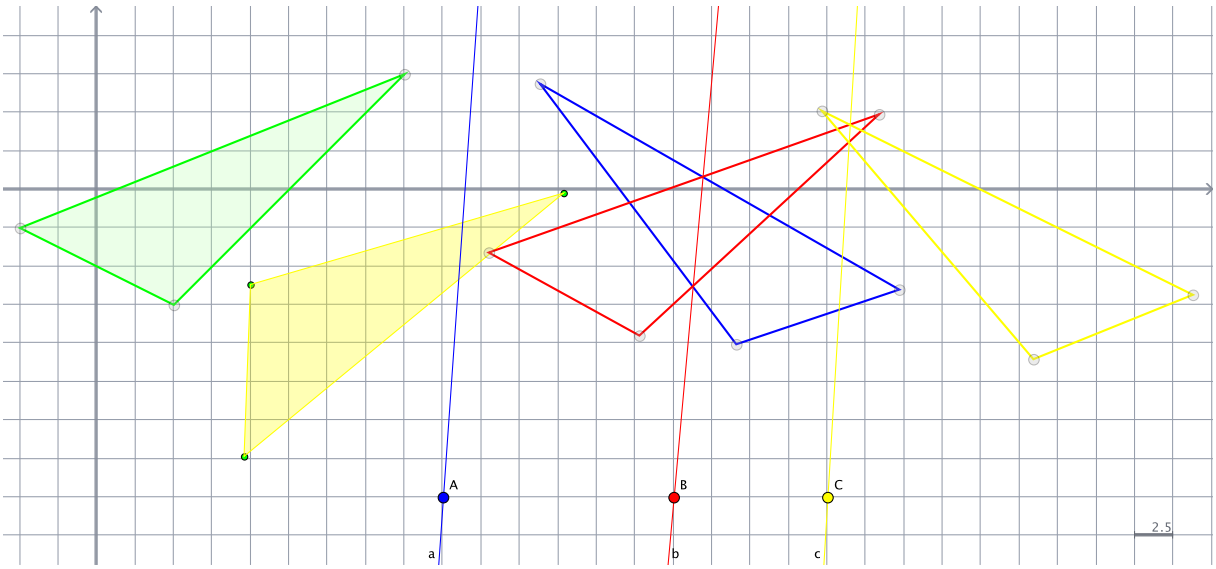
Wir experimentieren zu Beginn mit der Achsenspiegelung. Erzeugen Sie zwei Geraden, die sich in einem gemeinsamen Punkt *A* schneiden und die beide um diesen Punkt gedreht werden können. Stellen Sie beide Geraden anfangs nach Augenmaß in etwa senkrecht aufeinander.



Setzen Sie in einen der entstehenden Quadranten beliebig einen Punkt *B*. Dieser soll an einer der Achsen gespiegelt werden. Dies geschieht entweder über das Menü MODI / TRANSFORMATIONEN / SPIEGELUNGEN (ALT) oder über die zugehörige Schaltfläche. Spiegeln Sie ebenso den erhaltenen Bildpunkt an der zweiten Achse und dessen Bildpunkt wiederum an der ersten Achse. Bilden Sie aus den so erzeugten vier Punkten ein Polygon und gehen Sie der Frage nach, um was für ein Viereck es sich grundsätzlich handelt. Verändern Sie dabei sowohl den Winkel zwischen den beiden Spiegelachsen als auch die Lage des Urbildpunktes *B*.

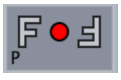
„Zwei Dreiecke sind kongruent, wenn Sie durch eine endliche Anzahl an Kongruenzabbildungen aufeinander abgebbildet werden können."

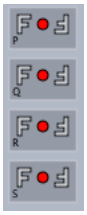
Laden Sie die Datei *kongruenzabbildung.cdy*. Sie enthält ein grün und ein gelb ausgefülltes Dreieck. Diese sind zueinander kongruent. Weisen Sie diese Kongruenz nach, indem Sie die drei Achsen *a*, *b* und *c* so positionieren, dass das letztresultierende Bild (gelb umrahmtes Dreieck) genau auf dem gelb ausgefüllten Dreieck zu liegen kommt. Dabei ist das blau umrandete Dreieck das Bild des grünen Dreiecks nach Spiegelung an der blauen Achse. Das rot umrandete Dreieck ist das Bild des blauen Dreiecks nach Spiegelung an der roten Achse und das gelb umrandete Dreieck ist das Bild des roten nach Spiegelung an der gelben Achse.



2.4.1.2 Punktspiegelung

Setzen Sie die vier Punkte *P*, *Q*, *R* und *S* links unten beginnend grob im Gegenuhrzeigersinn auf einem neuen Zeichenblatt ab und stellen Sie diese vier Punkte als Polygon dar. Jeder dieser Eckpunkte sei das Zentrum einer Punktspiegelung. Diese vier Punktspiegelungen definieren Sie über die Menüoption MODI / TRANSFORMATION / SPIEGELUNG und klicken dann sukzessive auf die Punkte *P*, *Q*, *R* und *S*. Nach jedem Klicken entsteht am rechten Rand des Zeichenfensters das jeweilige Abbildungsobjekt. Insgesamt müssten Sie danach insgesamt 4 Abbildungsobjekte erhalten haben, welche die vier Punktspiegelungen an *P*, *Q*, *R* und *S* symbolisieren.





Um Objekte (Punkte, Strecken, Geraden, Kreise, Polygone, ...) über solcherart definierte Abbildungen abzubilden, gehen Sie grundsätzlich immer gleich vor: Klicken Sie zuerst auf das Auswahlsymbol . Danach selektieren Sie das abzubildende Objekt ebenfalls durch Anklicken. Schließlich klicken Sie auf das Abbildungsobjekt, welches die Abbildung durchführen soll.

Erzeugen Sie etwas links vom Punkt *P* den Punkt *A*. Spiegeln Sie *A* an *P*, den Bildpunkt *B* an *Q*, den Bildpunkt *C* an *R* und schließlich den Bildpunkt *D* an *S*. Letzter Bildpunkt in dieser Kette ist *E*. Verbinden Sie *A* mit *B*, *B* mit *C*, *C* mit *D* und *D* mit *E* durch Strecken.

Wenn Sie *A* dynamisch bewegen, dann bewegen sich *B*, *C*, *D* und *E* ebenfalls. Schaffen Sie es, *A* auf eine Position zu bringen, wo *E* und *A* zusammenfallen? Wenn Sie *A* bewegen, welchen Eindruck haben Sie von der Bewegung von *E*?

Können Sie die Punkte *P*, *Q*, *R* und *S* so arrangieren, dass *A* und *E* aufeinander fallen? Beantworten Sie dazu folgende Fragen: Wie bewegt sich *C* in Abhängigkeit von *A*? Wie hängt diese Bewegung von den Spiegelpunkten *P* und *Q* ab? Wie bewegt sich *E* in Abhängigkeit von *C*? Wie hängt diese Bewegung von den Spiegelpunkten *R* und *S* ab? Welche Lage müssen daher *P*, *Q*, *R* und *S* einnehmen, damit *A* und *E* zusammenfallen? Welchen Satz haben Sie damit bewiesen?

2.4.1.3 Verschiebung und Drehung

Um eine Verschiebung oder Drehung zu definieren braucht es etwas Vorarbeit: Eine Verschiebung ist durch einen Verschiebungsvektor gegeben, folglich müssen wir zunächst einen solchen Vektor erstellen. Dabei muss es sich nicht um einen Vektor im mathematischen Sinne handeln, eine Strecke zwischen zwei Punkten *A* und *B* reicht aus. Ruft man über MODI / TRANSFORMATION / VERSCHIEBUNG die Definition einer Verschiebung auf, so muss man einen Urbild- (preimage-) und einen Bildpunkt (image-point) angeben. Wenn man dabei zunächst auf *A* und

dann auf *B* klickt, stellt *AB* den Verschiebungsvektor dar und die erzeugte Verschiebung wird als Abbildungsobjekt am rechten Rand des Zeichenblattes erstellt. Möchte man ein Objekt verschieben, so muss man es selektieren und hernach auf das Verschiebungsobjekt klicken: Erzeugen Sie zuerst ein Dreieck als Polygon, wählen es aus und klicken dann auf das Abbildungsobjekt. Verändert man *A* oder *B*, so verändert man damit den der Verschiebung zugrundeliegenden Vektor und damit die Verschiebung insgesamt.

Eine Drehung ist durch einen Drehpunkt und einen Drehwinkel festgelegt. Wir setzen deshalb einen Punkt *P* auf die Zeichenfläche. Den Drehwinkel konstruieren wir dynamisch, indem wir eine Strecke *QR* und eine weitere (vom selben Punkt *Q* ausgehende) Strecke *QS* zeichnen und den von beiden Strecken eingeschlossenen Winkel messen lassen. Über MODI / TRANSFORMATION / DREHUNG PKT|WINKEL wird dann die Drehung definiert, indem man zunächst auf den Drehpunkt *P* und dann auf das angezeigte Winkelmaß klickt. Als Ergebnis wird die Drehung als Abbildungsobjekt erzeugt. Objekte werden gedreht, indem man sie selektiert und hernach auf das Abbildungsobjekt klickt. Wieder ist die Drehung dynamisch veränderbar, man braucht nur den Drehpunkt verlagern oder den Drehwinkel anhand des Streckenzugs *RQS* verändert. Bilden Sie Ihr oben erzeugtes Dreieck über diese Drehung ab und variieren Sie den Drehpunkt und den Drehwinkel.

2.4.2 Ähnlichkeitsabbildung

Ähnlichkeitsabbildungen lassen sich in Cinderella einfach definieren. Konstruieren Sie zunächst ein beliebiges Dreieck aus drei Punkten *A*, *B* und *C* und verbinden Sie diese Punkte zu einem Polygon. Dann setzen Sie vier weitere Punkte *D*, *F*, *G*, *H* auf der Zeichenebene ab. Über MODI / TRANSFORMATION / ÄHNLICHKEIT teilen Sie Cinderella mit, dass Sie eine Ähnlichkeitsabbildung erzeugen wollen. Cinderella fordert Sie dann auf, auf zwei zusammengehörige Paare von Urbild- und Bildpunkten zu klicken. Klicken Sie nacheinander auf *D*, *F*, *G* sowie *H* und Cinderella wird ein zugehöriges Abbildungsobjekt in der rechten oberen Bildschirmecke erzeugen:

Klicken Sie auf ihr Dreieck *ABC* und hernach auf das erzeugte Abbildungssymbol. Cinderella wird das entsprechende Bilddreieck erzeugen. Verändern Sie die Form Ihres Dreiecks und die Lage der abbildungsbestimmenden Punkte und versuchen Sie, Zusammenhänge und Abhängigkeiten zu erkennen.

In der Schule haben Sie die zentrische Streckung als Ähnlichkeitsabbildung kennengelernt. Ordnen Sie die vier Punkte *D*, *F*, *G*, *H* so an, dass die in Cinderella erzeugte Ähnlichkeitsabbildung eine zentrische Streckung ergibt. Welche Lage müssen dazu die vier Punkte einnehmen?

2.4.3 Konforme Abbildungen

Bei allen Abbildungen, die Sie seither kennenlernten, dürfte es sich ausschließlich um Kollineationen gehandelt haben, also um Abbildungen, bei denen eine Gerade auf eine Gerade abgebildet wird. Bei konformen Abbildungen ist dies nicht mehr der Fall, solche Abbildungen treten insbesondere im Bereich der komplexen Zahlen auf[4]. Eine konforme Abbildung ist die *Möbius-Transformation* genannte Abbildung

$$z \to \frac{a \cdot z + b}{c \cdot z + d} \text{ (mit } z, a, b, c, d \in \mathrm{C})$$

Eine Möbius-Transformation bildet Kreise auf Kreise ab, um eine solche Transformation eindeutig festzulegen, muss daher ein Urbild- und ein Bildkreis angegeben werden. Da anderer-

[4] Wiewohl man einfache konforme Abbildungen – z.B. die Kreisinversion – auch völlig ohne komplexe Zahlen erzeugen kann.

seits ein Kreis durch drei Punkte festgelegt ist, wird eine Möbius-Transformation durch zwei Punkte-Tripletts festgelegt.

Wir benötigen daher sechs Punkte *A* bis *F*, die Sie auf dem Zeichenblatt in der Nähe des Koordinatenursprungs absetzen. Färben Sie alle sechs Punkte zur besseren Wiedererkennung blau.

Das Abbildungsobjekt erzeugen Sie über das Menü MODI / TRANSFORMATION / MOEBIUS TRANSFORMATION. Sie werden aufgefordert drei zugeordnete Punktepaare einzugeben, klicken Sie einfach nacheinander auf die Punkte *A* bis *F*. Sie erhalten dann das Abbildungsobjekt

.

Erzeugen Sie einen Punkt *Z* und lassen Sie diesen Punkt *Z* durch das Abbildungsobjekt abbilden: *Z* markieren und auf das Abbildungsobjekt klicken. Ggf. müssen Sie die Lage der sechs Punkte *A* bis *F* variieren, um den Bildpunkt auf dem Bildschirm zu sehen.

Zeichnen Sie eine Gerade und befestigen Sie den Punkt *Z* auf dieser Geraden. Sobald Sie den Urbildpunkt *Z* auf der Geraden bewegen werden Sie feststellen, dass der Bildpunkt keine Gerade mehr beschreibt. Um festzustellen, auf welcher Bahn sich der Bildpunkt bewegt, lassen Sie dessen Ortslinie zeichnen.

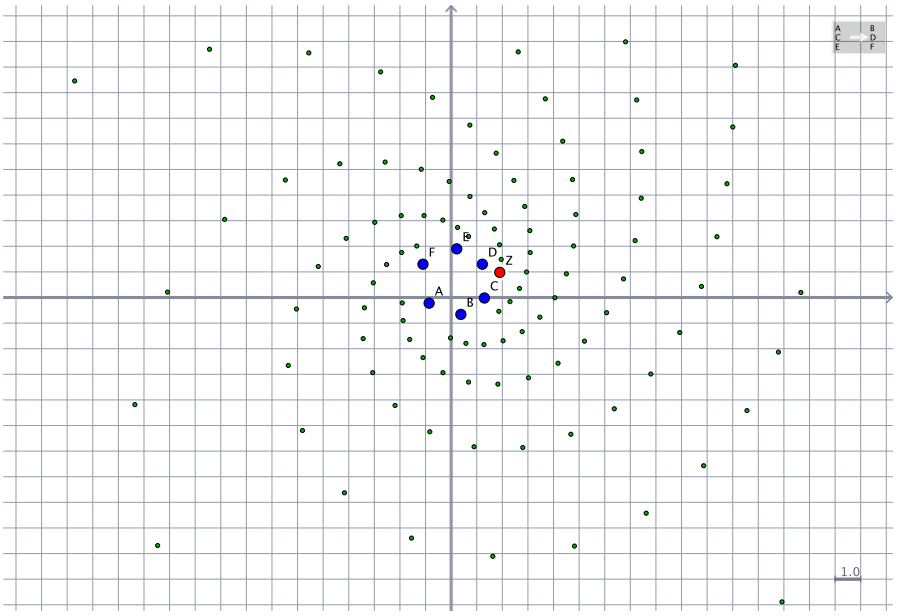
Dies geschieht durch Klicken auf die zugehörige Schaltfläche . Zuerst müssen der Urbildpunkt Z und dann der Bildpunkt angeklickt werden. Sofort wird die Ortslinie als Kreis sichtbar. Verändert man die Lage der Geraden, so ändert sich damit die Größe des Bildkreises. Versuchen Sie die Gerade so zu legen, dass der Bildkreis zu einer Geraden wird!

Konforme Abbildungen sind somit Abbildungen, welche eine Gerade auf einen Kreis (oder eine Gerade) abbilden. Da man sich eine Gerade als Kreis mit unendlich großem Radius vorstellen kann, sagt man auch, dass Kreis und Gerade *verwandt* sind und nennt konforme Abbildungen *kreisverwandt*. Akzeptiert man eine Gerade als „Grenzfall“ eines *Kreises mit unendlichem Radius*, so werden immer Kreise auf Kreise abgebildet.

Die allgemeine Möbius-Transformation lässt nun kaum mehr exakte Voraussagen über die entstehende Abbildung zu, als eben die Tatsache, dass Kreise auf Kreise abgebildet werden und dass Winkel erhalten bleiben, kurz: dass es sich um eine konforme Abbildung handelt. Der „Rest“ bleibt – ohne größeren Rechenaufwand betreiben zu wollen – verborgen und mutet damit etwas chaotisch an. Es ist aber verblüffend, dass trotz dieses „nicht vorhersagbaren“ Verhaltens regelmäßige Muster entstehen, wenn man die jeweilige Transformation iteriert, also wiederholt auf den jeweils zuvor erzeugten Bildpunkt anwendet. Gehen Sie bei den folgenden Schritten exakt und konzentriert vor:

1. Lösen Sie den Urbildpunkt *Z* wieder von der Geraden und löschen Sie die Gerade samt Stützpunkt.
2. Den Bildpunkt von *Z* machen Sie über die Eigenschaftseinstellungen etwas kleiner, färben ihn grün und verbergen seine Bezeichnung.
3. Dann selektieren Sie eben diesen Bildpunkt und klicken hernach fortlaufend gut 100 Mal hintereinander auf das Abbildungsobjekt. Klicken Sie einfach drauflos – völlig unabhängig davon, was auf dem Bildschirm passiert! Sie erzeugen damit eine Verkettung der definierten Möbius-Abbildung, d.h. der Bildpunkt wird seinerseits abgebildet und dessen Bildpunkt wieder und wieder.

Experimentieren Sie hernach mit der Lage der die Abbildung definierenden sechs Punkte *A* bis *F*. Jede Veränderung der Funktionsparameter führt zu einer Änderung des erzeugten Musters und trotz der chaotischen Änderungen bleibt immer eine gewisse Regelmäßigkeit und Symmetrie erkennbar.



2.5 Weitere Anwendungen in Cinderella

2.5.1 Makros in Cinderella

Cinderella enthält alle grundlegenden Konstruktionswerkzeuge der klassischen euklidischen Geometrie. Trotzdem mag es Aufgaben geben, für deren Bewältigung man sich komplexere Konstruktionsmöglichkeiten wünscht. So kann man in Cinderella den Mittelpunkt einer Strecke festlegen und man kann Senkrechte auf Strecken fällen – es gibt jedoch keine eingebaute Konstruktion, welche die Mittelsenkrechte zwischen zwei gegebenen Punkten erstellt. Solche Lücken kann man jedoch problemlos selbst füllen, indem man Makros erstellt. Wie dies geht, sei an dem eben genannten Beispiel der Mittelsenkrechten zu zwei Punkten demonstriert.

2.5.1.1 Mittelsenkrechte

Grundvoraussetzung ist, dass man die gewünschte Konstruktion einmal selbst durchführt, wir machen dies anhand der Zwei-Kreis-Konstruktion. Erzeugen Sie daher zwei Punkte *A* und *B* und zeichnen Sie hernach einen Kreis um *A* mit dem Radius *AB* und dann einen Kreis um *B* mit dem Radius *BA*. Beide Kreise werden miteinander geschnitten und durch die beiden Schnittpunkte eine Gerade – die Mittelsenkrechte – gelegt.

Von unserer Konstruktion sind drei Dinge wichtig: Die beiden Punkte *A* und *B* und die erzeugte Mittelsenkrechte. Markieren Sie alle drei Elemente, indem Sie im Bewegen-Modus diese drei Objekte nacheinander mit gedrückter Großschreibtaste anklicken.

Klicken Sie dann auf Bearbeiten / Eigenes Werkzeug erzeugen... und füllen Sie die Leerfelder im erscheinenden Fenster aus: Ihr Makro benötigt einen Namen und Sie können noch eine kurze Beschreibung angeben. Nach dem Klick auf OK erhalten Sie eine weitere graue Konstruktionsschaltfläche.

Löschen Sie alle Objekte auf dem Bildschirm indem Sie auf die Schaltfläche ALLES MARKIEREN klicken und hernach mit dem Radiergummi-Werkzeug (MARKIERTE ELEMENTE LÖSCHEN) alles löschen. Setzen Sie dann wieder zwei Punkte auf die Zeichenfläche; zu diesen beiden Punkten soll nun die Mittelsenkrechte erstellt werden. Klicken Sie also auf Ihre selbst erstellte Schaltfläche, es erscheinen zwei Punkte und die zugehörige Mittelsenkrechte. Jetzt können Sie mit dem ABLÖSEN UND ANKLEBEN-Werkzeug (Schraubenschlüssel) nacheinander die beiden Punkte Ihres Makros auf die vorhandenen Punkte montieren.

2.5.1.2 Regelmäßige *n*-Ecke

Erzeugen Sie Makros, mit denen Sie – jeweils ausgehend von zwei gegebenen Punkten – ein zugehöriges regelmäßiges Dreieck, Viereck (Quadrat) sowie das zugehörige regelmäßige Sechseck erzeugen können.

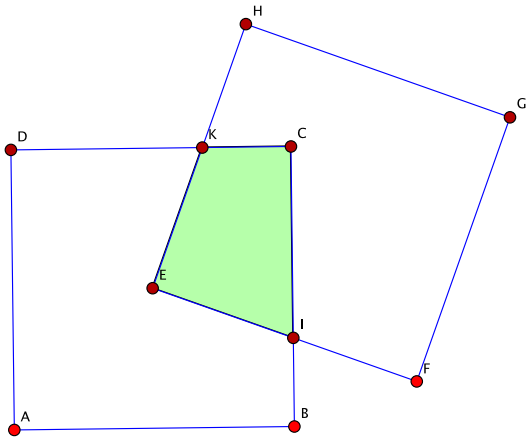
Hernach versuchen Sie sich an einem regelmäßigen Fünfeck:

1. Zeichne zwei Punkte *A* und *B* und lege eine Gerade *a* durch *A* und *B*.
2. Zeichnen einen Kreis um *A* mit dem Radius *AB* und einen Kreis um *B* mit dem Radius *BA*. Beide Kreise schneiden sich (oberhalb von *AB*) im Punkt *F*.
3. Der Schnittpunkt des Lots durch *F* auf *AB* mit *AB* sei der Punkt *G*.
4. Zeichne eine Parallele zu diesem Lot durch den Punkt *A* und schneide diese Parallele mit dem Kreis um *A*. Der oberhalb von *AB* gelegene Schnittpunkt sei *H*.
5. Zeichne einen Kreis um *G* mit dem Radius *GH* und schneide diesen Kreis mit der Geraden *a*; der links von *A* gelegene Schnittpunkt sei *J*.
6. Zeichne einen Kreis um *B* mit dem Radius *BJ*. Der obere Schnittpunkt dieses Kreises mit dem Kreis um *A* sei *E*, der obere Schnittpunkt mit dem Lot *FG* sei *D*.
7. Schneide den Kreis um *D* mit dem Radius *DE* mit dem Kreis um *B*, der rechte Schnittpunkt sei *C*.

Sicher wollen Sie Ihre erzeugten Makros auch speichern. Dies geschieht über SCRIPTING / EIGENE WERKZEUGE SPEICHERN... . In jeder neuen Cinderella-Datei können Sie dann über die Option EIGENE WERKZEUGE LADEN... die in der Makrodatei gesicherten Makros wieder laden.

2.5.1.3 Lösung des Zwei-Quadrate-Problems

Im Mittelpunkt eines Quadrats liegt die Ecke eines gleichgroßen anderen Quadrats. Wie groß ist die von beiden gemeinsam bedeckte Fläche?



Zeichnen Sie zwei Punkte *A* und *B* und vervollständigen Sie diese mit Hilfe Ihres Makros zu einem Quadrat *ABCD*. Über dessen Diagonalen erzeugen Sie seinen Mittelpunkt *E*.

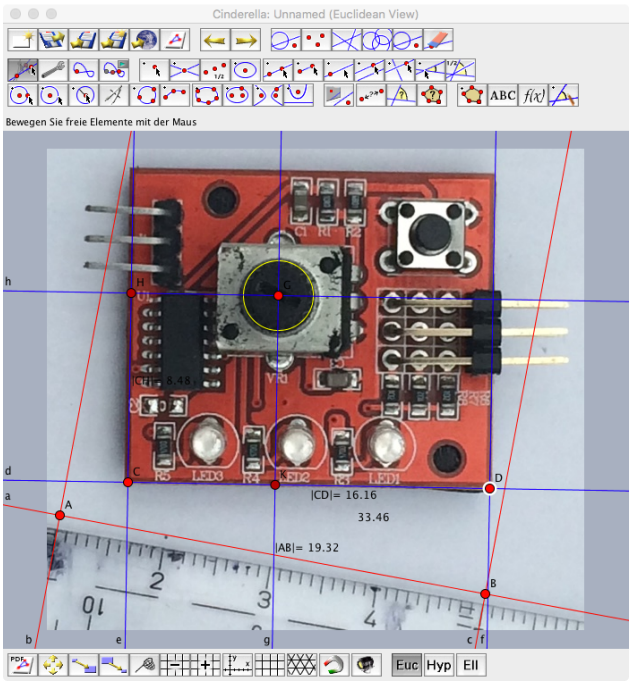
Nehmen Sie die Strecke *AB* in den Zirkel und tragen Sie diesen Kreis von *E* aus ab.

Auf dem Kreis befestigen Sie den Punkt *F* und konstruieren über *EF* wieder mit Hilfe des Makros das Quadrat *EFGH*.

Erzeugen Sie die Schnittpunkte *I* und *K* der beiden Quadrate miteinander, zeichnen Sie das Polygon *EICK* und lassen Sie dessen Flächeninhalt anzeigen. Bewegen Sie *F*, wie ändert sich der Flächeninhalt des Polygons? Sehen Sie den Beweis für dieses Verhalten?

2.5.2 Maße feststellen

Zur Abwechslung mal eine ganz realistische Aufgabenstellung aus dem Arbeitsalltag, die mit Cinderella gelöst werden kann: Sie haben eine kleine elektronische Platine, die in ein Gehäuse eingebaut werden soll. Dazu müssen Sie die exakten Maße der Befestigungsbohrungen und der Potentiometerachse wissen. Eine Maßzeichnung wurde leider nicht mitgeliefert.



Zunächst fotografieren Sie die Platine möglichst unverzerrt. Dies bedeutet, dass Bild- und Filmebene zueinander parallel sein müssen, in diesem Fall erhalten wir eine streckenverhältnistreue Ähnlichkeitsabbildung. Sorgen Sie auch dafür, dass auf dem Bild ein Maßstab (Ausschnitt eines Lineals mit mm-Markierung sichtbar ist.

In Cinderella können Sie mittels ABLAGE / HINTERGRUNDBILD LADEN / AUS DATEI LADEN das Bild auf die Zeichenoberfläche bringen. Je nach der Auflösung Ihres Computers müssen Sie das Bild zuvor in einem Bildbearbeitungsprogramm skalieren, d.h. die Bildgröße (in Pixeln) einstellen.

Das Bild erscheint dann zentriert auf der Arbeitsoberfläche von Cinderella und kann dort nicht weiterbearbeitet oder bewegt werden. Bewegt und skaliert werden kann jedoch die Zeichenebene über dem Bildhintergrund.

Zunächst müssen wir den Ähnlichkeitsfaktor unserer Abbildung bestimmen. Dazu zeichnen wir die Gerade durch zwei Punkte *A* und *B* und positionieren *A* in der Nähe der 1 cm-Markierug des mitfotografierten Lineals und *B* setzen wir in die Nähe der gerade noch sichtbaren 5 cm-

Markierung. Um den Maßstab möglichst exakt zu übernehmen, konstruieren wir durch *A* und *B* jeweils die Senkrechten. Jetzt werden *A* und *B* so bewegt, dass die Senkrechten exakt mit den mm-Markierungen auf dem Lineal fluchten. Die Länge von *AB* kann dann im Cinderella-Maß gemessen werden, im Bild ist dies das Maß |*AB*|=19.32.

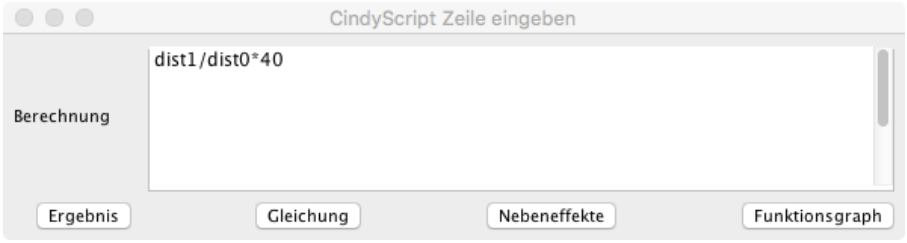
Dann bemaßen wir die Grundkante der Platine: Eine Gerade *CD* legen wir durch die Eckpunkte und mit Senkrechten durch *C* und *D* überprüfen wir die korrekte Lage. Die gemessene Länge der Strecke *CD* beträgt im Bild |*CD*|=16.16 Cinderella-Einheiten.

Sind wie oben angemahnt Bild- und Urbildebene parallel, so handelt es sich um eine streckenverhältnistreue Ähnlichkeitsabbildung und es muss die nachfolgende Verhältnisgleichung gelten:

$$\frac{\textit{Linealmaß in mm}}{\textit{Linealmaß in Cindyeinheiten}} = \frac{\textit{Platinenlänge in mm}}{\textit{Platinenlänge im Cindyeinheiten}}$$

$$\Rightarrow \textit{Platinenlänge in mm} = \frac{\textit{Platinenlänge im Cindyeinheiten} \cdot \textit{Linealmaß in mm}}{\textit{Linealmaß im Cindyeinheiten}}$$

Im obigen Bild beträgt das Linealmaß in Millimetern 40 mm, daher ergibt sich die folgende Berechnung für die Platinenlänge in mm:



Das obige Eingabefenster erhalten Sie über die Schaltfläche BERECHNUNG EINES WERTES. Anstatt *dist1* und *dist0* von Hand einzugeben, klicken Sie stattdessen nur auf die entsprechenden auf der Zeichenoberfläche angezeigten Maße. Das Divisionszeichen und die Multiplikation mit 40 müssen Sie hingegen von Hand eintippen. Beenden Sie die Eingabe mit einem Klick auf die Schaltfläche ERGEBNIS.

Entsprechend ermitteln Sie die weiteren benötigten Maße: Zeichnen Sie einen Kreis um die Potiachse, um deren Mittelpunkt genau zu ermitteln. Von diesem Mittelpunkt aus fällen Sie die Lote auf die Grundlinie *CD* und auf die dazu Senkrechte durch *C* und schneiden diese Lote mit diesen Linien. Messen Sie jeweils von *C* aus die Längen zu diesen Schnittpunkten und rechnen Sie, wie eben gezeigt, diese Längen in Millimeter um. genauso verfahren Sie mit den beiden Bohrlöchern für die Befestigung der Platine. Damit erhalten Sie – ausgehend vom Koordinatenursprung *C* – die Maße für die anzubringenden Bohrungen.

3 Computer-Algebra-Systeme

Als mathematische Werkzeuge sind Computer-Algebra-Systeme (CAS) unverzichtbar. Neben numerischen können Sie auch symbolische Berechnungen vornehmen und mit ihrem großen Funktionsumfang so gut wie jedes mathematische Problem lösen. Für Spezialanwendungen verfügen sie über eine Programmiermöglichkeit, mit deren Hilfe Lösungen für eigene, spezielle Probleme geschaffen werden können. Es ist kein Wunder, dass kommerzielle Algebra-Systeme aufgrund ihrer Leistungsfähigkeit – auch für Bildungseinrichtungen – relativ teuer sind. Glücklicher Weise steht jedoch mit Maxima ein Open-Source-Projekt völlig kostenfrei zur Verfügung, das trotzdem auch professionellen Ansprüchen genügt.

Bitte installieren Sie vorbereitend zu den nächsten Aufgabenstellungen das Programmpaket auf ihrem Rechner. Die Quelle und einige Installationshinweise finden Sie in Moodle.

Bitte beachten Sie, dass es sich bei Maxima nicht nur um ein Programm, sondern um eine Programmgruppe handelt: *Maxima* ist das eigentliche Programm, welches die Berechnungen ausführt. *wxMaxima* ist die Bedieneroberfläche, welche uns ein komfortables Eingabefenster und viele der unzähligen Maxima-Fähigkeiten in Menüs zur Verfügung stellt. Um Funktionen plotten zu können, braucht man das Programm *Gnuplot*. Außerdem werden einige spezielle Schriften (Fonts) mitgeliefert. Auf einem Windowsrechner läuft bei der Installation eine Routine, welche alle benötigten Teile an die richtigen Plätze verschiebt. Auf dem Mac muss man die einzelnen Apps (Maxima, wxMaxima und Gnuplot) aus dem Installationsfenster von Hand in das Verzeichnis Programme verschieben und außerdem die Fonts installieren. Beachten Sie die mitgelieferte Installationsanleitung!

Die Bedieneroberfläche wxMaxima kommuniziert mit Maxima über sogenannte *Ports*. Falls Sie eine Firewall auf Ihrem Rechner installiert haben kann es sein, dass die Firewall diese Portkommunikation aus Sicherheitsgründen unterbindet. Dies erkennen Sie daran, dass wxMaxima zwar startet, aber rechts unten am Bildschirmrand die Meldung erscheint „Maxima wurde beendet“. In diesem Fall müssten Sie in der Firewall die benötigten Ports freischalten.

Erkunden Sie Maxima anhand der Maxima-Anleitung, die Sie ebenfalls von der oben angegebenen Seite erhalten können. Diese Anleitung enthält außerdem eventuell hilfreiche Installationshinweise.

Bitte beachten Sie bei Ihrer Arbeit mit Maxima die folgenden Punkte:

- Vermeiden Sie deutsche Sonderzeichen (Ä, Ö, Ü, ß, ...)
- Schalten Sie die Option AUTOMATISCH PASSENDE KLAMMERN ERZEUGEN in den *Einstellungen* ein. Sie müssen dazu das bereits gesetzte Häkchen wegnehmen und gleich wieder setzen.
- In Maxima können Sie Überschriften und erläuternden Text eingeben. Schauen Sie hierfür im Menü ZELLEN nach.
- Speichern Sie Ihre Arbeit als *wxMaxima-Dokument* (mit der Endung *.wxm) und nicht als *wxMaxima-XML-Dokument* (mit der Endung *.wxmx).
- Nach dem Laden eines vormals gespeicherten Maxima-Arbeitsblatts sollten Sie zuerst ZELLEN / ALLE ZELLEN NEU AUSWERTEN aufrufen.

3.1 IBAN-Bestimmung und Überprüfung

Seit dem 1. Februar 2014 ersetzt die IBAN (International Bank Account Number) die seither übliche Kontonummer. Sie setzt sich in Deutschland aus 22 Stellen zusammen: Zwei einleitende Buchstaben als Länderkennung (DE – Deutschland), zwei Prüfziffern, der seitherigen 8-stelligen Bankleitzahl (BLZ) und der 10-stelligen Kontonummer, die nötigenfalls linksseitig mit Nullen aufgefüllt wird:

```
DEpp bbbb bbbb kkkk kkkk kk
DE68 2105 0170 0012 3456 78
```

Um im Prüfverfahren die Gültigkeit einer IBAN festzustellen, wird die erste Vierergruppe ans Ende gestellt:

```
2105 0170 0012 3456 78DE 68
```

Dann werden die Buchstaben D und E in Zahlen umcodiert: D ist 13 und E ist 14 und alle nun vorhandenen Ziffern werden als 24-stellige Zahl gelesen:

```
210501700012345678131468
```

3.1.1 Prüfverfahren

Diese 24-stellige Zahl wird durch 97 dividiert und zwar so, wie Kinder in der Grundschule dividieren. Dort kennt man ja noch keine Dezimalbrüche; die Division 19 geteilt durch 4 ergibt dort 4 Rest 3. Bei der Ganzzahldivision der IBAN ist nur der Rest interessant, der bei der Division durch 97 übrigbleibt.

Die meisten Programme und Rechner können nicht mit 24-stelligen Zahlen rechnen, Maxima ist hierzu jedoch problemlos in der Lage. Um den Rest einer Ganzzahldivision zu bestimmen, gibt es in Maxima die Funktion mod(), welche eine Ganzzahldivision vornimmt und den Rest dieser Division als Ergebnis zurückliefert. Um den Rest der obigen Grundschulaufgabe in Maxima zu ermitteln, lautet der Aufruf:

```
(%i12) mod(19,4);
(%o12) 3
```

Stellen Sie Ihre eigene IBAN nach dem oben gezeigten Schema um und ermitteln Sie den Rest, den diese so erzeugte Zahl bei Division durch 97 lässt. Untersuchen Sie auf die gleiche Weise weitere IBANs. Es wird Ihnen unschwer auffallen, dass alle gültigen IBANs den Rest 1 liefern.

3.1.2 Ermitteln der Prüfzahl

Eine IBAN laute DEXX 61450030 0000123456, wobei XX der Platzhalter für die Prüfzahl ist.

Setzen Sie die Prüfzahl vorläufig auf 00 und stellen Sie diese IBAN nach dem oben in Aufgabe 1 gezeigten Schema um: DE durch 1314 ersetzen und ans Ende stellen. Ganz zum Schluss wird dann noch die (vorläufige) Prüfzahl 00 angehängt.

Stellen Sie den Rest fest, den diese Zahl bei Division durch 97 lässt. Verändern Sie die Prüfzahl so lange, bis Sie den Rest 1 erhalten. Erarbeiten Sie eine Methode, wie Sie – ausgehend von einer vorläufigen Prüfzahl 00 – die korrekte Prüfzahl bestimmen können.

Dies geschieht, indem man den erhaltenen Rest von 98 subtrahiert. Diese Differenz ist die korrekte Prüfzahl.

3.1.3 Sicherheit des Prüfzahlverfahrens

Untersuchen Sie, wie sicher das Prüfzahlverfahren der IBAN ist. Begründen Sie jeweils Ihre Meinung!

Die Falscheingabe einer Ziffer wird immer sicher erkannt! Um denselben Rest mod 97 zu erhalten, müsste man die zu prüfende Zahl um ein Vielfaches von 97 verändern. Die Vielfachen unserer Ziffern ergeben allerdings nie 97, denn 97 ist eine Primzahl!

Die Falscheingabe von zwei beliebigen Ziffern wird jedoch nicht in jedem Fall sicher erkannt. Durch Manipulation von zwei beliebigen Ziffern kann man die Zahl so verändern, dass sie um ein Vielfaches von 97 verändert wird. Beispielsweise erhöht man die Hunderterstelle um eins und vermindert die Einerstelle um drei.

Bei der Frage, ob Zifferndreher immer sicher erkannt werden betrachten wir zunächst Zifferndreher zweistelliger Zahlen. Manche Dreher fallen gar nicht auf (55), manche haben nur eine kleine Auswirkung auf die Größe der Zahl (45), andere hingegen eine größere. Den größten Einfluss hat ein Dreher der Ziffern 90, hier wird die Größe der Zahl um 81 verändert. Eine Veränderung um 97 – wie es für eine unerkannte Manipulation nötig wäre – ist somit durch das Vertauschen der zwei Ziffern einer zweistelligen Zahl nicht zu erreichen.

Verändert man 900 in 090, so ändert man die Größe der Zahl um 810, dies ist kein Vielfaches von 97. Verändert man 9000 in 0900, so verändert man um 8100 – ebenfalls kein Vielfaches von 97.

Egal welche Ziffern man vertauscht und an welcher Stelle der IBAN sie vertauscht werden, immer wird die Größe der Zahl um maximal 81 und deren Zehnerpotenzen verändert (Zahlen mit zwei führenden Ziffern – max. 81 – ungleich Null und dann nur noch folgende Nullen). Die größte Veränderung, die durch das Vertauschen zweier Ziffern auftreten kann, beträgt somit 81 und deren Zehnerpotenzen. 81 (und all die anderen kleineren Zahlen) sind jedoch weder Teiler noch Vielfache von 97. Um Zehnerpotenzen von 91 zu erhalten, muss man 91 mit 10 (also den Primfaktoren 2 und 5) ggf. mehrfach multiplizieren. 2 und 5 sind aber auch keine Teiler von 97, so dass nie Vielfache von 97 entstehen können, Zifferndreher werden daher immer sicher erkannt!

3.1.4 IBAN aus BLZ und Kontonummer erzeugen

Wir wollen nun erarbeiten, wie wir aus einer Kontonummer und der Bankleitzahl eine gültige IBAN erstellen können. Hierzu müssen uns zunächst die Bankleitzahl und die Kontonummer vorliegen, in Maxima weisen wir beide Angaben eindeutigen Variablen zu:

```
blz:61450050;
kto:740986;
```

Laut den obigen Angaben wird die Kontonummer als zehnstellige Zahl – nötigenfalls linksbündig aufgefüllt mit Nullen – an die BLZ angehängt. Auf mathematischem Weg erledigen wir dies, indem wir zunächst die BLZ mit 10^{10} multiplizieren und dann die Kontonummer addieren:

```
iban1:blz*10^10+kto;
```

An dieses Konstrukt müssen nun die Ziffern der Ländercodierung – im Fall DE die Ziffernfolge 1314 – angehängt werden. Den nötigen Platz hierfür schaffen wir mit einer Multiplikation der seither geschaffenen Zahl mit 10^4 und anschließender Addition von 1314:

```
iban2:iban1*10^4+1314;
```

Schließlich benötigen wir noch am Ende den Platz für die zweistellige Prüfzahl, wir multiplizieren die bisher geschaffene Zahl mit 10^2, damit setzen wir gleichzeitig die (vorläufige) Prüfziffer auf 0:

```
iban3:iban2*10^2;
```

Von der so erstellten Zahl ermitteln wir den Rest bei Division durch 97:

```
rest:mod(iban3,97);
```

Wie bereits weiter oben dargestellt, muss der so ermittelte Rest von 98 subtrahiert werden, um die korrekte Prüfziffer zu erhalten:

```
pz:98-rest;
```

Um die generierte IBAN nun korrekt ausgeben zu können, muss die Prüfzahl vor die 18-stellige BLZ mitsamt der Kontonummer geholt werden. Dies erreichen wir, indem wir die Prüfzahl mit

10^{18} multiplizieren und dann die in der Variablen `iban1` abgelegte Kombination aus BLZ und Kontonummer addieren:

```
iban4:pz*10^18+iban1;
```

Jetzt muss noch das Länderkennzeichen DE davorgesetzt werden. Da es sich dabei um einen Text (string) und keine Zahlen handelt, können wir dies nicht auf der mathematischen Ebene erreichen, wir benötigen hierfür Stringoperationen – also spezielle Funktionen in Maxima, welche auf Zeichenketten operieren. Zunächst einmal müssen wir die in `iban4` generierte Zahl in einen Text umwandeln:

```
iban_text:string(iban4);
```

Für eine gültige IBAN müssen wir jetzt noch entscheiden, ob die ermittelte Prüfziffer ein- oder zweistellig ist. Je nachdem muss vor den generierten String `iban_text` der Präfix „DE“ oder der Präfix „DE0“ gestellt werden. Ob die Prüfziffer ein- oder zweistellig ist können wir durch einen Vergleich mit der Zahl 10 ermitteln und daraus eine bedingte Anweisung generieren:

```
praefix:if pz<10 then "DE0" else "DE";
```

Dann können wir die den Prüfix mittels einer Stringverkettung (*Konkatenation*) davorsetzen:

```
concat(praefix,iban_text);
```

3.1.5 Funktion für IBAN-Erstellung

Wenn Sie die eben dargestellten Maxima-Anweisungen zur Ermittlung einer IBAN aus einer BLZ und Kontonummer fortfolgend in Maxima eingegeben haben, dann können Sie das Verfahren ein weiteres Mal durchlaufen, indem Sie die Variablen für BLZ und Kontonummer neu belegen und alle Anweisungen nacheinander anklicken und jeweils <Shift>-<RETURN> drücken.

Noch bequemer wird die Erstellung einer IBAN, wenn wir die aufgeführten Schritte Maxima als *Befehlsliste* übergeben und Maxima nur anweisen, dass es diese Liste abarbeiten soll. Eine solche *Befehlsliste* nennt man allgemein ein *Programm*, in Maxima – das ja ausschließlich aus Funktionen aufgebaut ist – spricht man von einer *selbstdefinierten Funktion*. Da Maxima diesen Funktionsgedanken sehr konsequent mathematisch umsetzt, sollten wir uns zunächst vergegenwärtigen, was die Mathematik unter einer Funktion versteht:

Eine Funktion ist eine linkstotale und rechtseindeutige Zuordnung zwischen zwei Mengen, wobei diese Mengen nicht notwendiger Weise verschieden sein müssen.

In aller Regel haben Sie es seither mit Funktionen zu tun gehabt, in denen die Menge der reellen Zahlen R auf sich selbst abgebildet wurde. Eine Funktion ist aber auch die Zuordnung von einer Masse zu einem Preis oder die Zuordnung einer Flugnummer zur Abflugzeit oder die Zuordnung eines Geburtsdatums zu jeder Person usw.

Eher weniger sind seither mehrdimensionale Funktionen in Ihrem Fokus gestanden: So kann man jedem Punkt der Erdoberfläche – ausgedrückt in *x*- und *y*-Koordinaten – dessen Höhe über dem Meeresspiegel zuordnen. Man kann jedem Punkt der Atmosphäre – angegeben durch *x*-, *y*- und *z*-Koordinate – seinen Luftdruck zuordnen. Eine Funktion kann also – etwas umgangssprachlich ausgedrückt – mehr als einen Eingabewert benötigen, um einen Funktionswert zu ermitteln. In unserem Fall der IBAN-Erstellung benötigen wir zwei Werte, nämlich die Kontonummer und die BLZ. Solche Eingabewerte, die eine Funktion benötigt, um ein Funktionsergebnis zu generieren, nennt man *Aufrufparameter* der Funktion.

Wir haben oben vereinfachend aber treffend von *Befehlslisten* gesprochen. Es ist naheliegend, dass man mehrere und unterschiedliche Befehlslisten erstellen möchte, daher muss jede Befehlsliste eindeutig über einen Namen unterschieden und angesprochen werden können. Damit haben wir die notwendigen Voraussetzungen geklärt, um in Maxima eigene Funktionen zu definieren:

Die Funktion benötigt einen eindeutigen Namen! Bei der Namensvereinbarung sind Sie relativ frei, Sie sollten aber keine schon in Gebrauch befindlichen Namen verwenden, weil sonst die seitherige Funktion überschrieben wird. Da Maxima aus dem angelsächsischen Raum stammt, sind Sie mit deutschen Namen auf der sicheren Seite. Wir wollen unsere Funktion `iban` nennen.

Direkt nach dem Funktionsnamen müssen in runden Klammern die Aufrufparameter der Funktion angegeben werden, in unserem Fall die BLZ und die Kontonummer.

Dass der gewählte Name eine Funktion bezeichnen soll, wird mit dem speziellen Zuweisungsoperator `:=` bestimmt.

Wenn eine selbst erstellte Funktion aus mehreren Befehlen besteht, so müssen diese mit der Anweisung `block()` als Block gekennzeichnet werden. In das runde Klammernpaar dieser Blockanweisung kommen danach die einzelnen Befehle. Das Gerüst unserer Funktion sieht damit folgendermaßen aus:

```
iban(blz,kto):=block(


)
```

Die einzelnen Befehle sind genau diejenigen, die wir im vorhergehenden Abschnitt erarbeitet haben, wir schreiben sie genau in der obigen Reihenfolge zeilenweise und durch Kommata getrennt in die Block-Anweisung:

```
iban(blz,kto):=block(
iban1:blz*10^10+kto,
iban2:iban1*10^4+1314,
iban3:iban2*10^2,
rest:mod(iban3,97),
pz:98-rest,
iban4:pz*10^18+iban1,
iban_text:string(iban4),
praefix:if pz<10 then "DE0" else "DE",
concat(praefix,iban_text)
);
```

Das Ergebnis des letzten Befehles einer Maxima-Funktion ist automatisch das Funktionsergebnis, in unserem Fall die IBAN als Text.

Die Funktion kann nun beliebig oft unter ihrem Namen und mit der Angabe von BLZ und Kontonummer aufgerufen werden.

Die oben erstellte Funktion verwendet einige Variablen als Speicher für Zwischenwerte. Damit diese nicht mit eventuell außerhalb der Funktion verwendeten Variablen gleichen Namens kollidieren, sollten grundsätzlich alle innerhalb einer Funktion verwendeten Variablen als lokale Variable deklariert werden. Dies macht man, indem man eine Liste dieser Variablen als erstes Statement und erste Zeile nach der block()_Anweisung einfügt:

```
[iban1,iban2,iban3,rest,pz,iban4,iban_text],
```

Die Aufrufparameter einer Funktion werden von Maxima automatisch als lokale Variablen deklariert, sie dürfen keinesfalls in diese Liste der lokalen Variablen aufgenommen werden!

3.1.6 Funktion zur Überprüfung einer IBAN

Schreiben Sie eine Funktion, welche eine vorliegende IBAN auf deren Gültigkeit überprüft! Die zu überprüfende IBAN wird der Funktion als String übergeben.

Zur Lösung des Problems benötigen Sie die folgenden Funktionen:

`substring(text1,a,e)`	liefert einen `Ausschnitt` aus der übergebenen Zeichenkette beginnend mit dem *a*-ten bis zum *e*-ten Zeichen, wobei kurioserweise das *e*-te Zeichen nicht mehr geliefert wird! Lässt man die Angabe *e* weg, so wird der komplette String ab dem *a*-ten Zeichen geliefert.
`eval_string(„3.14“)`	wandelt eine als String übergebene Zahl wieder in den zugehörigen Zahlenwert um.
`is(`*`log. Ausdruck`*`)`	Wertet einen logischen Ausdruck aus und gibt als Ergebnis *true* oder *false* zurück. Der logische Ausdruck ist in diesem Fall eine Überprüfung auf Gleichheit in der Form $x = 7$.

Gehen Sie wieder so vor, dass Sie die einzelnen Schritte, die zur Überprüfung einer IBAN nötig sind, zunächst wieder im Direktmodus ausprobieren und optimieren. Erst hernach fassen Sie diese Schritte in einer Funktion zusammen.

3.2 Der Einkommensteuertarif

Das deutsche Steuerrecht teilt für die Bestimmung der Einkommensteuer das zu versteuernde Einkommen in fünf Tarifzonen ein. Für die Berechnung der Einkommensteuer galt im Jahr 2019:

Die **erste Zone**, der sogenannte „Grundfreibetrag“ reicht von 0 bis 9.168 € Jahreseinkommen. In diesem Einkommensintervall fallen überhaupt keine Steuern an:

$$steuer1(x) := 0$$

Die **zweite Zone** reicht von 9.169 € bis 14.254 €. In dieser Zone berechnet sich die zu entrichtende Einkommensteuer in Euro für das zu versteuernde Einkommen x € nach der Formel:

$$steuer2(x) := \left(980{,}14 \cdot \frac{(x-9168)}{10000} + 1400\right) \cdot \frac{(x-9168)}{10000}$$

Die **dritte Zone** erstreckt sich von 14.255 € bis 55.960 €. In diesem Bereich lautet die Formel:

$$steuer3(x) := \left(216{,}16 \cdot \frac{(x-14254)}{10000} + 2397\right) \cdot \frac{(x-14254)}{10000} + 965{,}58$$

In der **vierten Zone** von 55.961 € bis 265.326 € errechnet sich die Einkommensteuer nach der Formel:

$$steuer4(x) := 0{,}42 \cdot x - 8780{,}90$$

Die **fünfte Zone** schließlich gilt für alle Jahreseinkommen ab 265.327 €. Die hierfür zutreffende Steuer wird folgendermaßen errechnet:

$$steuer5(x) := 0{,}45 \cdot x - 16740{,}68$$

Hinweis: Für die Berechnung der Einkommensteuer wird grundsätzlich das zu versteuernde Einkommen auf den nächst niedrigen vollen Eurobetrag abgerundet. Dies müssen Sie in Ihren Funktionen allerdings nicht berücksichtigen!

Die Einkommenssteuer wird i.d.R. jährlich angepasst: Es verschieben sich die Grenzen zwischen den einzelnen Zonen, damit ändern sich die einzelnen Formeln. Die jeweils aktuellen Steuerformeln können Sie nötigenfalls im Internet nachschlagen.

Erstellen Sie in Maxima die fünf Funktionen `steuer1(x)` bis `steuer5(x)`, mit welchen Sie die Jahressteuer für die jeweiligen Einkommenszonen berechnen können. Hierbei kommt es im Wesentlichen nur darauf an, die obigen Terme korrekt in Maxima einzugeben!

Errechnen Sie die Steuern für Jahreseinkommen von 5.000 €, 12.000 €, 48.000 €, 125.000 € und 265.000 €. Weil wir diese Werte öfter verwenden werden, speichern Sie diese Zahlen in den Variablen `E1` bis `E5`. Vergleichen Sie die errechneten Steuerwerte mit denjenigen Ihrer Nachbarn!

Bei der Berechnung der Steuern mussten Sie eben für jedes Einkommen die zugehörige Steuerfunktion auswählen. Erstellen Sie daher nun eine Funktion `steuer(x)`, welche dies automatisch durchführt und für jedes beliebige Einkommen die Steuer nach der zugehörigen Funktion berechnet. Dies bedeutet, dass in der Funktion `steuer(x)` eine Fallunterscheidung getroffen werden muss in der Form:

Wenn $x < 9169$ dann steuer1(x) sonst

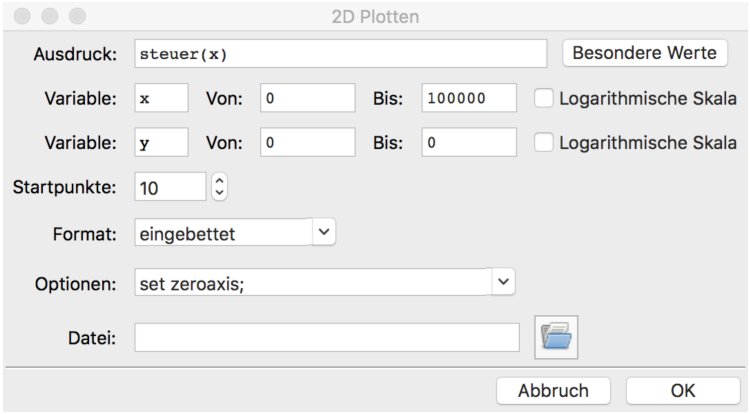
wenn $x < 14255$ dann steuer2(x) sonst

wenn $x < 55961$ dann steuer3(x) sonst

usw.

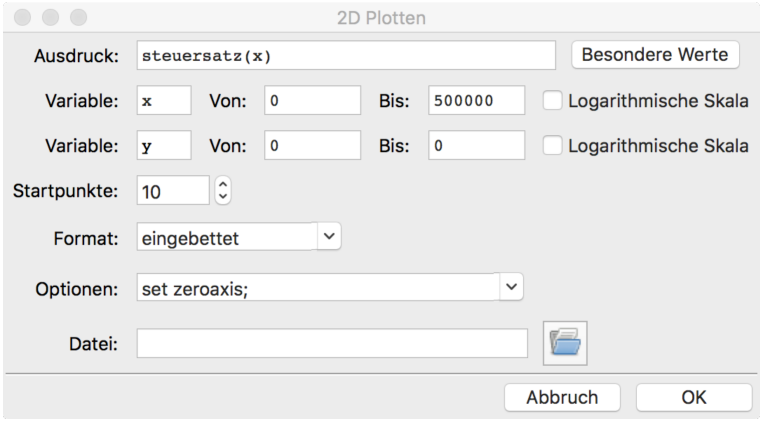
Wenn ... dann ... sonst heißt in Maxima: `if ... then ... else`. Ihre fertige Funktion überprüfen Sie, indem Sie die mit ihr die Steuern für die oben angegebenen Jahreseinkommen erneut berechnen.

Lassen Sie den Graph der Steuerfunktion in Maxima plotten (PLOTTEN / PLOT 2D...):



Nun soll der Steuersatz angegeben werden. Dafür müssen Sie berechnen, welcher Prozentsatz eines gegebenen Einkommens für die Steuer aufgewendet werden muss, letztlich ist es ausreichend, dafür den Quotienten aus `steuer(x)/x` zu bilden. Erstellen Sie die Funktion `steuersatz(x)`, die genau dieses macht. Berechnen Sie mit Hilfe dieser Funktion die Steuersätze der oben angegebenen Jahreseinkommen und vergleichen Sie Ihre Werte mit denjenigen Ihrer Kommilitonen.

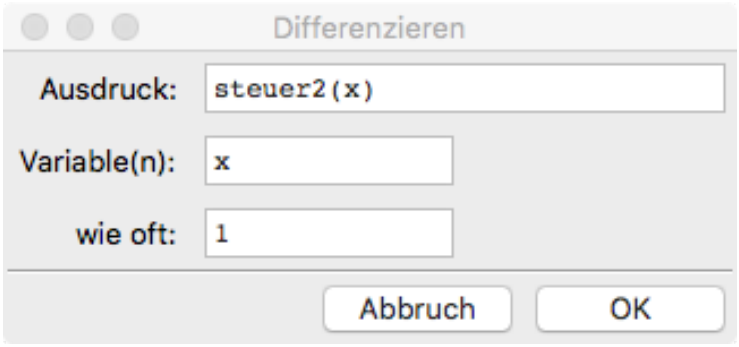
Lassen Sie plotten, wie sich der Steuersatz mit dem Einkommen verändert:



Ein Steuerpflichtiger mit einem Jahreseinkommen von x € erhält eine Gehaltserhöhung von Δx €. Damit steigt gleichzeitig auch seine Steuerlast um Δy € an. Jeder mehr verdiente Euro führt also zu einer Erhöhung der Steuerlast und der Quotient aus Δy (Steuererhöhung) zu Δx (Lohnzuwachs) nennt man den *Spitzensteuersatz*. Welcher mathematische Begriff aus der Analysis entspricht genau diesem *Spitzensteuersatz*?

In der Analysis bezeichnet der Quotient $\frac{\Delta y}{\Delta x}$ den Differenzenquotienten und dessen Grenzwert ist die Ableitung. Um den Spitzensteuersatz bestimmen zu können, benötigen wir daher die Ableitungen unserer Steuerfunktionen `steuer1(x)` bis `steuer5(x)`. Erfreulicherweise brauchen Sie nun nicht Ihre Analysiskenntnisse aus der Oberstufe hervorzukramen, um die Ableitungen zu berechnen, dies erledigt Maxima für uns:

Um die Ableitung einer Funktion zu bestimmen, rufen Sie in Maxima die Menüoption RECHNEN / DIFFERENZIEREN... auf:



Maxima ermittelt daraufhin den zugehörigen Ableitungsterm. Ermitteln Sie die weiteren Ableitungen für die Funktionen `steuer3(x)` bis `steuer5(x)`.

Da wir die Spitzensteuersätze (und damit die Ableitungen) für bestimmte Einkommen berechnen wollen, benötigen wir nicht nur die Terme, sondern die Ableitungsfunktionen. Als Ergebnis der Menüaufrufe hat Maxima die zugehörigen Befehle geschrieben, z.B.:

```
diff(steuer2(x),x,1);
```

Editieren Sie diese Zeile folgendermaßen:

```
define(spitze2(x),diff(steuer2(x),x,1));
```

Damit erhalten Sie die Ableitungsfunktion `spitze2(x)`. Tun Sie entsprechendes für die Ableitungsfunktionen `spitze3(x)` bis `spitze5(x)`.

Schließlich erstellen Sie eine Funktion `spitze(x)`, welche den Spitzensteuersatz für ein beliebiges Einkommen berechnet. Gehen Sie dabei genauso vor wie bei der Erstellung der Funktion `steuer(x)`, wobei Sie hier den Spitzensteuersatz in der ersten Tarifzone direkt auf 0 setzen können.

Vergleichen Sie den (Durchschnitts-)Steuersatz mit dem Spitzensteuersatz der oben genannten Einkommen!

Stellen Sie schließlich den Spitzensteuersatz im Bereich von 0 bis 300.000 € graphisch dar.

3.3 Newton-Verfahren

Die Funktionsweise des Newton-Verfahrens zur Ermittlung von Nullstellen haben wir bereits geometrisch mit Cinderella dargestellt. Hier soll es nun um die rechnerische Umsetzung mit Hilfe von Maxima gehen. Wir verwenden zur Darstellung dieselbe Funktion

$$f(x) = 0.3x^3 + 0.9x^2 - 1.3x - 0.5$$

3.3.1 Durchführung im Direktmodus

Diese Funktion wird zuerst global definiert:

```
f(x):=0.3*x^3+0.9*x^2-1.3*x-0.5
```

Für einen orientierenden Überblick plotten wir zuerst den Funktionsgraphen:

```
wxplot2d([f], [x,-5,5])
```

Für das Newton-Verfahren benötigen wir die Ableitung der Funktion, diese definieren wir ebenfalls als Funktion:

```
define(f_strich(x),diff(f(x),x,1))
```

Vom Funktionsgraphen wissen wir, dass eine Nullstelle in der Nähe von 2 liegt:

```
x0:2
```

Jetzt sind alle Vorarbeiten erledigt und wir können das Newtonverfahren beginnen. Zunächst benötigen wir die Steigung der Tangenten im Punkt x_0, müssen also die Ableitung an der Stelle x_0 berechnen:

```
m:f_strich(x0)
```

Daraus bestimmen wir über die Punkt-Steigungs-Form die Tangentengleichung und setzen diese gleich Null:

```
t_gl:m*x+f(x0)-m*x0=0
```

Den Schnittpunkt der Tangenten mit der x-Achse erhalten wir, indem wir die Tangentengleichung lösen:

```
loes:solve(t_gl,x)
```

Diese Lösung ist unser neuer Näherungswert x_0. Bevor wir allerdings x_0 auf den neu errechneten Wert setzen, sollten wir uns den alten Wert merken:

```
x0_alt:x0
```

Den Zahlenwert der Lösung erhalten wir über die Funktion `rhs()`, wir speichern diesen in x_0:

```
x0:rhs(loes[1])
```

Weil wir uns den alten x_0-Wert gemerkt haben, können wir nun die Differenz und damit die erreichte Annäherung berechnen:

```
dx:abs(x0-x0_alt)
```

Solange diese Differenz eine frei gewählte Größe überschreitet, führen wir die Schritte ab *„Jetzt sind alle Vorarbeiten erledigt..."* nochmals wiederholend durch, bis die Differenz klein genug geworden ist. Notieren Sie die schließlich erhaltene Nullstelle.

Ermitteln Sie aus dem Funktionsgraphen Näherungswerte für die anderen Nullstellen und führen Sie das Newtonverfahren wiederholt durch, um auch diese weiteren Nullstellen zu bestimmen.

Alle Nullstellen liefert uns die in Maxima-Funktion `allroots()`:

```
allroots(f(x)=0)
```

Vergleichen Sie Ihre Ergebnisse mit den von Maxima errechneten Nullstellen.

3.3.2 Erstellen einer eigenen Funktion

Wie im obigen Beispiel der Erstellung einer IBAN aus BLZ und Kontonummer ist mit diesen einzelnen Schritten im Direktmodus die eigentliche Arbeit geleistet. Natürlich soll von der dar-

aus zu erstellenden Funktion auch die Wiederholung der Schritte solange vorgenommen werden, bis die Differenz ein bestimmtes Maß unterschreitet. Diese Wiederholmöglichkeiten machen erst das Programmieren zu einer schlagkräftigen Methode und jede Programmiersprache enthält Befehle zum Wiederholen einzelner Programmabschnitte, sogenannte *Schleifen*.

Im vorliegenden Fall soll die Befehlssequenz solange durchlaufen werden, bis der Unterschied *dx* aus altem und neuem Näherungswert einen bestimmten Wert unterschreitet. Umgangssprachlich könnte man formulieren:

Solange dx größer als 0.0001 ist, rechne eine weitere Näherung.

In der Syntax von Maxima lautet der entsprechende Befehl:

```
while dx>0.0001 do (...)
```

Die auszuführenden Anweisungen werden, durch Kommata getrennt, von den runden Klammern umfasst. Damit die Schleife überhaupt durchlaufen wird, müssen Sie vor dem Start der Schleife *dx* auf einen beliebigen größeren Wert setzten, beispielsweise `dx:1`.

Damit kann die Funktion `newton(x0)` erstellt werden, die für eine global definierte Funktion $f(x)$ eine Nullstelle ausgehend von der Stelle x_0 annähert. Vergessen Sie nicht, die innerhalb der Funktion verwendeten Variablen in die Liste der lokalen Variablen einzutragen! Damit Ihnen diese Funktion den Näherungswert tatsächlich liefert und nicht nur die Bestätigung `done` als Bestätigung dafür ausgibt, dass die Schleife durchlaufen wurde, müssen Sie in der Funktion nach dem Abschluss der Schleife dafür sorgen, dass der dann aktuelle Wert von `x0` ausgegeben wird.

Nähern Sie ebenso die anderen Nullstellen der gegebenen Funktion $f(x)$ an und untersuchen Sie weitere, beliebige Funktionen $f(x)$ auf deren Nullstellen mit Hilfe Ihrer Newton-Funktion. Überschreiben Sie dazu die anfangs definierte Funktion $f(x)$ in Maxima, indem Sie einen anderen Funktionsterm eingeben. Lassen Sie sich den Graphen plotten, entnehmen sie daraus die Startstellen für das Newton-Verfahren und rufen Sie Ihre Funktion nacheinander mit diesen Startstellen auf. Überprüfen Sie die erhaltenen Ergebnisse, indem Sie die gegebene (mathematische) Funktion mit der Maxima-Funktion `realroots()` auf deren Nullstellen überprüfen.

3.4 RSA-Verschlüsselung

Seit es Menschen gibt versuchen diese, manche Botschaften vor anderen geheim zu halten. So wurden in der Vergangenheit immer wieder Verschlüsselungsmethoden entwickelt und geknackt, die Entwicklung heutiger Computer ist zu einem großen Teil für die Ver- und Entschlüsselung von Nachrichten im Zweiten Weltkrieg vorangetrieben worden. Während in der Vergangenheit *symmetrische Verschlüsselungsverfahren* verwendet worden sind, bei denen derselbe Schlüssel zum Ver- und Entschlüsseln verwendet wurde, sind seit den 70er Jahren des vorigen Jahrhunderts *asymmetrische Verschlüsselungsverfahren* entwickelt worden, bei denen der Schlüssel für das Verschlüsseln einer Nachricht allgemein bekannt ist (public key) und der Schlüssel für das Entschlüsseln nur der berechtigten Person vorliegen muss (private key). Das erste asymmetrische Verschlüsselungsverfahren wurde 1977 von Ronald L. Rivest, Adi Shamir und Leonard M. Adleman entwickelt und nach ihnen RSA-Verfahren genannt. Dieses Verfahren fußt im Prinzip darauf, dass es sehr leicht ist, durch Multiplikation zweier Primzahlen eine Zahl zu erzeugen, es aber selbst für leistungsfähige Computer nur mit großem zeitlichen Aufwand möglich ist, umgekehrt die Primfaktorzerlegung dieser Zahl zu ermitteln.

Solche Funktionen, bei denen eine Richtung leicht, die andere schwierig zu berechnen ist, bezeichnet man als *Einwegfunktionen* (engl. *one-way function*). Spezielle Einwegfunktionen sind *Falltürfunktionen* (engl. *trapdoor one-way function*), die mit Hilfe einer Zusatzinformation auch rückwärts leicht zu berechnen sind. Wenn Sie beispielsweise wissen, dass 281 ein Faktor von 93011 ist, können Sie den anderen Faktor 331 leicht berechnen. Nachfolgend wird das Prinzip des RSA-Verfahren deutlich vereinfacht dargestellt in dem Sinne, dass für diese Darstellung nur sehr kleine Zahlen verwendet werden. Es wird daher gleich an dieser Stelle darauf hingewiesen, dass mit solchen Zahlen natürlich keine wirklich sichere Verschlüsselung möglich ist, hierfür sind Primzahlen mit enormer Stellenzahl nötig: Zur sicheren Verschlüsselung werden typischerweise mindestens 600-stellige Zahlen empfohlen.

3.4.1 Öffentlicher Schlüssel

Für die Generierung des öffentlichen Schlüssels sucht man sich zwei Primzahlen p und q. In der Realität wird man natürlich möglichst große Zahlen aussuchen, wir verwenden für die Darstellung des Prinzips die Zahlen

$p = 11$ und $q = 17$.

Durch Multiplikation der beiden erhalten wird das sogenannte *RSA-Modul*

$N = 11 \cdot 17 = 187$

Dann wird die *Euler'sche φ-Funktion* dieses RSA-Moduls N berechnet. Die Euler'sche Phi-Funktion ist eine zahlentheoretische Funktion: Sie ordnet jeder natürlichen Zahl n die Anzahl der natürlichen Zahlen a von 1 bis n zu, die zu n teilerfremd sind, für die also $ggT(a, n) = 1$ ist. Um beispielsweise φ(8) zu bestimmen muss man überprüfen...

ggT(1, 8) = 1 ggt(2, 8) = 2 **ggt(3, 8) = 1**, ggt(4, 8) = 4, **ggt(5, 8) = 1**
ggt(6, 8) = 2 **ggt(7, 8) = 1** ggt(8, 8) = 8

Es gibt also insgesamt 4 Zahlen (die 1, die 3, die 5 und die 7), die zu 8 teilerfremd sind, φ(8) ist also gleich 4. Bestimmen Sie auf dieselbe Art und Weise φ(15)! Bestimmen Sie weiter die Euler'sche φ-Funktion von Primzahlen. Was fällt ihnen auf? Außerdem gilt, dass die Euler'sche φ-Funktion für zwei teilerfremde Zahlen m, n multiplikativ ist:

$\varphi(m \cdot n) = \varphi(m) \cdot \varphi(n)$

Für unsere obigen Zahlen 8 und 15, die ja zueinander teilerfremd sind, gilt demnach

$\varphi(120) = \varphi(8 \cdot 15) = \varphi(8) \cdot \varphi(15)$

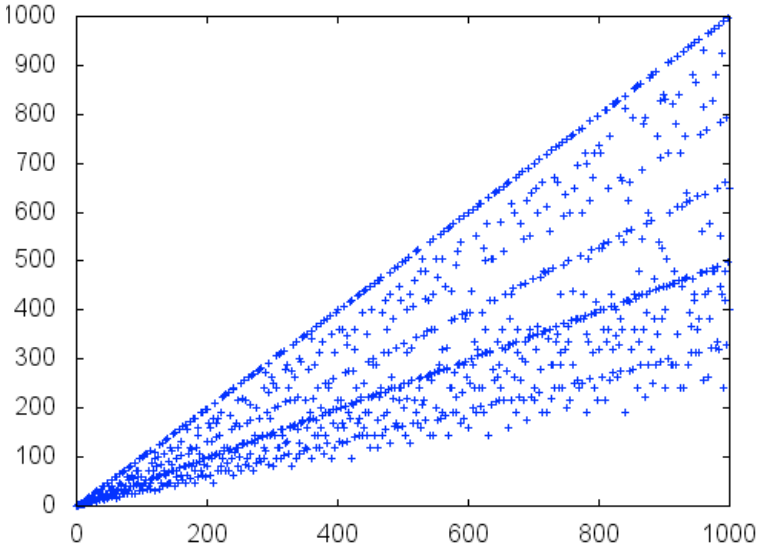
Aus beidem – der Multiplikativität und der besonderen φ-Werte von Primzahlen – folgt, dass die Euler'sche φ-Funktion von N folgendermaßen berechnet werden kann:

$$\varphi(N) = (p - 1) \cdot (q - 1)$$

In unserem Beispiel gilt somit

$$\varphi(N) = (11 - 1) \cdot (17 - 1) = 160.$$

In Maxima ist die Euler'sche Phi-Funktion als `totient(n)` implementiert. Der Plot der φ-Funktion liefert ein interessantes Bild:



Da die Funktion `totient()` eine diskrete Ganzzahlfunktion ist, ist für diesen Plot etwas Vorarbeit nötig. Zunächst müssen die Wertepaare im gewünschten Bereich erstellt werden. Dies geht am schnellsten über die Funktion `create_list()`:

```
wertepaare:create_list([n,totient(n)],n,1,100);
```

Damit werden die diskreten x-y-Wertepaare erstellt und in der Liste `wertepaare` abgelegt. Beim Plotten muss dann mit angegeben werden, dass es sich um diskrete Werte handelt:

```
wxplot2d([discrete,wertepaare],[x,0,100],[style,[points]]);
```

Nun benötigt man weiter eine beliebige, aber zu $\varphi(N) = 160$ teilerfremde Zahl e, die zwischen 1 und $\varphi(N)$ liegt, wir wählen 29.

$$e = 29$$

Dies ist eine Primzahl; damit gehen wir sicher, dass kein Faktor von 29 in 160 enthalten ist, außerdem wissen wir, dass 29 selbst kein Faktor von 160 ist.

Mit den beiden Zahlen

$$e = 29 \text{ und } N = 187$$

haben wir den *öffentlichen Schlüssel* erzeugt, mit dem jeder, der uns Nachrichten senden will, diese verschlüsseln kann.

3.4.2 Verschlüsseln der Nachricht

Um eine Nachricht m zu verschlüsseln, verwendet der Absender die Kongruenz

$$c \equiv m^e \pmod{N}$$

Die Nachricht m muss dabei eine Zahl kleiner als das RSA-Modul N (in unserem Fall 187) sein. Diese Kongruenz besagt, dass man die zu verschlüsselnde Zahl m mit e potenziert und dann ermittelt, welcher Rest bleibt, wenn man diese Potenz durch das RSA-Modul dividiert.

Machen wir ein Beispiel:

Die zu verschlüsselnde Nachricht sei 7. Wir berechnen daher zuerst die Potenz

$7^{29} = 321990575581317972683760 7$

und ermitteln dann den Rest dieser Potenz bei Division durch 187:

$\mathrm{mod}(321990575581317972683760 7, 187) = 74$

Mit unserem öffentlichen Schlüssel wird somit die Zahl 7 in 74 codiert.

3.4.3 Privater Schlüssel

Verschlüsselt wird, indem die Nachricht mit *e* potenziert und dann der Rest mod *N* bestimmt wird. Entschlüsselt wird umgekehrt, indem die verschlüsselte Nachricht mit der Zahl *d* potenziert und dann der Rest mod *N* bestimmt wird. Dieser Exponent *d* ist der *private Schlüssel*, der geheim gehalten werden muss!

Die beiden Exponenten *e* und *d* hängen voneinander ab, *d* muss das *multiplikativ inverse Element* zu *e* bezüglich der Restklassenbildung mod φ(*N*) sein. Inverse Elemente kennen Sie bereits aus den Grundrechenarten, es sind – etwas vereinfacht gesprochen – diejenigen Zahlen, mit denen man eine Operation rückgängig machen kann. Haben Sie beispielsweise zu einer Zahl 7 addiert, so können Sie durch Addition von –7 zur erhaltenen Summe die ursprüngliche Zahl wieder erhalten. Bezüglich der Addition sind also Zahl und Gegenzahl zueinander invers. Bei der Multiplikation sind Zahl und Kehrwert zueinander invers. Etwas mathematischer ausgedrückt müssen Zahl und ihr Inverses miteinander verknüpft immer das *neutrale Element* der Operation ergeben:

Das neutrale Element der Addition ist 0, daher sind *n* und –*n* additiv invers zueinander. Das neutrale Element der Multiplikation ist 1, deshalb sind *n* und $\frac{1}{n}$ multiplikativ invers zueinander. Das neutrale Element bei der Restklassenmultiplikation ist ebenfalls Eins, zwei Elemente *e* und *d* sind daher dann bezüglich der Restklassenmultiplikation zueinander invers, wenn das Folgende gilt:

$$e \cdot d \equiv 1 \bmod m$$

So sind beispielsweise 3 und 7 zueinander multiplikativ invers mod 10, denn es gilt:

$$3 \cdot 7 \equiv 1 \bmod 10$$

Allgemein kann man multiplikativ Inverse durch Aufstellen einer diophantischen Gleichung finden. Die Kongruenz

$$e \cdot d \equiv 1 \bmod m$$

heißt, dass das Produkt aus *e* und *d* bei Division durch *m* den Rest 1 lässt. Anders ausgedrückt ist das Produkt aus *e* und *d* ein Vielfaches von *m* plus 1:

$$e \cdot d = k \cdot m + 1$$

Durch Umstellen erhält man die diophantische Gleichung

$$e \cdot d + (-k) \cdot m = 1$$

In Maxima kann das multiplikativ Inverse einer Restklassenmultiplikation durch die Funktion `inv_mod(n,m)` bestimmt werden, wobei *n* die Zahl ist, deren Inverses man sucht und *m* den Modul angibt:

```
(%i7)    inv_mod(7,10);
(%o7)    3
```

In unserem aktuellen Zusammenhang suchen wir das zu $e = 29$ multiplikativ Inverse *d* bezüglich des Moduls $\varphi(N) = 160$:

```
(%i8)    inv_mod(29,160);
(%o8)    149
```

Das Ergebnis lautet somit 149.

Die Schwierigkeit für Unberechtigte beim Ermitteln des privaten Schlüssels besteht darin, dass man für die Ermittlung des Moduls $\varphi(N) = (p-1)(q-1)$ die beiden Ausgangsprimzahlen p und q benötigt. Man kennt wohl aus dem öffentlichen Schlüssel das Produkt der beiden Primzahlen N, aber eben nicht die Primzahlen selbst. Man kann natürlich auch direkt die Phi-Funktion auf N anwenden, aber dies ist für große N ebenfalls ein sehr zeitaufwändiges Verfahren.

3.4.4 Entschlüsseln der Nachricht

Die erhaltene Nachricht kann nun mit der Kongruenz

$$m \equiv c^d \mod N$$

wieder entschlüsselt werden.

Als verschlüsselte Nachricht haben wir 74 erhalten. Dies wird mit dem privaten Schlüssel 149 potenziert:

$$74^{149} = 32773757799285021060905208222 7[219 \text{ digits}]82100424171221638254860592 7424$$

und von dieser Zahl der Rest bei Division durch 187 bestimmt, was die unverschlüsselte Nachricht 7 wieder zutage fördert.

3.4.5 RSA-Verschlüsselung mit Maxima

3.4.5.1 Gliederung eines Maxima-Arbeitsblatts

Bei umfangreichen Arbeitsblättern kann man leicht den Überblick verlieren. Deshalb bietet Maxima die Möglichkeit einer Gliederung. Es können Titelzeilen eingegeben und Kapitelüberschriften samt –unterüberschriften erzeugt werden. Auch kann jederzeit normaler Text als Kommentar hinzugefügt werden. Die dafür nötigen Optionen finden sich im Menü ZELLEN.

3.4.5.2 Vorarbeiten

Alle benötigten Rechenoperationen sind bereits in Maxima als Funktionen enthalten, so dass eine Realisierung der RSA-Verschlüsselung in Maxima nicht schwerfällt:

Um die Teilerfreiheit der zu $\varphi(N)$ teilerfremden und selbstgewählten Zahl e zu überprüfen, bietet es sich an, den größten gemeinsamen Teiler (ggT, engl: gcd) von e und φ(N) bestimmen zu lassen. Wenn dieser 1 ist, dann sind die beiden Zahlen teilerfrei. In Maxima gibt es dafür die Funktion `gcd()`.

```
(%i1) gcd(18,12);
(%o1) 6
```

```
(%i2) gcd(17,11);
(%o2) 1
```

Kernstück des RSA-Algorithmus ist die Restklassenpotenzierung, die zum Ver- und Entschlüsseln angewendet wird:

$$c \equiv m^e \pmod N \qquad \text{und} \qquad m \equiv c^d \mod N$$

Dies erledigt in Maxima die Funktion `power_mod()`:

```
(%i19)   c:power_mod(m,e,N);
(c)      74
```

Da wir nicht nur Zahlen, sondern Texte verschlüsseln wollen, müssen wir zuerst dafür sorgen, dass die einzelnen Zeichen (Buchstaben, Ziffern, Satzzeichen usw.) unseres Textes in Zahlen

umgewandelt werden. Dies geschieht in der Computerwelt seit jeher mit Hilfe einer ASCII[5]-Tabelle. Diese Tabelle ist frei zugänglich – sie existiert in vielfacher Ausfertigung im Internet – und ist daher nicht für eine wirkliche Verschlüsselung geeignet. Wir benötigen aber diese Zuordnungstabelle, um aus unserem zu verschlüsselnden Text zunächst Zahlenwerte zu machen. Haben wir eine erhaltene Nachricht dekodiert, so müssen wir die erhaltenen ASCII-Werte wieder in für uns lesbare Buchstaben umwandeln.

ASCII-Zeichensatz

+	0	1	2	3	4	5	6	7	8	9
30				!	"	#	$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	\	]	^	_	`	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{	|	}	~			

Hilfreich für unsere Versuche mit der RSA-Verschlüsselung ist deshalb eine Funktion, welche aus einer übergebenen Zeichenkette eine Liste mit den ASCII-Werten der einzelnen Zeichen erstellt, die dann mit Hilfe des RSA-Algorithmus verschlüsselt werden kann. Deutsche Umlaute und andere Sonderzeichen können zum Absturz führen! Man sollte deshalb vor jedem Aufruf der nachfolgenden Funktionen seine Maxima-Umgebung speichern!

```
(%i1)   string_to_ascii(zeichenkette):=block(
        [asciilist],
        asciilist:[],
        for i:1 step 1 thru slength(zeichenkette) do
            asciilist:endcons(cint(charat(zeichenkette,i)),asciilist),
        asciilist
        );
(%o1)   string_to_ascii(zeichenkette):=block([asciilist],asciilist:[],for i
thru slength(zeichenkette) do asciilist:
endcons(cint(charat(zeichenkette,i)),asciilist),asciilist)
```

```
(%i2)   phgd:string_to_ascii("PH Schwaebisch Gmuend");
(phgd)  [80,72,32,83,99,104,119,97,101,98,105,115,99,104,32,71,109,117,
101,110,100]
```

Umgekehrt ist es ebenso nützlich, eine Funktion zu Hand zu haben, welche aus den decodierten ASCII-Werten der Nachricht diese wieder als Text ausgibt:

[5] ASCII: American Standard Code for Information Interchange

```
(%i10)  ascii_to_string(asciilist):=block(
        [zeichenkette],
        zeichenkette:"",
        for i in asciilist do
            zeichenkette:concat(zeichenkette,ascii(i)),
        zeichenkette
        );
(%o10)  ascii_to_string(asciilist):=block([zeichenkette],zeichenkette:,for i
 in asciilist do zeichenkette:concat(zeichenkette,ascii(i)),zeichenkette)
```

```
(%i11)  phgd;
(%o11)  [80,72,32,83,99,104,119,97,101,98,105,115,99,104,32,71,109,117,
101,110,100]
```

```
(%i12)  ascii_to_string(phgd);
(%o12)  PH Schwaebisch Gmuend
```

3.4.6 Konkrete Durchführung in Maxima

Um verschlüsselte Nachrichten auszutauschen, müssen Sie zunächst Ihren öffentlichen und Ihren privaten Schlüssel erzeugen.

Den öffentlichen Schlüssel geben Sie Ihrem Kommunikationspartner, welcher damit eine Botschaft an Sie verschlüsselt und Ihnen die verschlüsselte Nachricht übergibt.

Sie verwenden Ihren privaten Schlüssel, um die erhaltene Nachricht zu dekodieren.

Im Folgenden sind die nötigen Arbeitsschritte im Detail beschrieben:

3.4.6.1 Vorarbeiten

Die Funktionen zum Übersetzen von Zeichen in den ASCII-Code und zurück sind in der Maxima-Datei `rsa_stringtoascii.wxm` enthalten, die Sie aus StudIP herunterladen sollen und die dann als Basis für Ihre weiteren Versuche dient. Lassen Sie nach dem Laden zuerst die enthaltenen und oben dargestellten vier Befehle ausführen!

3.4.6.2 Schlüssel generieren

Generieren Sie Ihren öffentlichen Schlüssel, indem Sie für *p* und *q* zwei Primzahlen wählen:

```
p:11;
q:17;
```

Sie bestimmen daraus durch Multiplikation das RSA-Modul *N*:

```
N:p*q;
```

und weiter die Euler'sche Phi-Funktion von *N*:

```
phi_N:(p-1)*(q-1);
```

Jetzt brauchen Sie noch eine Zahl *e*, welche teilerfremd zu phi(*N*) sein muss, Sie wählen willkürlich:

```
e:29;
```

Durch Ermitteln des ggT können Sie feststellen, ob die beiden Zahlen wirklich teilerfremd zueinander sind:

```
gcd(e,phi_N)
```

Ist dies der Fall, haben Sie mit *e* und *N* einen öffentlichen Schlüssel erzeugt.

Zum Entschlüsseln müssen Sie Ihren privaten Schlüssel generieren. Dazu benötigen man das zu e multiplikativ Inverse bezüglich des Moduls phi(*N*) Die Maxima-Funktion `inv_mod()` erledigt das Gewünschte:

```
d:inv_mod(e,phi_N);
```

Die beiden Zahlen *d* und *N* stellen Ihren privaten Schlüssel dar!

3.4.6.3 Funktionstest

Überprüfen Sie zunächst, ob alles wie gewünscht funktioniert:

Eine erste Nachricht sei die Zahl 7:

```
m:7;
```

Das Verschlüsseln erfolgt mit der Maxima-Funktion `power_mod()`:

```
c:power_mod(m,e,N);
```

Für mehr Flexibilität verwendet man eine eigene Funktion `krypt()`:

```
krypt(message):=power_mod(message,e,N);
```

Den Aufruf von `power_mod()` in eine eigene Funktion `krypt()` zu kapseln hat den Vorteil, dass `krypt()` nur mit einem Parameter aufgerufen werden muss, nämlich dem zu verschlüsselnden Wert. Die zum Verschlüsseln notwendigen Werte *e* und *N* liegen ja global vor. Damit kann später `krypt()` mittels `map()` mehrfach angewendet werden.

Das Entschlüsseln läuft nun genauso ab wie das Verschlüsseln, nun allerdings mit dem privaten Schlüssel *d* anstatt dem öffentlichen Schlüssel *e*:

```
m:power_mod(c,d,N);
```

Jetzt müssten Sie wieder die Zahl 7 erhalten.

Auch für das Entschlüsseln schreiben wir – aus genau denselben Gründen wie oben genannt – eine eigene Funktion:

```
dekrypt(message):=power_mod(message,d,N);
```

Durch die Anwendung der beiden Funktionen, welche einen String in die zugehörigen ASCII-Werte und zurück verwandeln und mittels der Mehrfachanwendung von `krypt()` und `dekrypt()` lassen sich schnell und einfach kleine Texte ver- und entschlüsseln.

3.4.6.4 Nachrichten verschlüsseln

Dies erproben Sie nun zusammen mit Ihrer Nachbarin bzw. Ihrem Nachbarn, die in die Rolle Ihres Kommunikationspartners schlüpfen.

Geben Sie Ihren öffentlichen Schlüssel *e* und *N* ihrem Kommunikationspartner.

Er oder sie verschlüsselt mit Hilfe diesen Schlüssels eine Nachricht. Dazu wird zunächst der erhaltene öffentliche Schlüssel in dessen Maxima bekannt gemacht:

```
e:29;
N:160;
```

Dann wird vom Partner der zu verschlüsselnd Text in eine Liste der zugehörigen ASCII-Werte umgewandelt:

```
ascii_liste:string_to_ascii(„Computeranwendungen");
```

Diese ASCII-Liste wird anschließend durch eine Mehrfachanwendung der Funktion `krypt()` verschlüsselt. Um eine Funktion mehrfach mit den Werten einer Liste als Argumente aufzurufen gibt es die Funktion `map()`:

```
code:map(krypt,ascii_liste);
```

Es wird eine Liste mit den codierten ASCII_Werten erzeugt.

Diese verschlüsselte Nachricht (Liste von Zahlenwerten) kann aus Maxima heraus abgespeichert werden. Hierfür ist es sinnvoll, zunächst den Speicherort und den Dateinamen als Pfad festzulegen:

```
pfad:"C:/Users/xxxx/desktop/test.txt";
```

Die obige Festlegung geht davon aus, dass die Datei auf dem Desktop des angemeldeten Users *xxxx* unter dem Namen `test.txt` abgelegt wird. Wie Ihr Username auf dem verwendeten Computer lautet, kann nötigenfalls im Dateiexplorer überprüft werden: Im Verzeichnisbaum in der Festplatte *C:* im Verzeichnis *Windows* und dort im Unterverzeichnis *Benutzer* ist dieser Name aufgeführt. An einem Rechner der Hochschule wird das jeweilige Homedirectory Z: verwendet.

```
pfad:"Z:/test.txt";
```
Auf einem Mac lautet der Pfad:
```
pfad: "/users/xxxx/desktop/test.txt"
```
`xxxx` muss auch hier durch den Benutzernamen ersetzt werden. Grundsätzlich gilt, dass Sonderzeichen im Benutzernamen zu Unverträglichkeiten führen können!

Das Speichern selbst geschieht mit dem Befehl
```
write_data(code,pfad);
```
wobei *code* die Liste mit den verschlüsselten Werten enthalten muss.

Jetzt wurde entweder auf dem Desktop oder im Z-Laufwerk eine Datei `test.txt` angelegt, die Sie Ihrem Kommunikationspartner per Mail oder auf einem Stick übergeben.

3.4.6.5 Entschlüsseln

Speichern Sie eine erhaltene Datei auf Ihrem Desktop oder in Ihrem Z-Laufwerk.

Gelesen werden kann die erhaltene Datei – nach der Festlegung des Dateipfads (s.o.) – mit dem Befehl
```
code_get:read_list(pfad);
```
Die Variable *code_get* enthält danach die Liste mit den codierten Werten, welche Sie mit Hilfe Ihres privaten Schlüssels dekodieren können.

Für das Entschlüsseln nutzen Sie die Mehrfachanwendung der Funktion `dekrypt()` mittels `map()` wieder entschlüsseln:
```
ascii_liste2:map(dekrypt,code_get);
```
Aus der erhaltenen ASCII-Liste erstellt die Funktion `ascii_to_string()` schließlich die erhaltene Nachricht in lesbarer Form:
```
ascii_to_string(ascii_liste2);
```
Tauschen Sie die Rollen!

3.5 Die Methode der kleinsten Quadrate

3.5.1 Minimierung von Fehlern

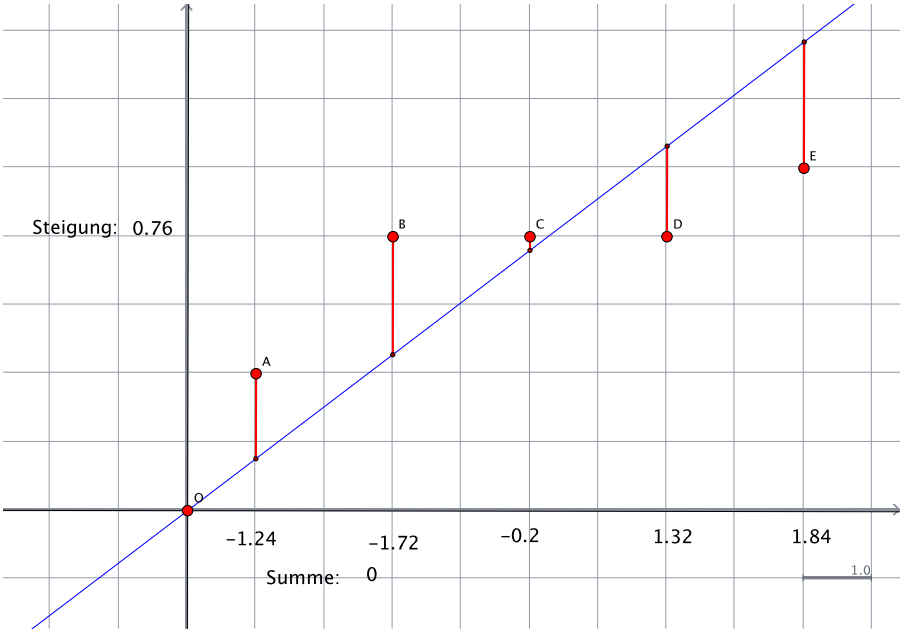
Anfangs des Jahres 1801 wurde der Zwergplanet Ceres von Guiseppe Piazzi entdeckt. Bevor dieser hinter der Sonne verschwand, konnten einige Bahndaten notiert werden. Carl Friedrich Gauß war damals bereits in der Lage, aus drei Beobachtungen die elliptische Bahn des neu entdeckten Himmelskörpers zu berechnen. Da deutlich mehr Bahnpunkte vorlagen – die ja alle einen gewissen Messfehler enthielten – wandte er die von ihm schon 1795 entwickelte Methode der kleinsten Quadrate an, um die Genauigkeit der Messungen zu erhöhen. Tatsächlich wurde Ceres im Dezember desselben Jahres an genau der Position wiederentdeckt, die Gauß vorhergesagt hatte.

Das Verfahren fußt auf einer Idee von Pierre Simon Laplace, aus vielen (fehlerbehafteten) Messwerten den „tatsächlichen“ Wert so festzulegen, dass sich die vorzeichenbehafteten Abstände der Messwerte zum tatsächlichen Wert zu Null aufaddieren.

Die grundsätzliche Idee des Verfahrens besteht darin, diskret gewonnene Messwerte (Wertepaare) durch einen Graphen und damit eine mathematische Funktion bestmöglich anzunähern. Für eine erste Annäherung gehen wir davon aus, dass in einem physikalischen Versuch die folgenden Messwertepaare gemessen wurden:

x_i:	1	3	5	7	9
y_i:	2	4	4	4	5

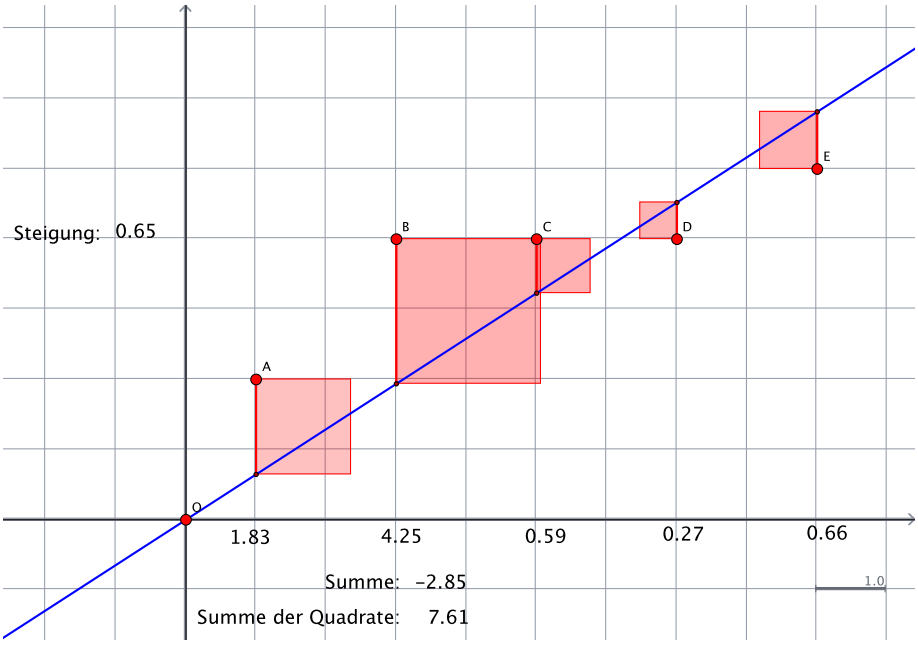
Wenn man zudem weiß, dass die zugrundeliegende Funktion eine Proportionalität ist, dann geht es darum, die Ursprungsgerade gerade so einzupassen, dass diese die gefundenen Messwerte bestmöglich repräsentiert:



Diese „bestmögliche Anpassung“ sollte nach der Idee von Laplace so geschehen, dass man von den gemessenen Punkten das Lot auf die *x*-Achse fällt und die Länge der Strecke vom Messpunkt bis zum Schnittpunkt des Lots mit dem Graphen als gerichtete Strecke misst. Die Entfernung des Punktes *A* vom Graphen beträgt in der obigen Darstellung –1,24, der Punkt *D* hingegen hat die Entfernung 1,3. In einem dynamischen Geometriesystem kann man nun die Ursprungsgerade so um den Ursprung drehen, dass die Summe der fünf Entfernungen genau Null ergibt. Laut dem obigen Bild ist dies für eine Geradensteigung von 0,76 der Fall.

Gauß veränderte diese Laplace’sche Idee dahingehend, dass er nicht mit den vorzeichenbehafteten Entfernungen rechnete, sondern mit den (immer positiven) Quadraten dieser Entfernun-

gen. Zum einen eliminiert man damit ein fehlerträchtiges Hantieren mit Vorzeichen ohne in die Bedrängnis des Rechnens mit Beträgen zu kommen, zum anderen wollte Gauß erreichen, dass größere Ablagen etwas stärker gewichtet werden. Die „bestmögliche“ Annäherung des Graphen an die Messpunkte ergibt sich nun genau dann, wenn die Summe der Abstandsquadrate minimal ist:



Für das obige Bild wurde diese optimale Lage wieder durch Probieren ermittelt, die hierbei erreichte Quadratsumme von 7,61 ist minimal. Es wird deutlich, dass dies bei einer etwas flacheren Geradensteigung von 0,65 der Fall ist. Rechnet man diesen Fall mit den vorzeichenbehafteten Entfernungen, so würde man nun ein Ergebnis von –2,85 erhalten.

3.5.2 Realisation in Cinderella

Setzen Sie die 5 Punkte *A* bis *E* anhand der gegebenen Koordinaten in ein Cinderella-Arbeitsblatt. Schalten Sie hierfür die Gitterdarstellung und die Koordinatenachsen ein.

Erstellen Sie eine um den Ursprung drehbare Gerade.

Legen Sie nun durch jeden Punkt eine Gerade senkrecht zur *x*-Achse und konstruieren Sie die Schnittpunkte jeder dieser Geraden mit der Ursprungsgeraden.

Messen Sie jeweils die Entfernungen dieser Schnittpunkte von den Punkten *A* bis *E*.

Berechnen Sie die Summe der Quadrate dieser Entfernungen.

Berechnen Sie außerdem die Steigung der Ursprungsgeraden. Sie können hierfür den Schnittpunkt der Senkrechten durch *A* mit der Ursprungsgeraden verwenden.

3.5.3 Proportionalität

Das probierende Lösen in einem dynamischen Geometriesystem ist ein guter Ansatz, um ein Verständnis für das Funktionieren der Methode aufzubauen, für den täglichen Gebrauch benötigen wir jedoch ein effizienteres Verfahren. Für die Etablierung eines solchen Verfahrens formalisieren wir zunächst die seither durchgeführten Schritte und Überlegungen:

Es geht grundsätzlich darum, gefundene diskrete Messwerte (x_i, y_i) durch eine stetige Funktion f anzunähern.

Diese Annäherung sei optimal, wenn die Summe der Entfernungen vom Messpunkt zum Graphen der Funktion minimal ist. Für die Ermittlung der Entfernung benötigen wir den gemessenen y-Wert, also y_i, und den y-Wert, den unsere Funktion f an der Stelle x_i liefern würde, also $f(x_i)$. Die Differenz beider Werte wird quadriert:

$(y_i - f(x_i))^2$

Dies machen wir für alle vorhandenen *n* Messwerte, summieren diese Entfernungsquadrate auf und erhalten so die Entfernungssumme *E*, die ein Maß für den Fehler (error) ist zwischen den praktisch gemessenen Werten und einer theoretisch konstruierten Funktion:

$$E = \sum_{i=1}^{n} \left(y_i - f\left(x_i\right) \right)^2$$

Für unser obiges Beispiel haben wir festgelegt, dass sich die Messung bzw. der zugrundeliegende Sachverhalt durch eine Proportionalität beschreiben lässt. Diese einfache Funktion hat die Form

$f(x) = m \cdot x$

Wie gut der Graph bzw. die Funktion „passt“, hängt somit einzig und allein von der Steigung m ab. Genau dies haben wir auch beim Probieren in Cinderella durchgeführt. Anders ausgedrückt: Die Summe der Fehlerquadrate und damit der Fehler *E* hängen von der Wahl von *m* ab, *E* ist daher eine Funktion von *m*. Wir schreiben:

$$E(m) = \sum_{i=1}^{n} \left(y_i - m \cdot x_i \right)^2$$

In dieser Notation haben wir außerdem die oben noch allgemein formulierte Funktion $f(x_i)$ durch die konkret zugrunde gelegte Proportionalität $m \cdot x_i$ ersetzt.

Unsere Gerade „passt“ am besten, wenn *E* minimal wird, $E(m)$ somit ein Minimum hat. Um das Minimum einer Funktion festzustellen, benötigen wir deren erste Ableitung, die wir gleich Null setzen. $E(m)$ nach *m* abzuleiten ist nicht schwierig, wir müssen dafür die Kettenregel anwenden:

$$E'(m) = \sum_{i=1}^{n} -2 \cdot \left(y_i - m \cdot x_i \right) \cdot \left(x_i \right)$$

Das Summenzeichen können wir beim Ableiten getrost ignorieren, denn die Ableitung einer Summe ist die Summe der Ableitungen. Diese Ableitung muss – um ein Minimum zu erhalten – gleich Null gesetzt werden:

$$\sum_{i=1}^{n} -2 \cdot \left(y_i - m \cdot x_i \right) \cdot \left(x_i \right) = 0$$

Den konstanten Faktor –2 im Summenterm können wir vor das Summenzeichen schreiben

$$-2 \cdot \sum_{i=1}^{n} \left(y_i - m \cdot x_i \right) \cdot \left(x_i \right) = 0$$

... und die Gleichung sofort durch –2 dividieren:

$$\sum_{i=1}^{n} \left(y_i - m \cdot x_i \right) \cdot \left(x_i \right) = 0$$

Jetzt wird der Summenterm ausmultipliziert ...

$$\sum_{i=1}^{n} \left(y_i \cdot x_i - m \cdot x_i^2 \right) = 0$$

... und die Summe aufgespaltet:

$$\sum_{i=1}^{n} y_i \cdot x_i - \sum_{i=1}^{n} m \cdot x_i^2 = 0$$

Die Gleichung wird neu arrangiert:

$$\sum_{i=1}^{n} m \cdot x_i^{\,2} = \sum_{i=1}^{n} y_i \cdot x_i$$

Auf der linken Seite wird der konstante Faktor *m* vor die Summe gezogen ...

$$m \cdot \sum_{i=1}^{n} x_i^{\,2} = \sum_{i=1}^{n} y_i \cdot x_i$$

... und wir erhalten schließlich:

$$m = \frac{\sum_{i=1}^{n} y_i \cdot x_i}{\sum_{i=1}^{n} x_i^{\,2}}$$

Damit können wir die gesuchte Steigung des Graphen aus unseren Messwerten bestimmen. Wir benötigen dazu die Summe aus allen Produkten $x_i y_i$ (Zähler des Bruchs) sowie die Summe der Quadrate aller Messwerte x_i (Nenner des Bruchs).

Erstellen Sie die nachfolgende Tabelle in EXCEL, wobei nur die fünf Messwerte x_i und y_i als Zahlenwerte eingegeben werden. Die jeweiligen Produkte $x_i y_i$ und die Quadrate der Abstände lassen Sie von EXCEL jeweils berechnen, ebenso die Summen der Produkte bzw. Quadrate! Schließlich lassen Sie den Quotienten aus Produktsumme geteilt durch Quadratsumme berechnen.

x_i	y_i	$x_i y_i$	x_i^2
1	2	2	1
3	4	12	9
5	4	20	25
7	4	28	49
9	5	45	81
Summen:		107	165

Wir erhalten mit diesen Werten $m = \frac{107}{165} = 0{,}6\overline{48}$ und damit eine hervorragende Bestätigung unseres oben rein experimentell gefundenen Ergebnisses.

Verändern Sie die Messwerte in Ihrer Tabelle und in Cinderella und überprüfen Sie beide Systeme auf Übereinstimmung.

3.5.4 Linearität

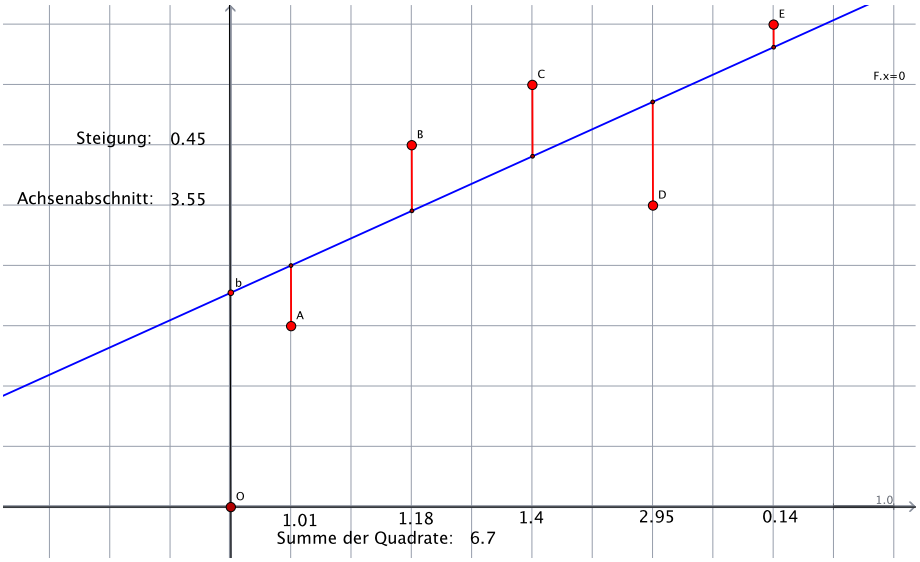
Die Anpassung des Graphen einer Proportionalität an gefundene Messwerte ist experimentell und rechnerisch relativ einfach zu bewerkstelligen und man erhält damit einen guten Einblick in die Methode der kleinsten Quadrate. Im angelsächsischen Sprachgebrauch wird dieses Verfahren *least squares fit* (kurz *lsf*) genannt.

Während bei einer Proportionalität nur die Steigung des Graphen ermittelt werden muss, benötigt man für eine allgemeine Linearität schon zwei Parameter, nämlich neben der Steigung *m* auch noch den Achsenabschnitt *b*. Natürlich kann man sich zunächst wieder mit Hilfe eines dynamischen Geometriesystems auf die Suche machen und wird recht schnell feststellen, dass das Verändern von nun zwei Parametern die Suche nach der kleinsten Summe der Fehlerquadrate recht aufwändig macht. Mit etwas Geduld wird man aber nach einiger Zeit auch zu einem Ergebnis kommen und die nach der Methode der kleinsten Quadrate am besten passende Gerade finden. Wir verwenden für diese Aufgabe eine andere Messwertreihe:

x_i:	1	3	5	7	9
y_i:	3	6	7	5	8

Erstellen Sie eine zweite Cinderella-Datei ähnlich der vorhergehenden. Der einzige Unterschied besteht darin, dass der Schnittpunkt des Graphen mit der *y*-Achse nun auf der *y*-Achse verschiebbar sein soll.

Das Probieren in Cinderella liefert mit etwas Geduld den kleinsten Wert für die Summe der Quadrate bei der Steigung 0,45 und dem Achsenabschnitt 3,55:



C14

	A	B	C	D	E
1		Least Square Fit			
2					
3		xi	yi	f(xi)	(f(xi)–yi)^2
4		1	3	4,50	2,25
5		3	6	5,50	0,25
6		5	7	6,50	0,25
7		7	5	7,50	6,25
8		9	8	8,50	0,25
9				Summe der Quadrate:	**9,25**
10					
11	m:	0,50			
12	b:	4,00			

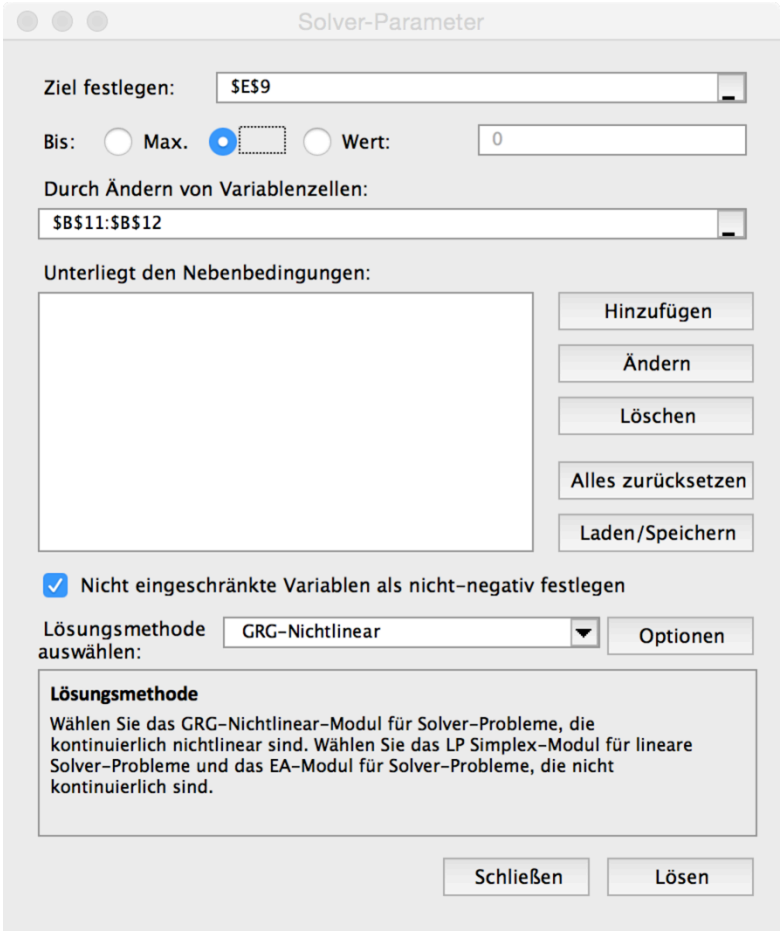
Auch diesmal dient diese experimentell-empirische Lösungssuche mit Cinderella in erster Linie dem eigenen Verständnis und weniger dem Auffinden der konkreten Lösung. Für Letzteres nehmen wir besser die Rechnerhilfe in Anspruch und bedienen uns der Tabellenkalkulation EXCEL, mit der wir diese experimentelle Suche nach den beiden Funktionsparametern *m* und *b* direkt durchführen können:

In ein Tabellenblatt schreiben wir in den Spalten *B* und *C* in die Zeilen 4 bis 8 die Koordinaten x_i und y_i unserer gefundenen Messwerte.

In der Spalte *D* lassen wir jeweils rechts daneben die aufgrund der linearen Funktion $y = mx + b$ berechneten Funktionswerte berechnen. Die Funktionsparameter *m* und *b* haben wir (zunächst grob geschätzt) in die Felder `B11` und `B12` eingetragen.

In der Spalte `E` werden dann die Quadrate der Differenzen $y_i - f(x_i)$ eingetragen. Die Summe dieser Quadrate 9,25 wird in `E9` berechnet und dargestellt. Während man im Geometrieprogramm die Lage der Geraden direkt geometrisch beeinflusst, kann man in EXCEL die Gerade numerisch über Verändern der Werte für *m* und *b* manipulieren. Dabei behält man die Summe der Quadrate in `E9` im Auge und trachtet so danach, dort einen möglichst kleinen Wert zu erzielen – was schließlich nicht minder mühsam ist.

EXCEL enthält allerdings den *Solver*, der genau diese mühevolle Tätigkeit für uns übernehmen kann! Den *Solver* haben wir bereits im Abschnitt 1.3 bei den Beispielen zur Optimierung verwendet. Voraussetzung für seine Nutzung ist jedoch, dass im Menü EXTRAS /ADD INS die Option `Solver.xlam` angekreuzt wurde; erst dann wird der Menüpunkt SOLVER im Menü EXTRAS aufgeführt.



Bei dessen Auswahl erscheint ein Fenster, welches die benötigten Angaben erfragt:

In „Ziel festlegen“ trägt man den Bezug derjenigen Zelle ein, welche den zu manipulierenden Ergebniswert enthält. In unserem Fall ist dies die Zelle `E9` mit der Summe der Quadrate.

In der nächsten Zeile legt man fest, ob der Wert maximal oder minimal werden oder aber einen bestimmten Wert erreichen soll. Da wir den Minimalwert erreichen wollen, markieren wir den zweiten Button.

Die nächste Eingabe betrifft diejenigen Variablenzellen, die verändert werden sollen, um das Gewünschte zu erreichen, dies sind die beiden Zellen `B11` und `B12`.

Schließlich klickt man auf die Schaltfläche „Lösen“ und kurz darauf präsentiert EXCEL die gefundene Lösung.

Neben dem Meldungsfenster werden die gefundenen Werte sofort in das Tabellenblatt an die jeweiligen Stellen eingetragen.

EXCEL hat für die Steigung m den Wert 0,45 und für den Achsenabschnitt b den Wert 3,55 gefunden. Die Summe der Quadrate hat dabei ihr Minimum bei 6,7.

Damit haben wir wohl ein Ergebnis erreicht, allerdings hat auch EXCEL dieses Ergebnis nur durch Probieren gefunden. Wir wenden uns deshalb schließlich und endlich noch der konkreten Berechnung zu!

Der Vorgang ist kaum anders als oben bereits im Zusammenhang mit der Proportionalität erläutert. Der einzige wesentliche Unterschied ist jetzt, dass wir hinter den gefundenen Messwerten bzw. der damit beschriebenen konkreten Situation einen allgemein linearen Zusammenhang vermuten. Nach wie vor gilt, dass unser Fehler die Summe der Abweichungen ist:

$$E = \sum_{i=1}^{n} \left(y_i - f\left(x_i\right) \right)^2$$

Jetzt ist unsere zugrundeliegende Funktion allerdings eine allgemeine Linearität

$$y = m \cdot x + b$$

Damit hängt E jetzt von zwei Parametern m und b ab. Es gilt somit:

$$E(m,b)=\sum_{i=1}^{n}\left(y_i-\left(m\cdot x_i+b\right)\right)^2$$

Das mathematisch Interessante ist nun, dass wir damit eine Funktion zweier Veränderlicher vorliegen haben: Die Lage der Geraden und damit die Summe der Quadrate hängt von deren Steigung *m* und dem Achsenabschnitt *b* ab. Nach wie vor gilt jedoch, dass wir das Minimum dieser Funktion suchen und dafür benötigen wir die ersten partiellen Ableitungen, die wieder gleich Null gesetzt werden. Funktionen mit mehreren Veränderlichen werden partiell abgeleitet, indem man wie gewohnt jeweils nach einer Variablen ableitet und die andere(n) Variable(n) als Konstante behandelt. Wir leiten *E*(*m*, *b*) zum Einen nach *m* und zum Anderen nach *b* ab, dabei wird wieder die Kettenregel angewandt:

$$E'_m(m,b)=\sum_{i=1}^{n}2\left(y_i-\left(m\cdot x_i+b\right)\right)\cdot -x_i$$

$$E'_b(m,b)=\sum_{i=1}^{n}2\left(y_i-\left(m\cdot x_i+b\right)\right)\cdot -1$$

Beide partiellen Ableitungen müssen für das Vorliegen eines Minimums gleich Null werden:

$$\sum_{i=1}^{n}2\left(y_i-\left(m\cdot x_i+b\right)\right)\cdot -x_i=0$$

$$\sum_{i=1}^{n}2\left(y_i-\left(m\cdot x_i+b\right)\right)\cdot -1=0$$

Wieder ziehen wir den konstanten Faktor 2 bzw. –2 vor das Summenzeichen und dividieren beide Gleichungen sofort durch 2 bzw. –2:

$$\sum_{i=1}^{n}\left(y_i-\left(m\cdot x_i+b\right)\right)\cdot -x_i=0$$

$$\sum_{i=1}^{n}\left(y_i-\left(m\cdot x_i+b\right)\right)\cdot -1=0$$

Wir multiplizieren die Summenterme aus:

$$\sum_{i=1}^{n}\left(-x_iy_i+m\cdot x_i^2+bx_i\right)=0$$

$$\sum_{i=1}^{n}\left(-y_i+m\cdot x_i+b\right)=0$$

Wieder werden die einzelnen Summanden aufgespaltet ...

$$-\sum_{i=1}^{n}x_iy_i+\sum_{i=1}^{n}m\cdot x_i^2+\sum_{i=1}^{n}bx_i=0$$

$$-\sum_{i=1}^{n}y_i+\sum_{i=1}^{n}m\cdot x_i+\sum_{i=1}^{n}b=0$$

... und die Gleichungen umgestellt:

$$\sum_{i=1}^{n}m\cdot x_i^2+\sum_{i=1}^{n}bx_i=\sum_{i=1}^{n}x_iy_i$$

$$\sum_{i=1}^{n}m\cdot x_i+\sum_{i=1}^{n}b=\sum_{i=1}^{n}y_i$$

Die Konstanten werden vor die Summenzeichen gezogen:

$$m \cdot \sum_{i=1}^{n} x_i^2 + b \cdot \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i y_i$$

$$m \cdot \sum_{i=1}^{n} x_i + \sum_{i=1}^{n} b = \sum_{i=1}^{n} y_i$$

Schließlich können aus den Messwerten die benötigten Summen ermittelt ...

	x_i	y_i	x_i^2	$x_i y_i$
	1	3	1	3
	3	6	9	18
	5	7	25	35
	7	5	49	35
	9	8	81	72
Summen:	25	29	165	163

... und in die Gleichungen eingesetzt werden:

$$m \cdot 165 + b \cdot 25 = 163$$

$$m \cdot 25 + b \cdot 5 = 29$$

Dieses lineare Gleichungssystem wird gelöst[6] und ergibt

$m = 0{,}45$ und $b = 3{,}55$,

was wieder unser mit Hilfe von Cinderella experimentell gefundenes Ergebnis bestätigt.

3.5.5 Bestimmung mit Maxima

Neben linearen Zusammenhängen kann es für gefundene Messewerte viele weitere Zusammenhänge geben: Es kann eine quadratische Funktion oder allgemein eine Potenzfunktion höheren Grades für eine experimentell überprüfte Situation modelliert werden oder eine Exponentialfunktion oder eine trigonometrische Funktion oder derlei mehr. Das Prinzip ist jedoch immer dasselbe: Der Graph der entsprechenden Funktion soll so an die gefundenen Messwerte angepasst werden, dass die Summe der Entfernungsquadrate minimal wird. Jede Funktionsklasse wird durch entsprechende Parameter eindeutig festgelegt, so beispielsweise eine allgemeine quadratische Funktion ax^2+bx+c durch die Parameter a, b und c, eine Funktion 5. Grades $ax^5+bx^4+cx^3+dx^2+ex+f$ benötigt schon 5 Parameter für eine eindeutige Festlegung. Während der prinzipielle Rechenweg derselbe bleibt, steigt damit der rechnerische Aufwand. Im letztgenannten Fall müssen fünf partielle Ableitungen gebildet und schließlich ein lineares Gleichungssystem mit 5 Variablen gelöst werden. Da es uns hier nicht um den Nachweis statistischer Verfahren geht, übergeben wir die konkrete Durchführung der Methode der kleinsten Quadrate getrost dem Computer und lassen Maxima den rechnerischen Aufwand für uns durchführen.

Maxima benötigt dafür eine Matrix mit den gefundenen Messwerten und die konkrete Funktion, welche wir hinter dem Zusammenhang vermuten. Zur Klarstellung muss Maxima auch noch wissen, welches die Funktionsvariablen sind (in der Regel x und y) und welche Bezeichnungen die Funktionsparameter haben.

Damit jedoch Maxima mit der Methode der kleinsten Quadrate überhaupt arbeiten kann, muss zunächst das entsprechende Paket mit

```
load(lsquares);
```

[6] EXCEL kann natürlich lineare Gleichungssysteme lösen. Wie dies geht, soll jedoch hier nicht diskutiert werden. Nötigenfalls können Sie dieses einfache System von Hand lösen!

geladen werden. Die Funktion, die uns die gewünschten Parameter der zugrundeliegenden Funktion liefert, heißt `lsquares_estimates()`, sie hat die folgende Syntax:

`lsquares_estimates(<matrix>,<fkt_var>,<fkt_gl>,<fkt_param>).`

`matrix` ist eine Matrix mit den Messwerten,

`fkt_var` ist eine Liste mit den Funktionsvariablen, in aller Regel [x,y],

`fkt_gl` ist die passende Funktionsgleichung, beispielsweise y=m*x+b und

`fkt_param` ist eine Liste mit den Funktionsparametern, beispielsweise [m,b].

Um unser erstes diskutiertes Beispiel von Maxima lösen zu lassen, erstellt man zuerst die Matrix mit den Messwerten:

```
M1:matrix([1,2],[3,4],[5,4],[7,4],[9,5]);
```

dann ruft man `lsquares_estimates()` folgendermaßen auf:

```
lsquares_estimates(M1,[x,y],y=m*x,[m]);
```

und erhält als Ergebnis

$$\left[m=\frac{107}{165}\right],$$

woran man unschwer erkennen kann, dass Maxima intern genauso arbeitet, wie wir in unserem ersten Beispiel.

Für die Lösung des zweiten Beispiels definieren wir wieder zuerst die Matrix mit den Messwerten:

```
M2:matrix([1,3],[3,6],[5,7],[7,5],[9,8]);
```

und rufen auf:

```
lsquares_estimates(M2,[x,y],y=m*x+b,[m,b]);
```

Als Ergebnis erhalten wir

$$[[m=\frac{9}{20},b=\frac{71}{20}]],$$

was wiederum mit unserem oben gefundenen Ergebnis übereinstimmt.

Sie haben die nachfolgenden zehn Messwerte aufgenommen und wissen, dass es sich dabei um eine allgemeine quadratische Funktion handelt:

```
[0,7], [1,4.0], [2,2.5], [3,2], [4,4.5], [5,4.5], [6,6.5],
[7,12], [8,17], [9,24]
```

Bestimmen Sie in Maxima den am besten passenden Funktionsterm!