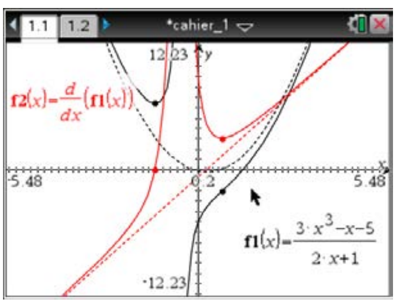
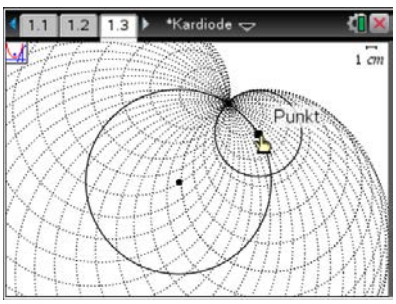
Aufgaben zur Analysis mit TI-Nspire™ CX CAS

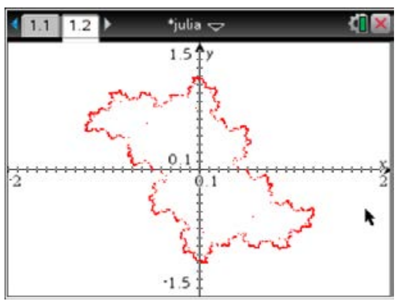
Didier Deses

bearbeitet für Version 3.6 und ergänzt

vom *ACDCA* in Kooperation mit *T³ Österreich*







Inhalt

1 Einleitung 4

2 Einfache Aufgaben 6

3 Etwas schwierigere Aufgaben 13

3.1 Standardaufgaben 13
3.2 Weitere Koordinatensysteme 25
3.3 Ganz einfach mit TI-Nspire™ CX CAS! 32

4 Und noch ein wenig schwieriger 40
4.1 Wie arbeitet TI-Nspire™ CX CAS 40
4.2 Programmieren mit TIBASIC ist einfach 44
4.3 Jetzt ohne Programmieren 57

5 Einige weitere Möglichkeiten mit TI-Nspire™ CX CAS 61
5.1 Lösen von Gleichungssystemen 61
5.2 Animationen 65
5.3 Hüllkurven und Ortslinien 66
5.4 Taylorpolynome 69
5.5 3D-Gafik 70
5.6 Julia-Mengen 72
5.7 Programme – Funktionen 73

Impressum

Das vorliegende Material bezieht sich auf das T³ Cahier 33, erstellt von Didier Dedes, T³ Vlaanderen (www.t3vlaanderen.be). Es wurde bearbeitet und ergänzt für Version 3.6 des TI-Nspire™ CX CAS im Rahmen von Veranstaltungen des ACDCA (Austrian Center for Didactics of Computer Algebra) in Kooperation mit T³ Österreich (www.t3oesterreich.at). Die Koordination dieses Projekts wurde von Mag. Josef Böhm, ACDCA, übernommen.

Dieses und weiteres Material steht zum pdf-Download hier bereit: www.ti-unterrichtsmaterialien.net
Mehr Informationen zu T³ Europe: www.t3europe.eu

Vorwort

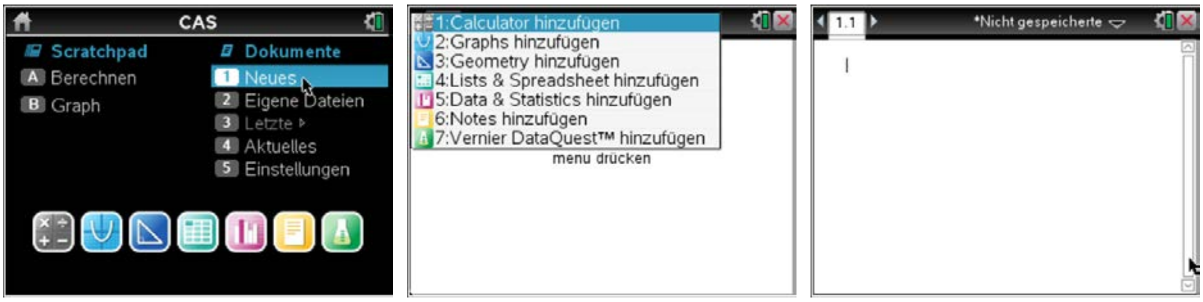
Dieses Skriptum ist im Wesentlichen eine Übersetzung eines belgischen T³-Hefts aus dem Holländischen. T³-Flandern hat eine große Reihe von „Cahiers“ herausgegeben, unter denen sich auch Cahier 14, *Voorbeelden met de TI-84+ uit de analyse*, vom selben Autor befindet. Mit dem hier vorliegenden Cahier 33 zeigt *Didier Dedes* an den selben Aufgaben den durch den Einsatz von CAS erreichbaren Mehrwert auf.

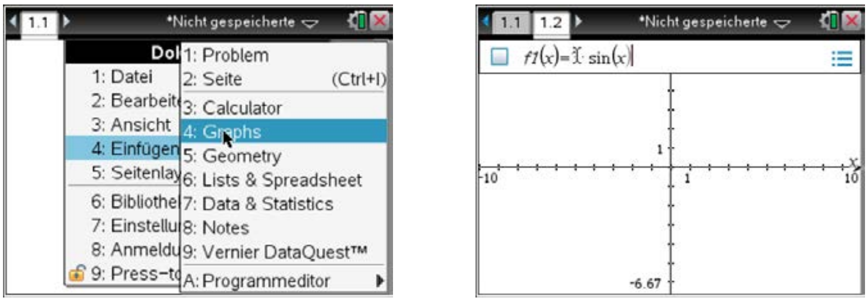
Cahier 33 wurde an die aktuellste Version von TI-Nspire™ CX CAS (Version 3.6) angepasst. Auf die – wenigen - Hinweise auf die flämischen Lehrpläne wurde verzichtet. Einige Aufgaben wurden ergänzt bzw. hinzugefügt, so dass nun anstelle von ursprünglich 47 Aufgaben insgesamt 60 angeboten werden. Der kurze letzte Abschnitt *Programm – Funktion* wurde überhaupt neu als Ergänzung zum Kapitel über das Programmieren aufgenommen. Die Julia-Mengen sind – obwohl nicht unbedingt zur Analysis gehörig – bereits im Original zu finden. Sie stellen sicherlich einen attraktiver Aufputz dar.

Josef Böhm, im Herbst 2013

1 Einleitung

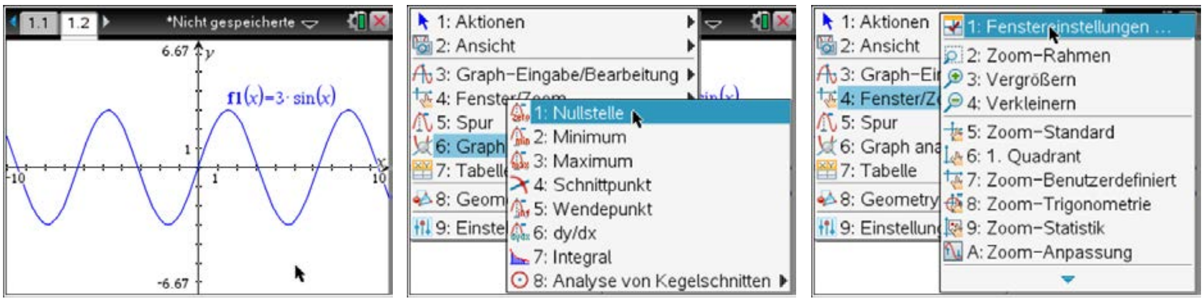
Die für die Analysis nützlichsten Funktionen werden wir in der *Calculator*- und in der *Graphs*-App (Rechen- und Grafikfenster) finden. Wir wollen ein Dokument erzeugen, das aus einem Rechen- und einem Grafikfenster besteht. Dafür drücke [⌂on] und eröffne eine neues Dokument. Füge zuerst den 1:Calculator hinzu, gehe mit [doc▾] nochmals zu 4:Einfügen und öffne auch eine 2:Graphs-Seite.



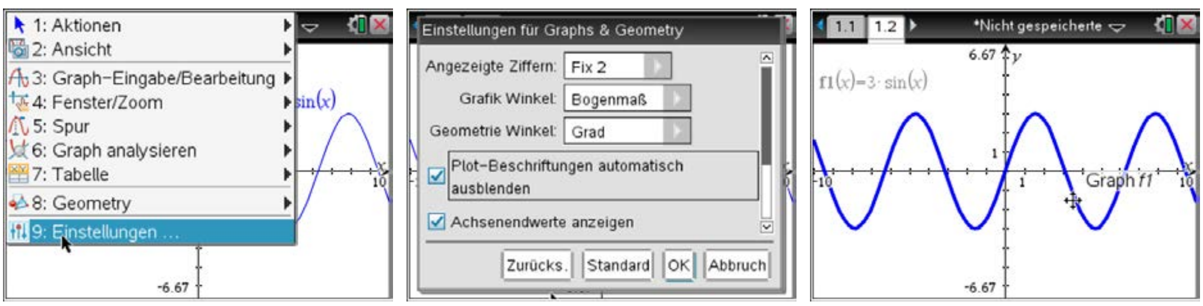


Nachdem wir eine Grafik im *Graphs*-Fenster erzeugt haben, lassen sich die gebräuchlichsten Funktionen über ein Menü aufrufen. Hier wurde für *f1*(*x*) die Funktion 3*sin(*x*) gewählt.

Wir drücken [menu] > 6:Graph Analysieren. Jetzt kann der Graph untersucht werden. Das Grafikfenster wird angepasst über [menu] > 4:Fenster/Zoom Die Option 5:Zoom-Standard ergibt ein im Allgemeinen passendes Koordinatensystem.



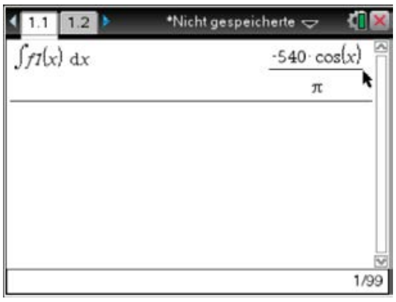
Gemeinsam mit dem Graph wird die Funktionsgleichung angegeben. Wenn mehrere Graphen gezeichnet werden, kann dadurch der Schirm überladen werden. Die Beschriftungen lassen sich aus- und wieder einschalten über [menu] > 9:Einstellungen ...



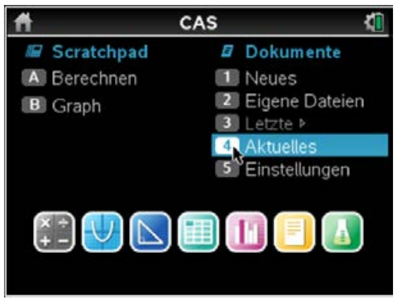
Die Beschriftung ist ausgeschaltet. Wenn der Cursor in die Nähe des Graph kommt, zeigt sie sich und der Graph wird in einer dicken Strichstärke angezeigt.

Ein nützlicher Hinweis:

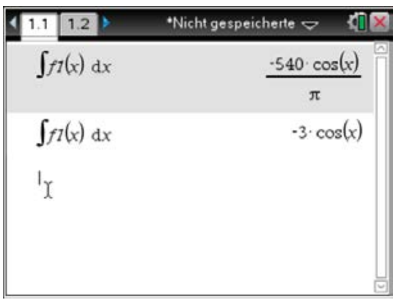
Wenn wir in dem dichten Wald von Menüs einen Befehl nicht mehr finden, oder diesen ganz einfach vergessen haben, dann verwenden wir den Katalog (📖). Hier finden wir die vollständige Liste aller verfügbaren Befehle und Funktionen mit Hinweisen zur richtigen Eingabe der notwendigen Parameter. Wenn wir den – uns eventuell noch bekannten – Anfangsbuchstaben des Befehls wissen, tippen wir diesen ein und sind gleich in der Nähe des Befehls. Wir wollen die Funktion *f1*(*x*) integrieren, daher tippen wir ein: I und finden das Integralzeichen, das wir in das Rechenfenster übernehmen. Zwischen den Fenstern wechseln wir mit ctrl ◀ bzw. ctrl ▶.



Das haben wir wohl nicht als Stammfunktion erwartet – es kann aber passieren. warum? Hier wird offensichtlich im Gradmaß gerechnet. In der Mathematik üblich ist aber das Bogenmaß. Wir müssen daher die Dokumenteinstellungen ändern. Die Abbildungen zeigen, wie das geschieht. Dann erhalten wir auch das erwartete Ergebnis für die Integration. Über on gelangen wir in den Eröffnungsschirm, wo wir die Dokumenteinstellungen ändern können.



Dann kehren wir zur aktuellen Bearbeitung zurück und integrieren ein zweites mal:



Es sei gleich hier vermerkt, dass wir das Werkzeug zum Integrieren auch über die zweite Registerkarte im Katalog gefunden hätten.

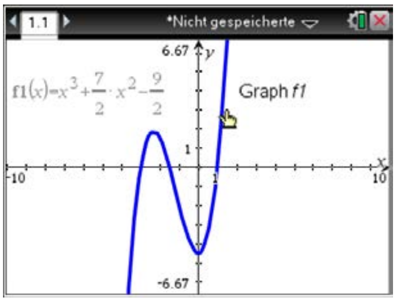
2 Einfache Aufgaben

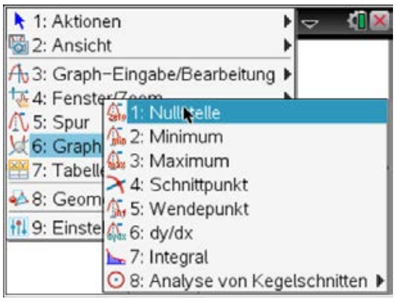
Aufgabe 1 *Nullstellen berechnen*

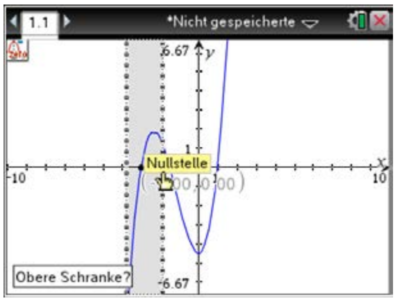
Erzeuge den Graph der reellen Funktion $f:\mathbb{R}\to\mathbb{R}$ mit der Vorschrift: $f(x)=x^3+\frac{7}{2}x^2-\frac{9}{2}$. Berechne die Nullstellen sowohl händisch als auch mit dem TI-Nspire™ CX CAS.

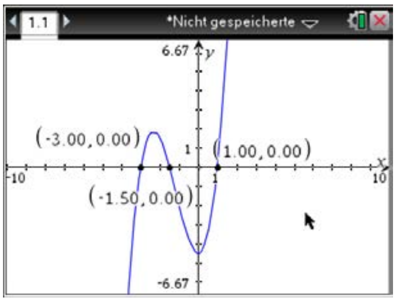
Lösung 1 Um die Nullstellen zu finden, können wir das Polynom faktorisieren. Wir erkennen, dass die Summe der Koeffizienten null ist, daher wissen wir, dass das Polynom durch $(x-1)$ teilbar ist. Nach Anwendung des Horner-Schemas bleibt eine quadratische Gleichung, die leicht gelöst werden kann. Damit können die Nullstellen 1, -3 und $\frac{3}{2}$ aufgefunden werden. Mit dem TI-Nspire™ CX CAS gehen wir folgendermaßen vor:

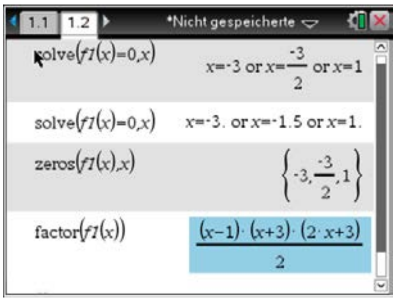
Wir geben die Funktionsgleichung ins Grafikfenster ein und lassen den Graph zeichnen. Mit ctrl + G öffnet sich immer die Eingabezeile. Dann werden über menu > 6:Graph Analysieren > 1:Nullstelle (dreimalige Anwendung) alle drei Nullstellen gefunden und angezeigt.









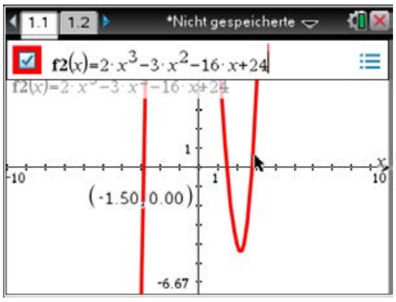


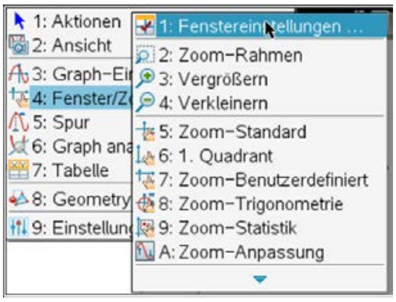
Die Nullstellen lassen sich auch im Rechenfenster auf verschiedene Arten bestimmen. Die Ausgabe als Dezimalzahl erzwingen wir, indem wir anstelle von enter mit ctrl+enter (= ≈) abschließen.

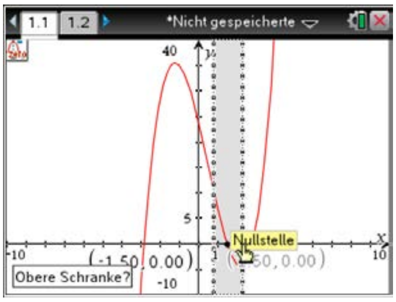
Aufgabe 2 *Zerlegen in Faktoren*

Die Zerlegung von Polynomen in Linearfaktoren (wie bereits oben gezeigt) kann in vielen Fällen eine echte Hilfe sein. Zerlege den Term $2x^3-3x^2-16x+24$ in seine Linearfaktoren. Es geht hier nicht so unmittelbar wie in Aufgabe 1, da wir keine ganzzahligen Nullstellen erkennen können.

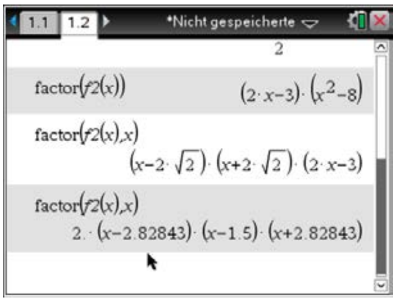
Lösung 2 Wir erzeugen zuerst den Graph und passen über menu > 4:Fenster > 1:Fenstereinstellungen die Skalierung der *y*-Achse so an, dass sich ein vernünftiges Bild ergibt. Wieder über menu > 6:Graph Analysieren > 1:Nullstelle suchen wir zumindest eine Nullstelle, die wir bei $x=\frac{3}{2}$ entdecken können. (Überprüfe das händisch!).







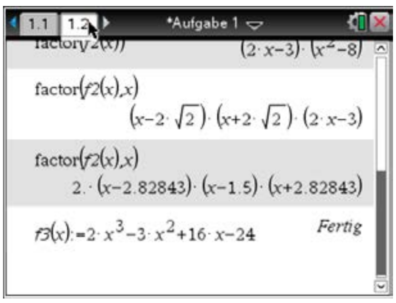
Über das Verfahren von Horner (oder durch Polynomdivision) können wir auf ein quadratisches Polynom reduzieren und die entsprechende quadratische Gleichung lösen. Die endgültige Faktorisierung lautet: $2\left(x-\frac{3}{2}\right)\left(x-2\sqrt{2}\right)\left(x+2\sqrt{2}\right)$. Mit dem CAS lassen wir das natürlich auch TI-Nspire™ CX CAS ausrechnen (sowohl exakt als auch numerisch):

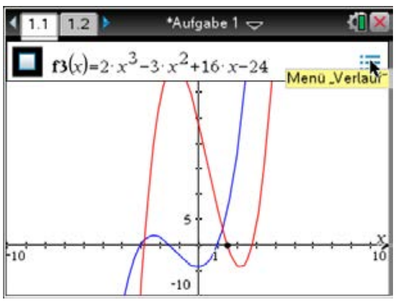


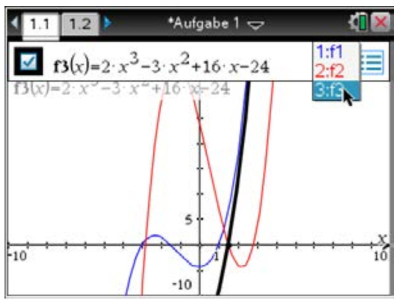
Aufgabe 3 *Zerlegen in Faktoren 2*

Zerlege den Term $2x^3-3x^2+16x+24$ in seine Linearfaktoren. Es geht hier nicht so leicht wie in Aufgabe 2.

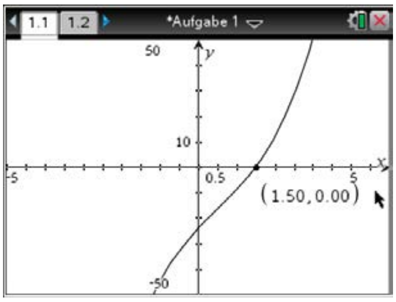
Lösung 3 Wir definieren die Funktion im Rechenfenster und stellen den Graph in einem geeigneten Fenster dar. Über menu > 6:Graph Analysieren > 1:Nullstelle suchen wir die – einzige? – Nullstelle, die wir wieder bei $x=\frac{3}{2}$ entdecken können. (Überprüfe das händisch!).

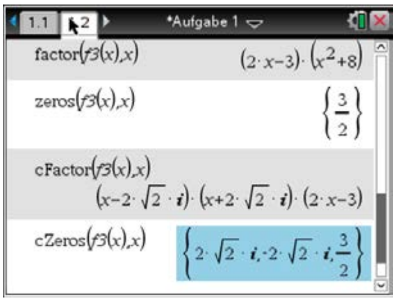






Über den „Verlauf“ rechts oben werden die aktiven Graphen angezeigt. Hier können dann die beiden vorher definierten Funktionen *f1* und *f2* deaktiviert werden. *f3* stellt sich nun so dar. Wir können wieder auf ein quadratisches Polynom reduzieren und finden zwei weitere, aber konjugiert komplex Lösungen, die wir nun auch im Rechenfenster aufspüren wollen.



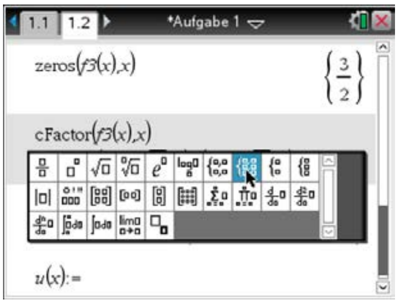


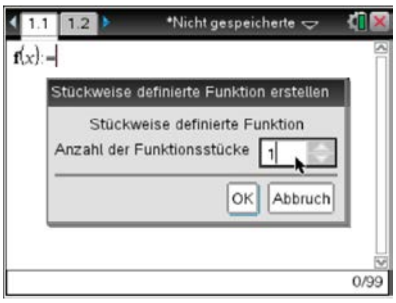
Wir sehen, dass weder factor noch zeros zum Erfolg führen. Mit cFactor und cZeros werden auch die komplexen Nullstellen und die zugehörigen Linearfaktoren angegeben.

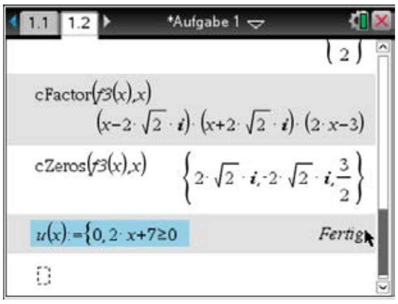
Aufgabe 4 *Ungleichungen*

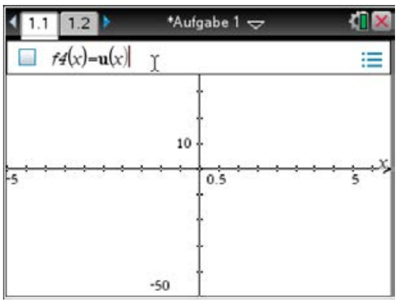
Löse grafisch die Ungleichung $2x+7 \geq 0$ in $\mathbb{R}$.

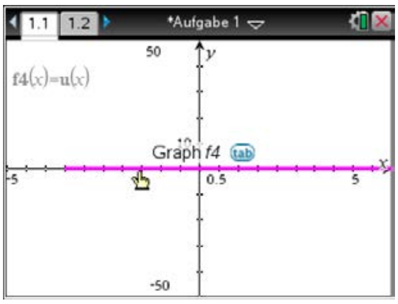
Lösung 4 Wir definieren die Funktion im Rechenfenster als eine abschnittsweise definierte Funktion (mit nur einem Abschnitt) und verwenden dazu eine vorgegebene Schablone. Das Fenster mit allen verfügbaren Schablonen rufen wir auf über . Wir füllen die Schablone aus: nur wenn $2x + 7 \geq 0$ soll der Funktionswert 0 angenommen werden. Dann stellen wir den Graph in einem geeigneten Fenster dar. Wir sehen sofort den Bereich, für den die Ungleichung erfüllt ist.

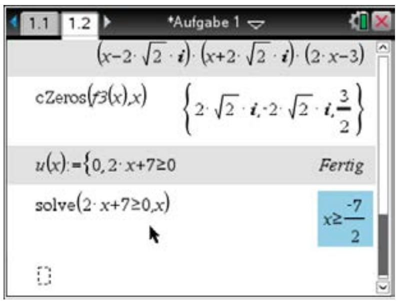












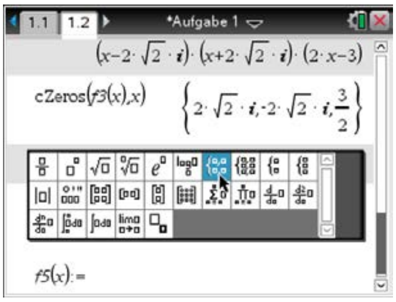
Wenn wir mit dem „Händchen“ zum Graph von f4 kommen, könnten wir ja auch die *x*-Achse meinen. Mit der tab-Taste können wir bei Mehrdeutigkeit des Pointers (hier das „Händchen“) zwischen den möglichen Objekten umschalten.

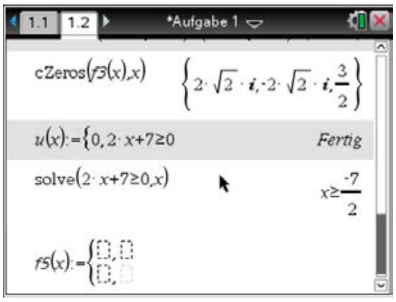
Wie wir im Rechenfenster sehen, kann TI-Nspire™ CX CAS auch hier mit der Ungleichung umgehen.

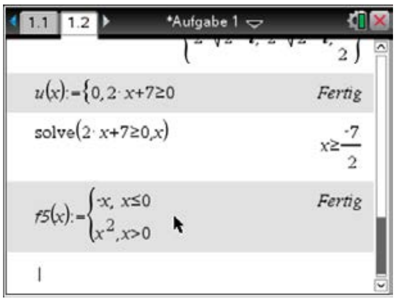
Aufgabe 5 *Abschnittsweise definierte Funktionen*

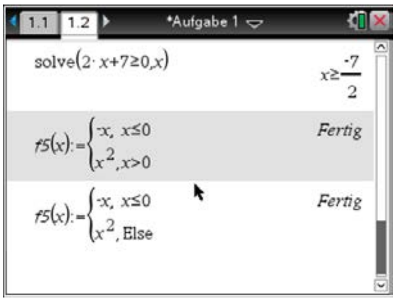
Erzeuge den Graph der abschnittsweise definierten Funktion $f(x)=\begin{cases}-x & \text{für } x\le 0\\ x^2 & \text{für } x>0\end{cases}$ und untersuche deren Stetigkeit.

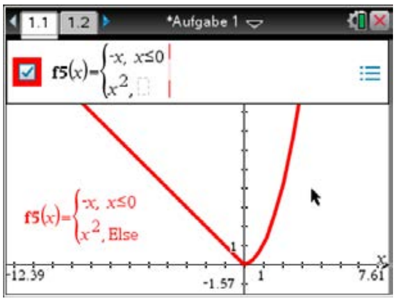
Lösung 5 Die Funktion wird wie in Aufgabe 4 über die geeignete Schablone eingegeben. Das kann auf zwei Arten erfolgen, am Ergebnis ändert sich nichts.











Am Graph erkennen wir die Stetigkeit der Funktion.

Aufgabe 6 *Grenzwerte*

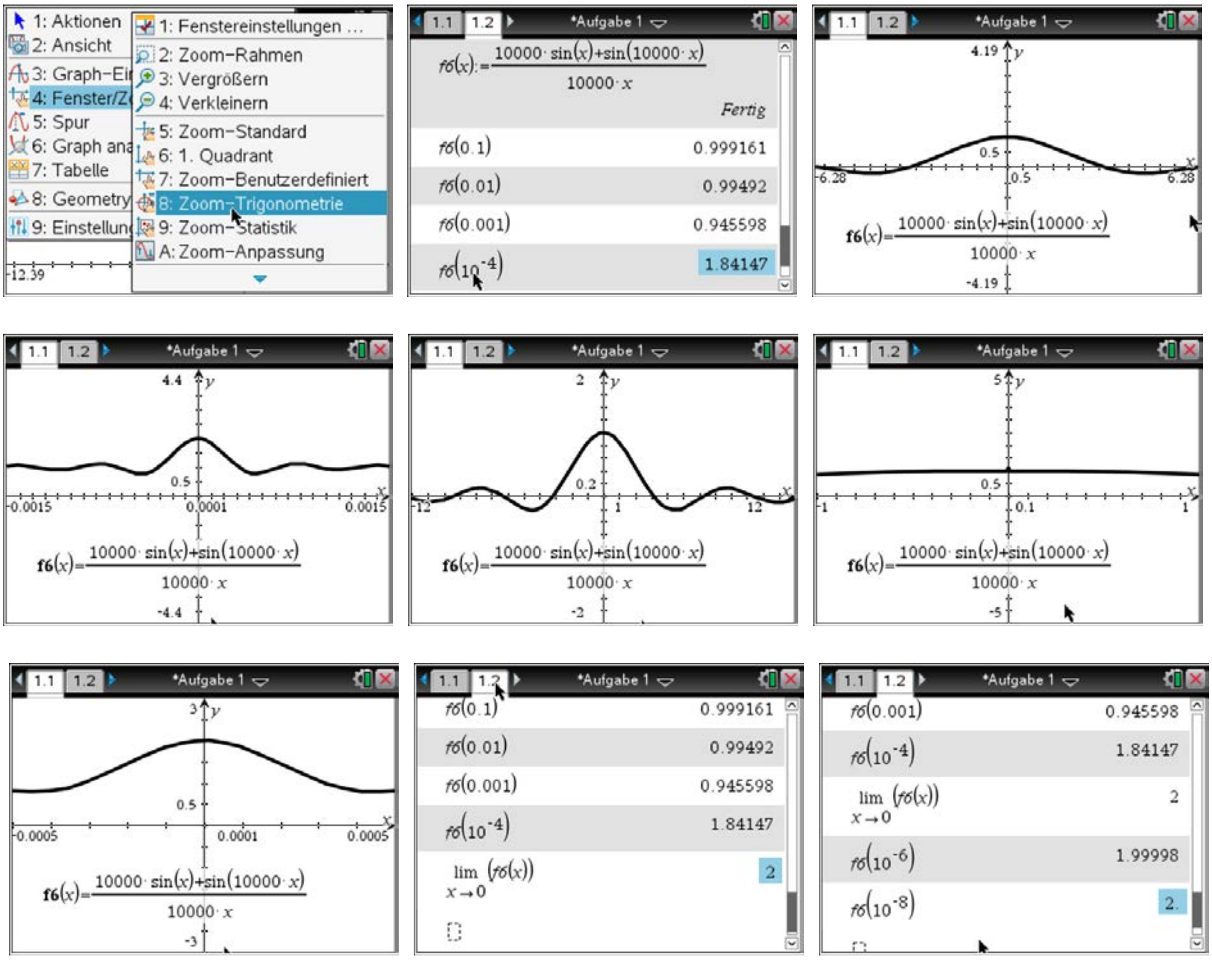
Manchmal führt der Einsatz des TI-Nspire™ CX CAS zu einem falschen Schluss. Dazu wollen wir ein klassisches Beispiel behandeln. Erzeuge den Graph der Funktion $f(x)=\dfrac{10000\sin x+\sin 10000x}{10000x}$. Für die Fenstereinstellung des Grafikfensters wähle die Option 8:Zoom-Trigonometrie. Bestimme zuerst im *Calculator* die Funktionswerte nahe bei null und vergleiche die Ergebnisse mit dem Graph. „Zoome“ dich dann immer näher zum Koordinatenursprung, etwa bis [-0,001;0.001] und beobachte die Veränderung des Graphs. Lass zum Schluss den Grenzwert berechnen.

Lösung 6 Wenn man weiß, dass $\lim\limits_{x\to 0}\dfrac{\sin x}{x}=1$, dann ist der Grenzwert einfach zu berechnen:

$$\begin{aligned}\lim_{x\to 0}\frac{10000\sin x+\sin 10000x}{10000x}&=\lim_{x\to 0}\frac{10000\sin x}{10000x}+\lim_{x\to 0}\frac{\sin 10000x}{10000x}\\&=1+1=2\end{aligned}$$

Mit TI-Nspire™ CX CAS erhalten wir vorerst einen Graph, der eine falsche Antwort vermittelt. Erst nachdem die Fensterkoordinaten angepasst wurden, zeigt sich die wahre Gestalt des Graphs und damit auch der richtige Grenzwert.

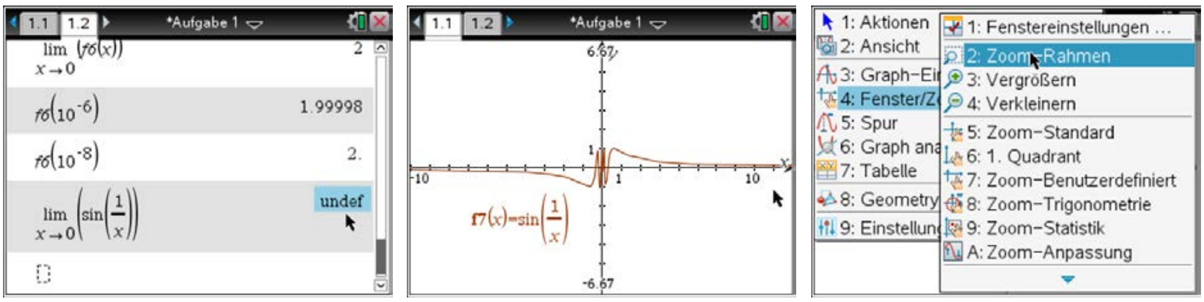
Im Rechenfenster wird der Grenzwert ohne Mühe gefunden.



Aufgabe 7 *Grenzwerte*

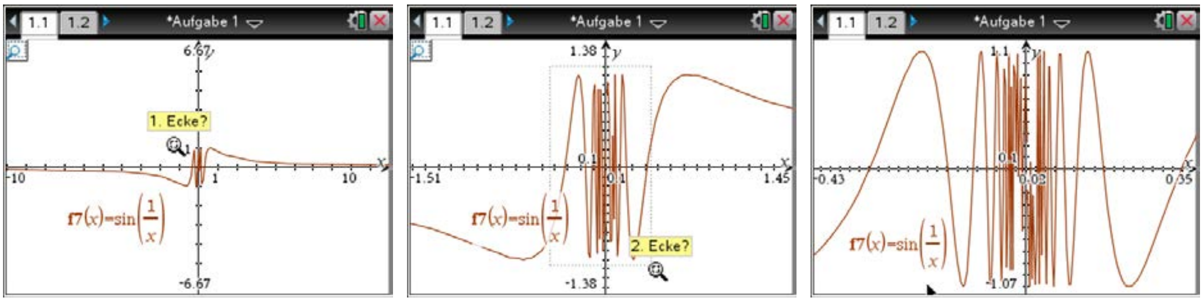
Der Grenzwert $\lim\limits_{x\to 0}\sin\dfrac{1}{x}$ existiert nicht. Versuche mit dem TI-Nspire™ CX CAS heraus zu finden, was an der Stelle $x = 0$ wirklich geschieht.

Lösung 7 Das ist ebenfalls ein klassisches Grenzwertproblem. Sobald $x \to 0$ geht, beginnt die Funktion immer rascher zu oszillieren. Mit dem TI-Nspire™ CX CAS kann man das zumindest einigermaßen illustrieren.

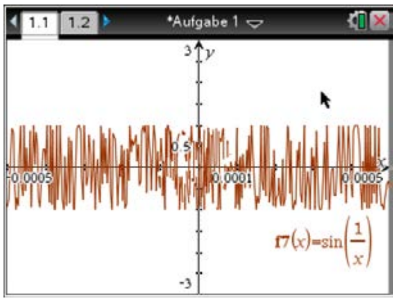


Zuerst testen wir, ob es da wirklich keinen Grenzwert gibt.

Der Graph im Standardfenster ist nicht sehr aufschlussreich. Mit der Option 2:Zoom-Rahmen können wir uns Schritt für Schritt ins Zentrum des Geschehens hinein versetzen und sehen tatsächlich, dass der Graph immer wilder nach oben und unten ausschlägt.



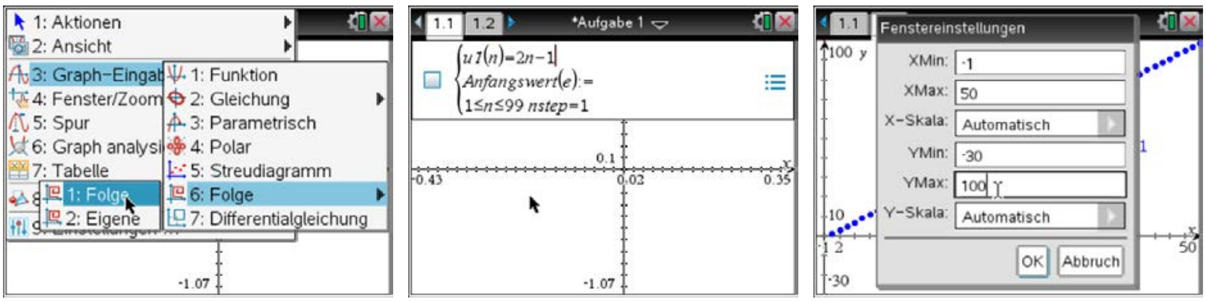
In der rechten Abbildung sehen wir, wie sich der Graph bei weiterer Annäherung an die Stelle $x = 0$ entwickelt.

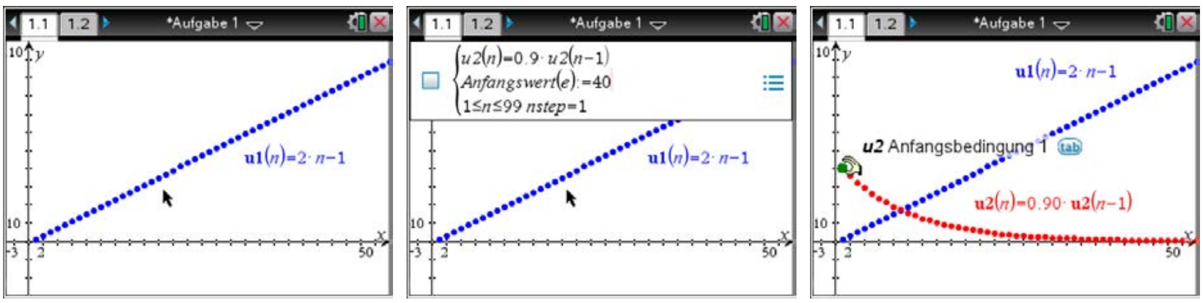


Aufgabe 8 *Folgen*

Erstelle die Graphen der Folgen $u_n = 2n - 1$ und $v_n = 0{,}9 \cdot v_{n-1}$ mit $v_1 = 40$.

Lösung 8 Die grafische Darstellung von Folgen geschieht so, dass wir zuerst die Eingabe für den Graph auf die Eingabe einer Folge über menu > 3.Graph-Eingabe/Bearbeitung > 6:Folge > 1:Folge umstellen.



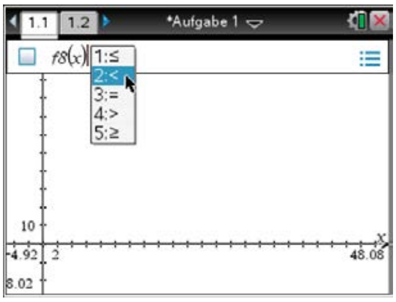


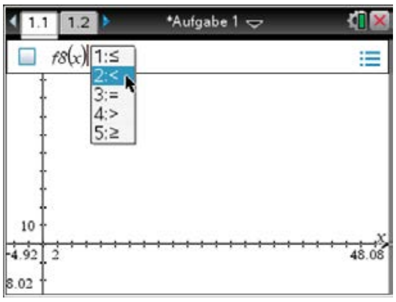
Wenn wir keinen Anfangswert angeben, wird der Anfangswert 1 angenommen. Ein abweichender Anfangswert wird durch einen grünen Punkt ausgezeichnet.

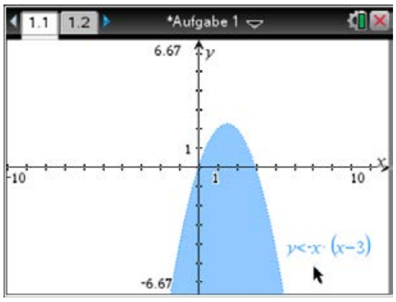
Aufgabe 9 *System von Ungleichungen*

Löse grafisch das Ungleichungssystem $\begin{cases} y + x(x-3) < 0 \\ y > x^2 - 2 \end{cases}$.

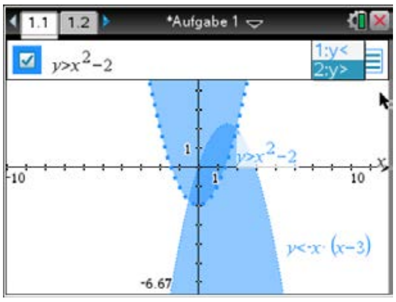
Lösung 9 Das können wir mit dem TI-Nspire™ CX CAS sehr einfach erledigen. Zuerst müssen die Ungleichungen in die Form $y <, \leq, >$ oder $\geq f(x)$ gebracht werden. Im Grafikfenster werden wir in der Eingabezeile im vorliegenden Text ´$f8(x) =$´ das Gleichheitszeichen löschen, dann öffnet sich ein Fenster mit allen Relationszeichen. Wir wählen das geeignete Zeichen und schreiben die Ungleichung fertig hin. Das ergibt eine nach unten geöffnete Parabelfläche

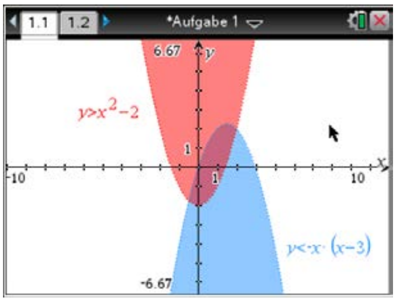


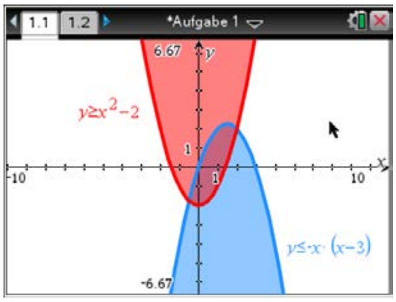




Anschließend geben wir auf die gleiche Weise die zweite Ungleichung ein und erhalten eine weitere Parabelfläche. Die Bereiche können in verschiedenen Farben gefüllt werden.







Für das letzte Bild wurden die Relationszeichen auf ≤ und ≥ geändert. Damit gehören die Bereichsränder zum Lösungsbereich. Vorher waren die Bereichsränder punktiert eingezeichnet. Wie wir sehen lässt sich die Farbe der Bereiche und deren Ränder unabhängig von einander ändern.

3 Etwas schwierigere Aufgaben

3.1 Standardaufgaben

Aufgabe 10 *Gerade und ungerade Funktionen*
Die folgende Behauptung sagt, dass sich jede Funktion als Summe aus einer geraden und einer ungeraden Funktion darstellen lässt.

Behauptung 1: *Sei* $f:\mathbb{R}\to\mathbb{R}$ *eine beliebige Funktion und*

$$f_g(x)=\frac{f(x)+f(-x)}{2} \quad \textit{sowie} \quad f_u(x)=\frac{f(x)-f(-x)}{2},$$

dann ist f_g gerade und f_u ungerade und $f(x)=f_g(x)+f_u(x)$.

Beweise diese Behauptung. Verifiziere die Behauptung mit einigen Polynomen. Woher stammen die Bezeichnungen „gerade“ und „ungerade“? Was geschieht, wenn *f* selbst eine gerade oder eine ungerade Funktion ist? Setze TI-Nspire™ CX CAS dazu ein, um für eine beliebige – vielleicht recht „exotische“ – Funktion f f_g und f_u zu erzeugen und graphisch darzustellen. Für $f(x)=e^x$ ist $f_g(x)=\cosh x$ und $f_u(x)=\sinh x$. Überzeuge dich davon in Aufgabe 11.

Lösung 10 Der Beweis der Behauptung ist eine einfache Verifizierung. Wenn man aber die Behauptung formuliert wie folgt, ist der Beweis ausführlicher und führt dann automatisch zu den oben angeführten Formeln.

Behauptung 2: *Jede Funktion ist die Summe von einer geraden und einer ungeraden Funktion.*

Beweis: Es gilt, dass $f(x)=f_g(x)+f_u(x)$, wobei f_g gerade und f_u ungerade ist. Dann erhalten wir:

$$\begin{cases} f(x)=f_g(x)+f_u(x) \\ f(-x)=f_g(x)-f_u(x) \end{cases}$$

Wenn wir dieses Gleichungssystem nach f_g und f_u auflösen, erhalten wir die oben angeführten Formeln.

Wenn $f(x)$ ein Polynom ist, dann wird f_g von allen Termen geraden Grades und f_u von allen Termen ungeraden Grades gebildet. Wenn *f* selbst eine gerade Funktion ist, dann gilt $f_g(x)=f(x)$ und $f_u(x)=0$. Der gleiche Schluss wird gezogen, wenn *f* eine ungerade Funktion ist.

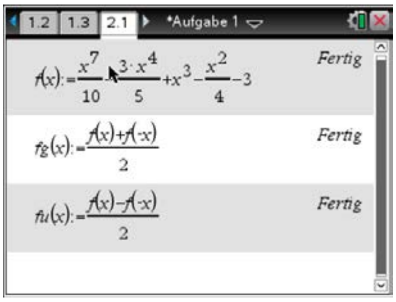
Aufgabe 11 Die Behauptung lässt sich mit TI-Nspire™ CX CAS sehr leicht illustrieren:

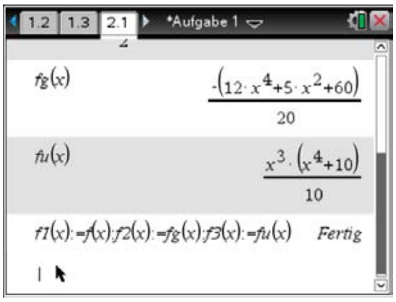
Wir wählen a) ein Polynom $f(x)=\frac{x^7}{10}-\frac{3x^4}{5}+x^3-\frac{x^2}{4}-3$

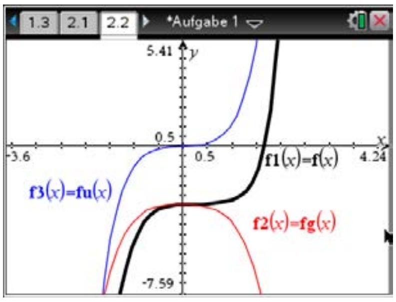
b) eine willkürliche Funktion $f(x)=\frac{2\sin(2x)-x^2+e^{-x}}{x^2+1}$

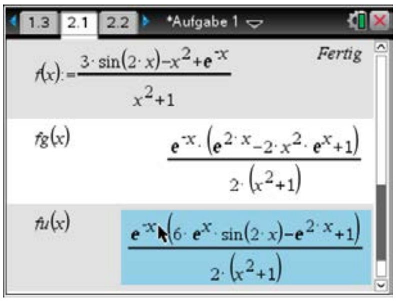
c) die Exponentialfunktion $f(x)=e^x$

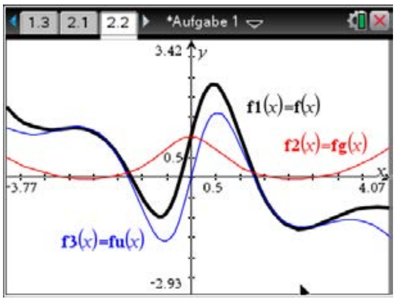
Lösung 11 Die Bildschirme sollen für sich sprechen.

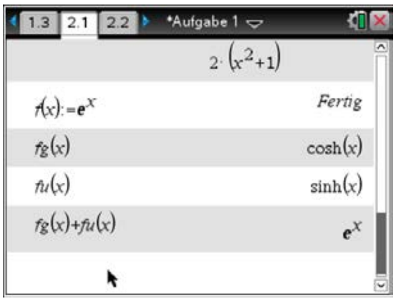


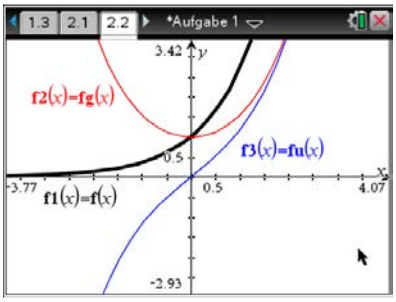






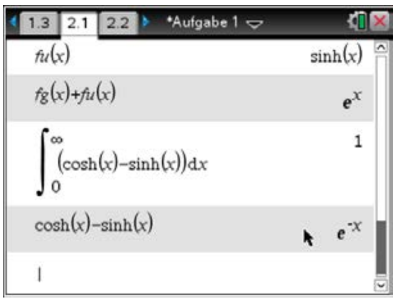


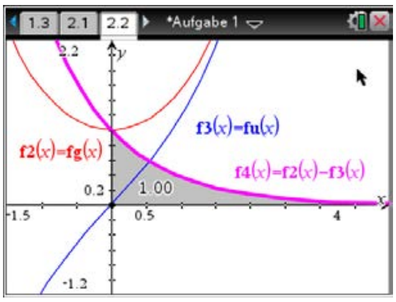




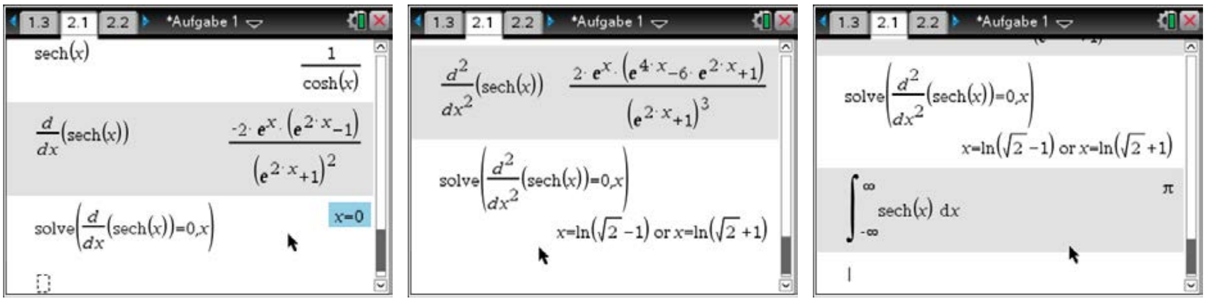
Das letzte Beispiel nehmen wir zum Anlass, um über die *hyperbolischen Funktionen* sinh(x) und cosh(x) zu sprechen. Wenn auch manchmal diese Funktionen als „von weit hergeholt" bezeichnet werden, haben sie doch viele Anwendungen, wie z.B. in der Physik (Kettenlinie zur Beschreibung von durchhängenden Ketten und Seilen) oder auch in der Architektur (Bauten von Gaudi).

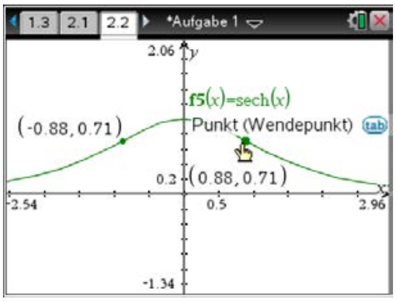
Zusatzaufgabe: Berechne $\int_0^\infty \cosh(x) - \sinh(x)\,dx$ und erzeuge eine zugehörige graphische Darstellung. In diesem Fall ist die händische Berechnung interessanter als der Einsatz des CAS.





Zusatzaufgabe: Führe eine Funktionsdiskussion für $\operatorname{sech}(x) = \dfrac{1}{\cosh(x)}$ durch. Auch hier ist die Durchführung mit Papier und Bleistift interessanter als mit dem Einsatz des Rechners.





Aufgabe 12 *Tangenten*

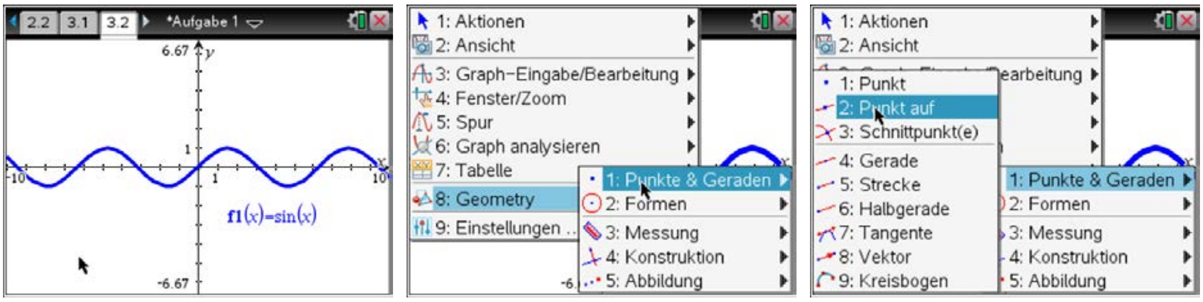
Finde die Tangente an den Graph von $y = \sin x$ an der Stelle $a = \frac{\pi}{4}$. Stelle den Graph gemeinsam mit der Tangente mit TI-Nspire™ CX CAS dar.

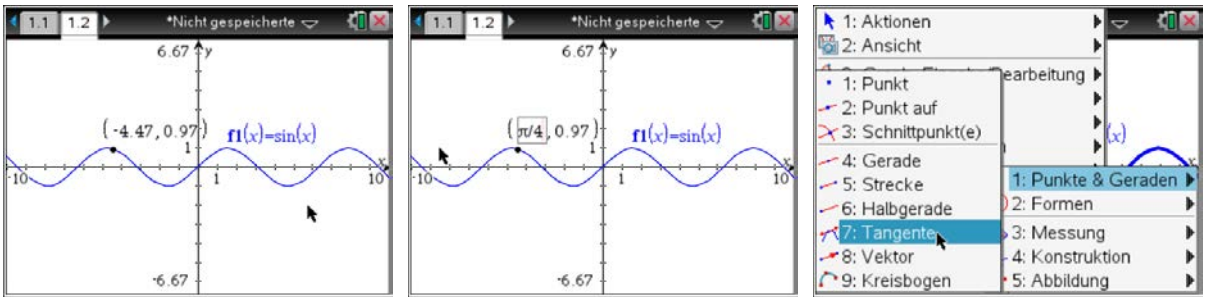
Lösung 12 Als erstes wollen wir die Lösung im Rechenfenster bestimmen.

Zuerst wenden wir die bekannte Tangentengleichung an: $y - y(a) = f'(a) \cdot (x - a)$. Im Katalog unter den Funktionen zur Analysis finden wir ein geeignetes Werkzeug zur direkten Berechnung der Tangente. Die Gleichung wird in beiden Fällen exakt – aber interessanterweise in unterschiedlicher Form – ausgegeben. Die numerische Approximation zeigt die Identität der beiden Terme.

Nun zur grafischen Vorgangsweise:

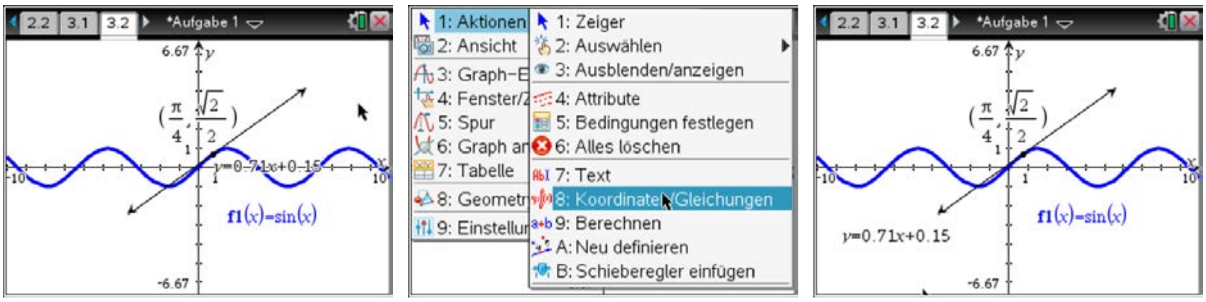
Wir zeichnen den Graph und legen dann über menu > 8:Geometry > 1:Punkte & Geraden > 2:Punkt auf einen beliebigen Punkt auf der Sinuslinie fest. Nach einem Doppelklick auf die x-Koordinate des Punkts kann der Wert auf $\dfrac{\pi}{4}$ geändert werden, worauf dieser Punkt sofort den gewünschten Platz am Graph einnimmt.





Unter 1:Punkte & Geraden finden wir die Option 7:Tangente. Wir führen den Cursor zum Punkt und wenn der Cursor dann zu einem kleinen Händchen wird, das zum Punkt weist wird nach einem [enter] die Tangente mit zwei Pfeilspitzen eingezeichnet. Die Pfeilspitzen lassen sich in beide Richtungen verlängern.

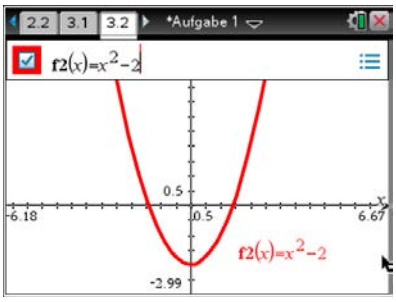
Zur– allerdings nur gerundeten numerischen – Tangentengleichung gelangen wir, indem wir über [menu] > 1:Aktionen > 8:Koordinaten/Gleichungen anschließend die Tangente ansteuern bis die Gleichung gezeigt wird, die dann mit [enter] fixiert wird.

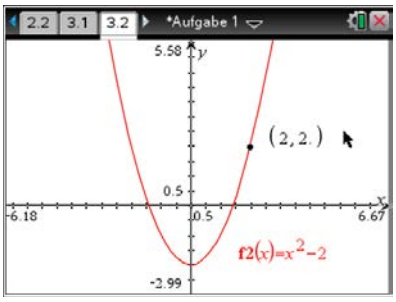


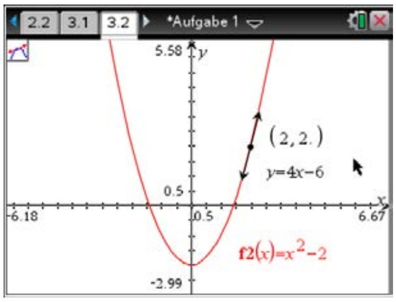
Aufgabe 13 *Tangenten – Irrationale Zahlen*
Wir wissen, dass sich irrationale Zahlen nicht in Bruchform darstellen lassen. Man kann sie allerdings durch einen Bruch approximieren. Heute ist das kein Problem, da Rechenmaschinen wie TI-Nspire™ CX CAS mit irrationalen Zahlen exakt umgehen können. Früher war das ein echtes Problem. Eine Möglichkeit, irrationale Zahlen näherungsweise durch rationale zu ersetzen war es, die Nullstelle der Tangente an eine geeignete Kurve zu berechnen. Suche einen Näherungswert für $\sqrt{2}$ mithilfe der Tangente an den Graph von $y = x^2 - 2$ an der Stelle $x = 2$. Mit dieser Methode lässt sich die n-te Wurzel jeder natürlichen Zahl näherungsweise finden. Versuch das doch einmal!

Lösung 13 Die Nullstellen von $f(x) = x^2 - 2$ sind $-\sqrt{2}$ und $\sqrt{2}.$ Die erste Ableitung ist $f'(x) = 2x.$ Die allgemeine Tangentengleichung an den Graph von f and der Stelle $x = a$ ist gegeben durch (siehe auch Aufgabe 11) $y = f'(a)\cdot(x-a) + f(a),$ was in unserem Fall zur Tangentengleichung

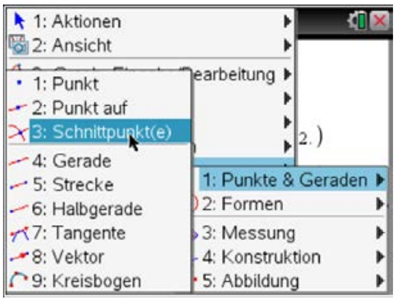
$y = 4x - 6$ führt. Ihr Schnittpunkt mit der *x*-Achse ist ein Näherungswert für $\sqrt{2} \approx \frac{3}{2}$. Das wollen wir nun mit dem TI-Nspire™ CX CAS sehen lassen. (Einige Tätigkeiten wurden bereits in Aufgabe 11 erklärt und durchgeführt.)

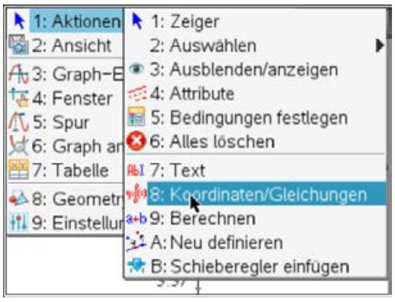


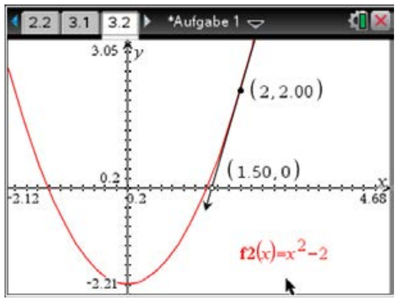




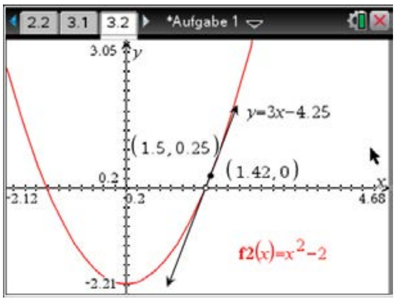
Die Tangente wird verlängert und über über menu > 8:Geometry u.s.w. mit der *x*-Achse zum Schnitt gebracht. Anschließend werden die Koordinaten des so ermittelten Punkts gesucht.

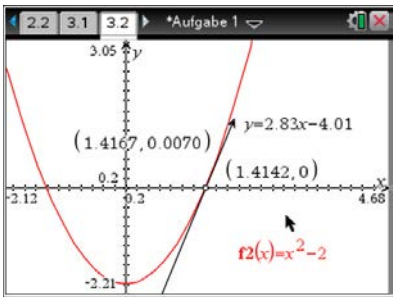


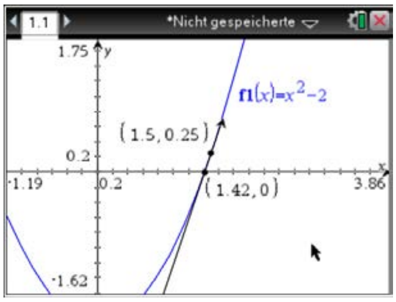




Der erste Näherungswert ist daher 1,5. Es ist nun leicht, den Vorgang zu wiederholen, indem man anstelle von $a = 2$ den gewonnenen Näherungswert $a = 1{,}5$ verwendet. Es ist nur der Wert 2 im Koordinatenpaar des Ausgangspunkts zu überschreiben.





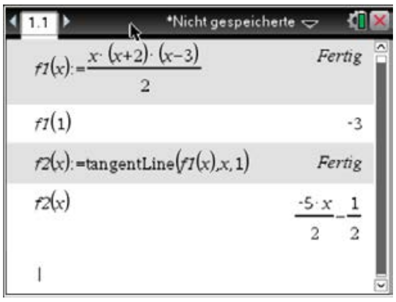


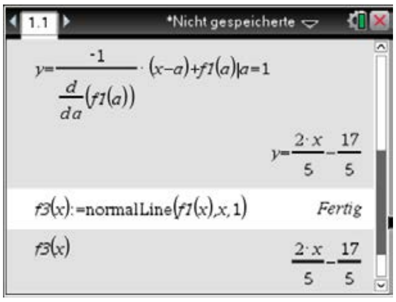
Damit ist ein besserer Näherungswert mit 1,42 gefunden worden. Sobald man diese Methode automatisiert und weitere Schritte durchführt (iterativ vorgeht), spricht man von der *newtonschen Näherungsmethode* zur numerischen Lösung von Gleichungen (siehe auch später die entsprechende Aufgabe 37). Ich habe über menu > 9:Einstellungen die Ausgabe auf 4 Dezimalstellen erhöht und eine weitere Approximation durchgeführt. Damit erreiche ich den bekannten Näherungswert für $\sqrt{2} \approx 1{,}4142$.

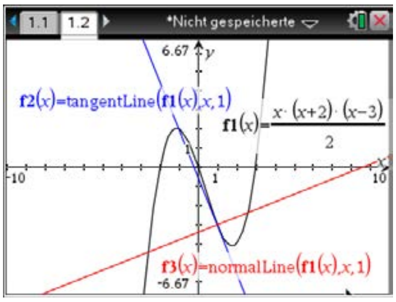
Aufgabe 14 *Kurventangente und Kurvennormale*

Erzeuge den Graph der Funktion $y=\dfrac{x\cdot(x+2)\cdot(x-3)}{2}$ gemeinsam mit der Kurventangente und Kurvennormale an der Stelle x = 1. Führe ihre Berechnung durch und verwende dazu die von TI-Nspire™ CX CAS bereitgestellten Werkzeuge.

Lösung 14 Die Tangente wird ermittelt wie in Aufgabe 12. Für die Kurvennormale gilt, dass ihr Anstieg (die Steigung) an der Stelle $x = a$ gegeben ist mit $-\dfrac{1}{f'(a)}$. Beachte, dass im Zeichenfenster auf beiden Achsen die gleiche Skalierung eingestellt ist (menu > 4:Fenster > 5:Zoom-Standard).







Aufgabe 15 *Die Ableitung*
Stelle mit TI-Nspire™ CX CAS den Graph einer Funktion gemeinsam mit dem Graph ihrer ersten Ableitung dar. Zeige, dass die Extremwerte der Funktion an den Nullstellen mit ungerader Vielfachheit der Ableitung liegen und dass ein allfälliger Wendepunkt der Funktion an einer Extremstelle der Ableitung liegt. Eine schiefe Asymptote der Funktion gehört zu einer waagrechten Asymptote der Ableitung.

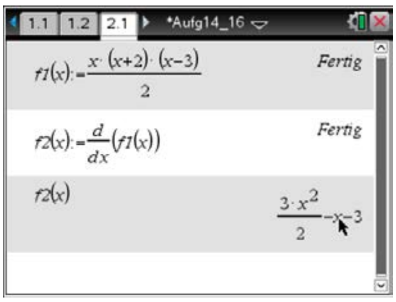
Zusatzaufgabe Untersuche den Zusammenhang zwischen der Näherungskurve der Funktion und der Näherungskurve der Ableitung für eine rational gebrochene Funktion

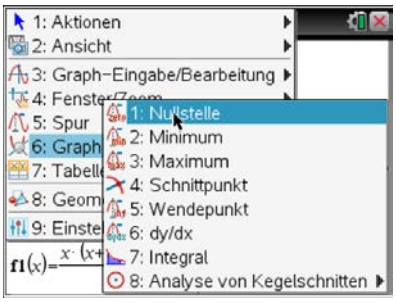
Lösung 15 Wir demonstrieren dies an den folgenden Funktionen:

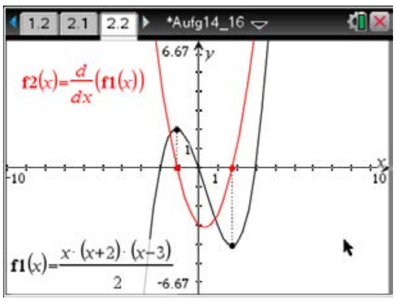
a) $f(x)=\dfrac{x\cdot(x+2)\cdot(x-3)}{2}$ b) $f(x)=\dfrac{(x-3)^3(x+1)}{20}-3$

c) $f(x)=\dfrac{3x^2-8}{2x}$ d) $f(x)=\dfrac{3x^3-x-5}{2x+1}$

a)

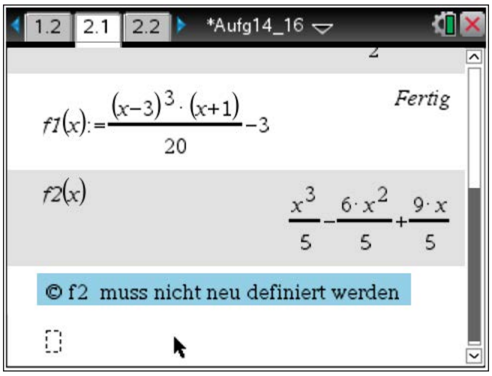


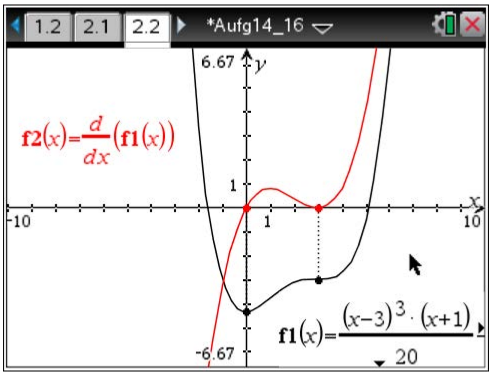




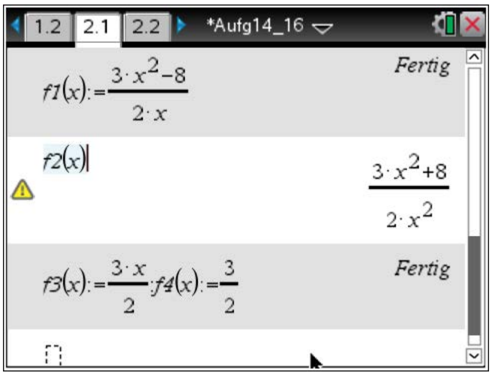
Nullstellen, Extremwerte und Wendpunkte finden wir über Graph Analysieren. Die Verbindungsstrecken zwischen den Punkten werden im menu in der Option 8:Geometry > 1:Punkte & Geraden erzeugt. Über die Attribute können die senkrechten Strecken (Ordnerlinien) punktiert dargestellt werden.

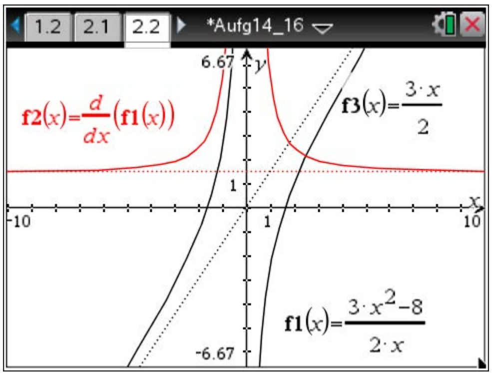
b)



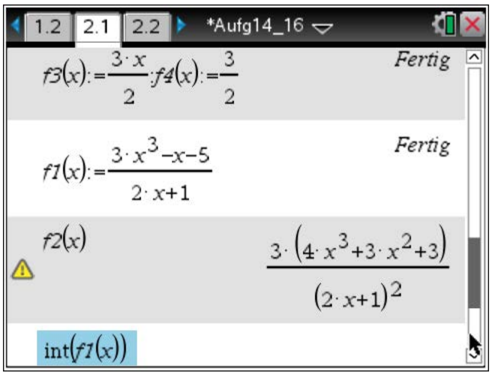


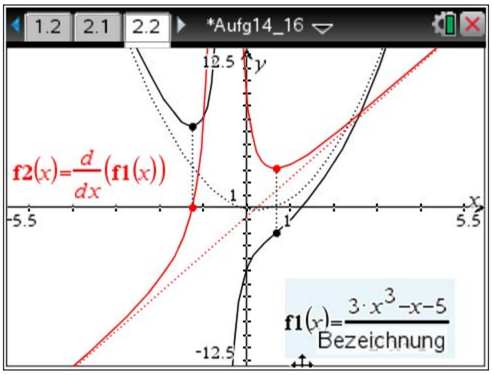
c)



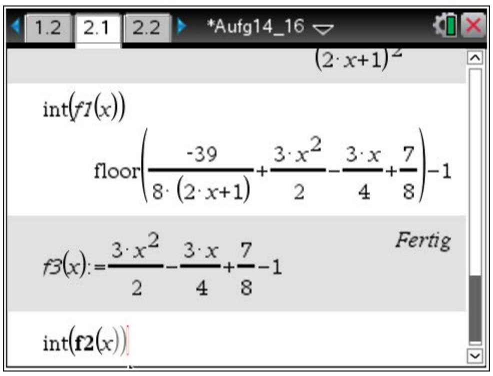


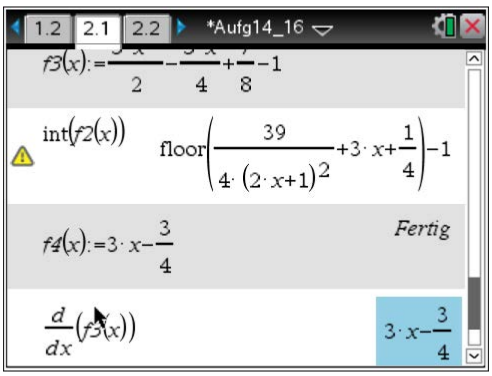
d)





Die Gleichungen der Näherungskurven sind der ganzzahlige Teil der ausgeführten Division Zählerterm durch Nennerterm. Mit der int-Funktion können wir diese Division durch TI-Nspire™ CX CAS durchführen lassen.



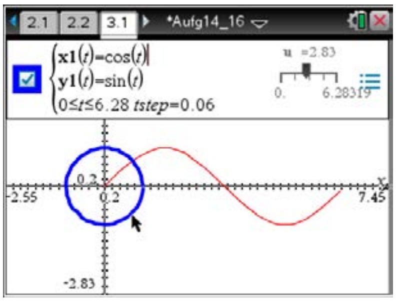


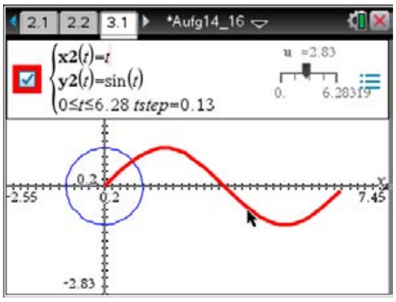
Wir erkennen, dass die Näherungskurve der Ableitungsfunktion offensichtlich die Ableitung der Näherungskurve der Ausgangsfunktion ist. Alle anderen Eigenschaften sind durch übereinander liegende Punktepaare dargestellt. Führe die Polynomdivision auch ohne CAS durch!

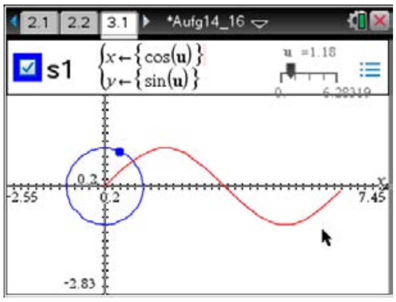
Aufgabe 16 *Winkelfunktionen*

Es ist eine schöne Übung, den Einheitskreis gemeinsam mit einer oder mehreren Winkelfunktionen grafisch darzustellen. Verwende dazu die Parameterdarstellungen um die Graphen von (cos(t), sin(t)) und (t, cos(t)) bzw. (t, sin(t)) zu erzeugen.

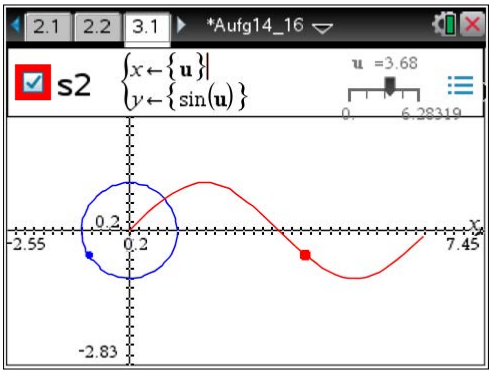
Lösung 16 Wir zeigen zuerst die Sinusfunktion.

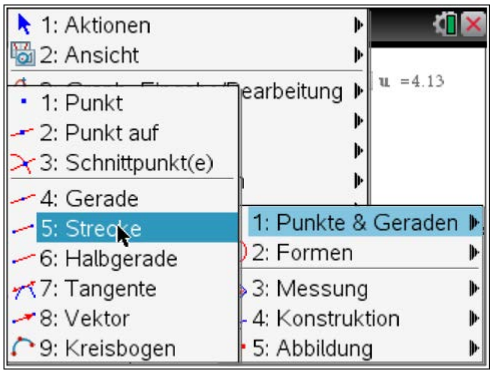




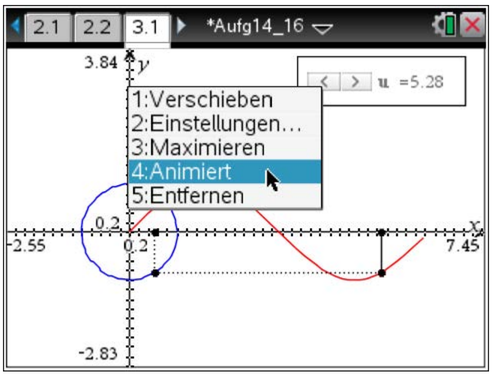


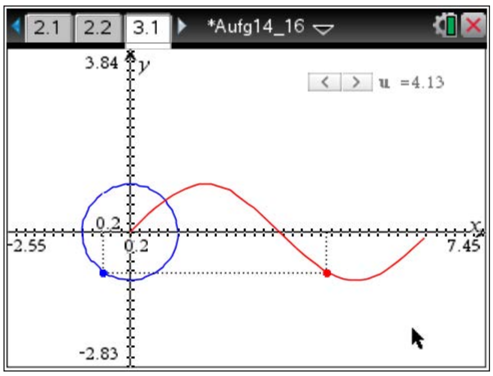
Besonders eindrucksvoll wird die Grafik, wenn wir sie mit einem Schieberegler verbinden, der einen Punkt auf dem Kreis bewegt.





Über das Geometry-Menü können wir noch die passenden Strecken eintragen. Abschließend lässt sich der Schieberegler noch minimieren und schließlich auch animieren und wir können sehr schön beobachten, wie die Sinusschwingung entsteht. (Zur Animation siehe Abschnitt 5.2)



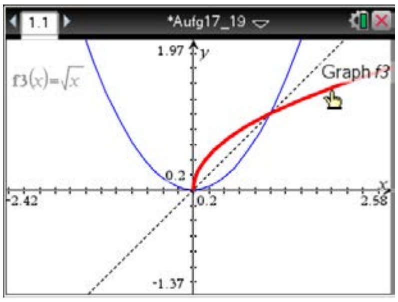


Aufgabe 17 *Die Umkehrfunktion*

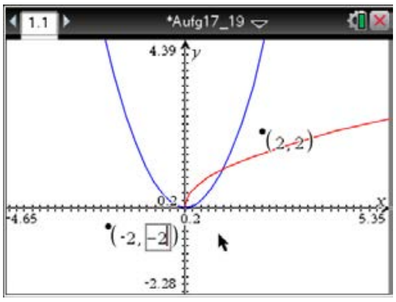
Erzeuge den Graph einer Funktion und ihrer Umkehrfunktion, wie z.B. $y = x^2$ und $y = \sqrt{x}$ oder $y = \cos x$ und $y = \arccos x$. Trage auch den Graph der 1. Mediane (Symmetrale des 1. und 3. Quadranten) ein. Wie liegen die Graphen bezüglich der Mediane? Das gilt allerdings nur für einen Teil des Graphs. Versuche die Möglichkeiten von menu > 8:Geometry, um den Graph der Funktion vollständig zu spiegeln. Damit erhältst du den Graph der *Umkehrrelation*. Ist das auch der Graph einer Funktion? Kannst du nun einsehen, warum man für die inverse Funktion (= Umkehrfunktion) Wertebereich und Bildbereich angeben muss?

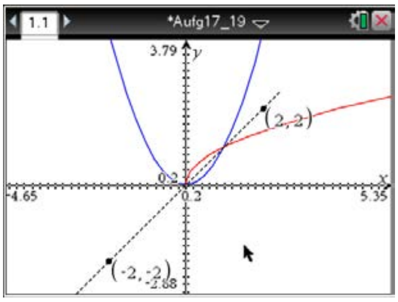
Der Graph der Umkehrfunktion (Umkehrrelation) kann leicht über die Parameterdarstellung gewonnen werden. Zeige dies an zwei selbst gewählten Beispielen.

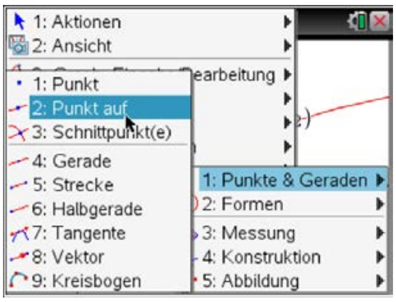
Lösung 17 Erzeuge die Graphen wie beschrieben.

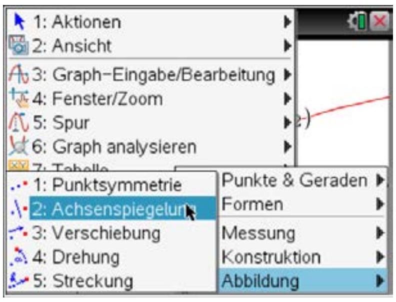


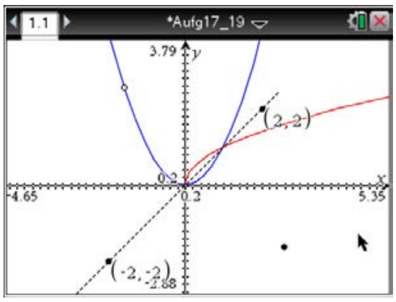
Wenn wir die Ausgangsfunktion (blau) an der Winkelhalbierenden spiegeln wollen, müssen wir das Geometriewerkzeug wählen – und hier können wir nicht über die als Gleichung definierte Gerade arbeiten, sondern müssen die Mediane als eine mit zwei Punkten konstruierte Gerade erzeugen.



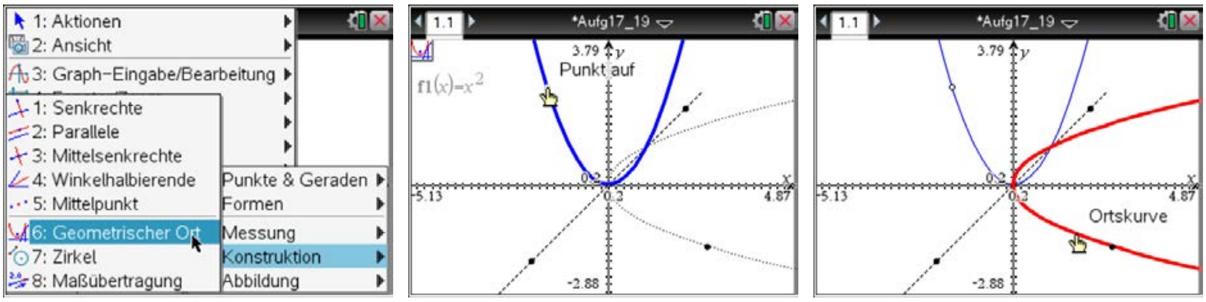




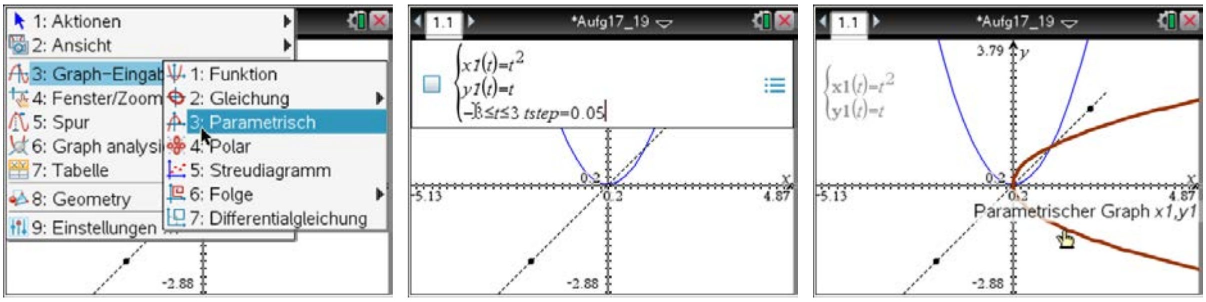




Leider können wir den Graph der Funktion nicht als Objekt spiegeln. Aber wir können uns eines einfachen Tricks bedienen: wir spiegeln erst einen beliebigen Punkt der Kurve an der Mediane und dann lassen wir uns die Ortslinie dieses Punktes konstruieren, die entsteht, wenn sich der erst gewählte längs der Parabel bewegt. Im Geometry-Menü finden wir 4:Konstruktion > 6:Geometrischer Ort.



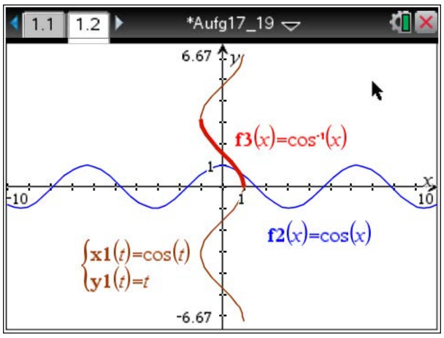
Die Gleichung der Umkehrfunktion (-relation) erhält man, indem man *x* und *y* vertauscht und die so entstandene Gleichung nach *y* auflöst. *x*- und *y*-Werte vertauscht man ganz leicht in der Parameterdarstellung . Aus ($x = t$, $y = t^2$) wird dann eben ($x = t^2$, $y = t$).

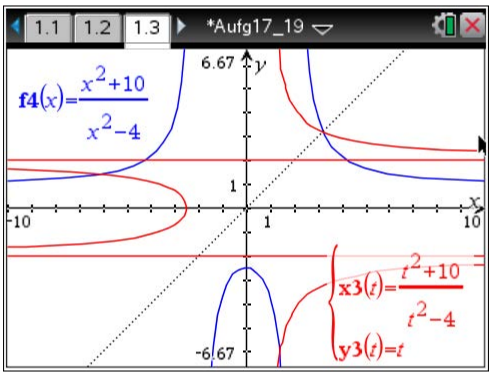


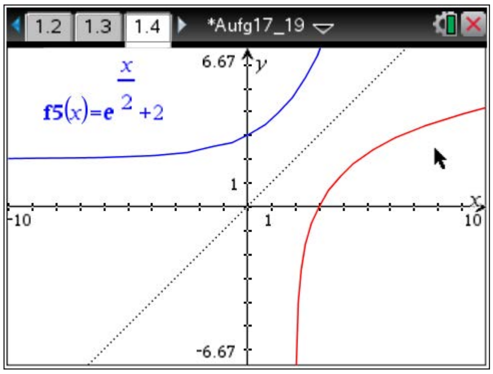
Wir wiederholen diese Prozedur mit $f(x) = \cos x$ über die Ortslinie oder über die Parameterdarstellung.

Sehr deutlich ist der Unterschied zwischen der Umkehrfunktion $\arccos(x) = \cos^{-1}(x)$ und der Umkehrrelation zu erkennen.

Es folgen zwei weitere Beispiele:







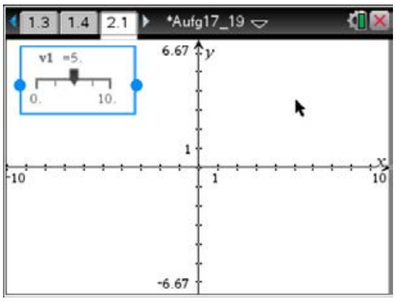
Im letzten Beispiel gibt es eine Umkehr**funktion**. Ihre Gleichung kannst du sicher auch ohne CAS ermitteln.

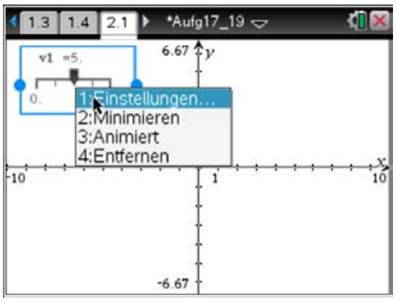
Aufgabe 18 *Folgen – das Modell von Verhulst*

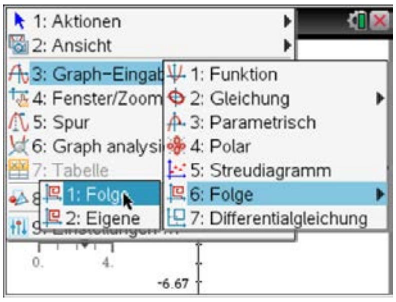
Erstelle den Graph zur Folge $u_n = a \cdot u_{n-1} \cdot (1 - u_{n-1})$ mit $u_1 = 0{,}2$ für verschiedene Werte von a mit $0 \le a \le 4$. Erzeuge auch die zugehörigen Spinnwebdiagramme.

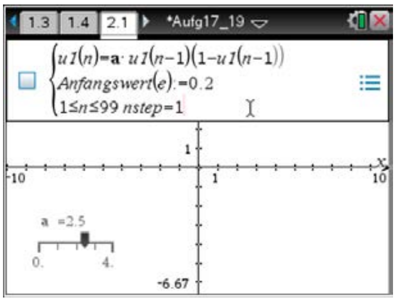
Lösung 18 Die rekursiv definierte Folge $u_n = a \cdot u_{n-1} \cdot (1 - u_{n-1})$ ist in der Biologie als *Verhulst-*Modell bekannt. Ihr Konvergenzverhalten ändert sich für verschiedene Werte von *a* vollständig. Dazu gibt es viel Literatur – auch im Internet zu finden.

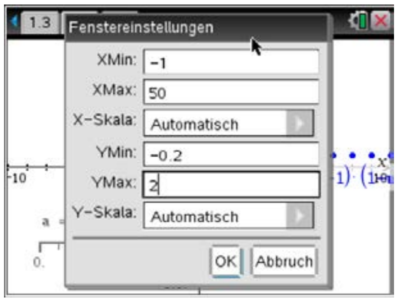
Um die Abhängigkeit vom Parameter *a* zu zeigen verwenden wir einen Schieberegler, der über menu > 1:Aktionen > B:Schieberegler einfügen in einem Grafikfenster eingerichtet wird. Sowohl die Grenzen für den Parameter als auch die Schrittweite können festgelegt werden. Die Parameterwerte werden dann durch die Bewegung des Schiebereglers verändert.

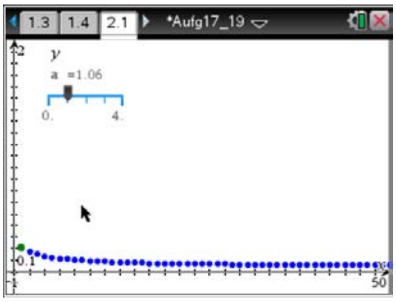


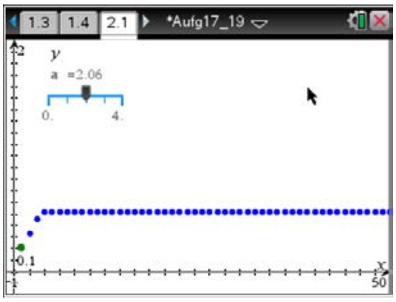


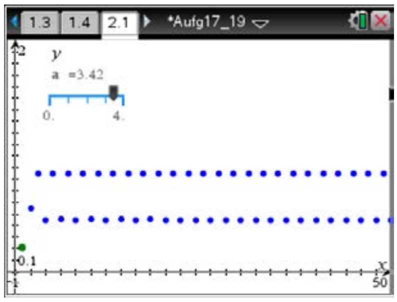


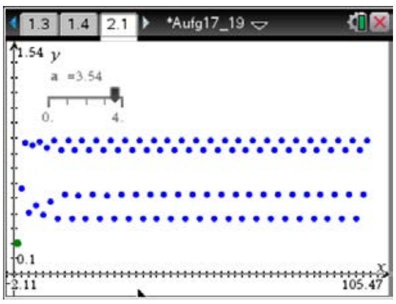


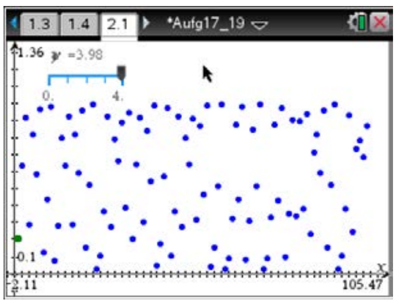




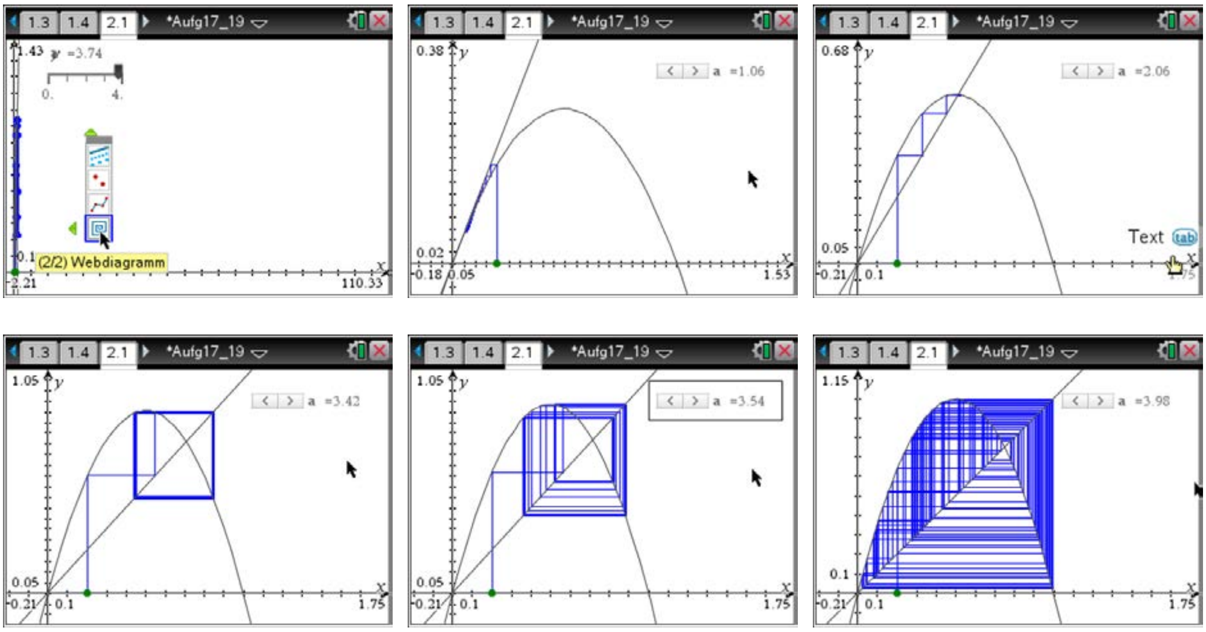






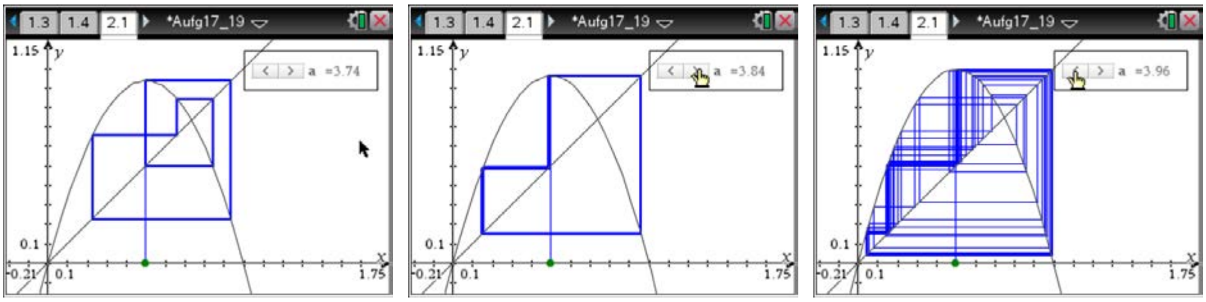


Die Umstellung zur Darstellung als Spinnwebdiagramm (Cobweb Diagram) erfolgt über die Attribute. Diese Diagramme zeigen auf andere Art die *Attraktoren* der Folge abhängig vom Parameter *a* – bis eine *chaotische* Entwicklung eintritt. Wir erkennen zuerst einen, dann zwei und vier Attraktoren und dann wird's „chaotisch“.



Der Schieberegler kann animiert werden, wie im Abschnitt 5.2. nochmals genauer erläutert wird.

Natürlich lässt sich auch der Startwert u_1 variieren. Hier sind drei Bilder für den Anfangswert 0,5:

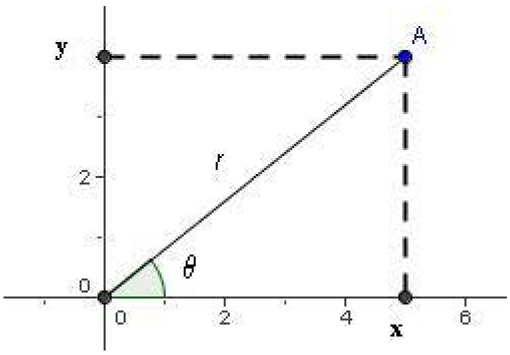


3.2 Weitere Koordinatensysteme

Polarkoordinaten

In der Schulmathematik nimmt das kartesische Koordinatensystem eine zentrale Rolle ein. Das liegt wohl auch daran, dass in der Analysis vor allem Funktionen der Form $y = f(x)$ behandelt werden. In den Naturwissenschaften (und auch in der Mathematik) kommen aber auch einige andere Koordinatensysteme vor.

Die *Polarkoordinaten* haben eine geometrische Interpretation, die geeignet ist, im Zusammenhang mit der trigonometrischen Form der komplexen Zahlen behandelt zu werden. Jeder Punkt A in der Zahlenebene kann festgelegt werden durch seine Koordinaten (x, y) in einem kartesischen (rechtwinkligen) Koordinatensystem oder durch seinen Abstand r vom Koordinatenursprung gemeinsam mit dem Winkel θ, den der Strahl $\overrightarrow{OA}$ mit der positiven x-Achse einschließt.



Der Winkel θ kann in den Intervallen $[0, 2\pi]$, $]-\pi, \pi]$ oder auch in $\mathbb{R}$ liegen, In diesem Zusammenhang lassen sich auch Funktionen behandeln:

kartesisch	polar
(x, y)	(r, θ)
$y = f(x)$	$r = f(\theta)$

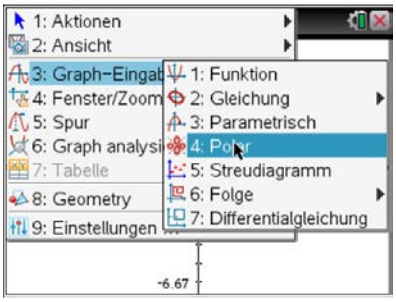
Die Transformation von kartesischen zu Polarkoordinaten erfolgt mittels der wohl bekannten Formeln:

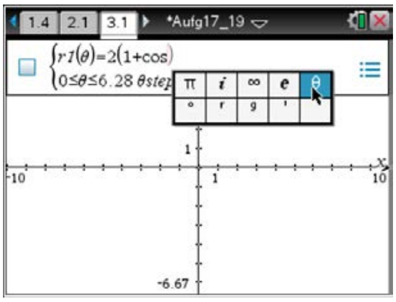
$$\begin{cases} x = r\cos\theta \\ y = r\sin\theta \end{cases}$$

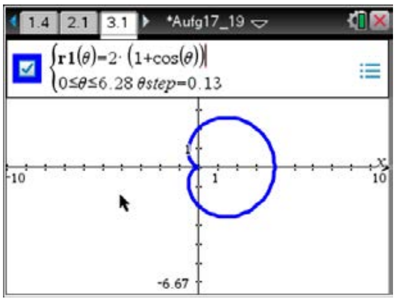
In den Wissenschaften kommt es oft vor, dass eine Kurve nicht in kartesischer Form $y = f(x)$ sondern in Polarform $r = f(\theta)$ vorliegt. Auch in diesem Fall kann TI-Nspire™ CX CAS helfen.

Aufgabe 19 *Die Kardioide*
Erzeuge den Graph von $r = 2(1+ \cos\theta)$. Warum wird diese Kurve *Kardioide* genannt? Hast du diese Kurve schon irgend wo beobachten können?

Lösung 19 Verwende im Grafikfenster menu > 3:Graph-Eingabe/Bearbeitung > 4:Polar um die Funktionsvorschrift einzugeben.





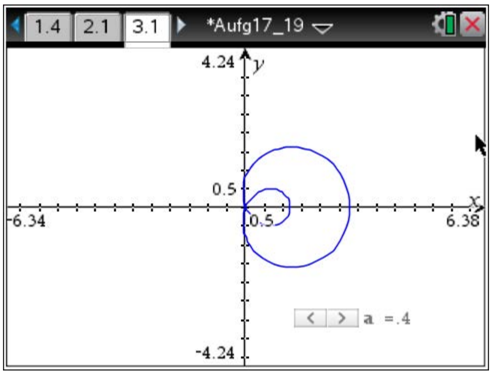


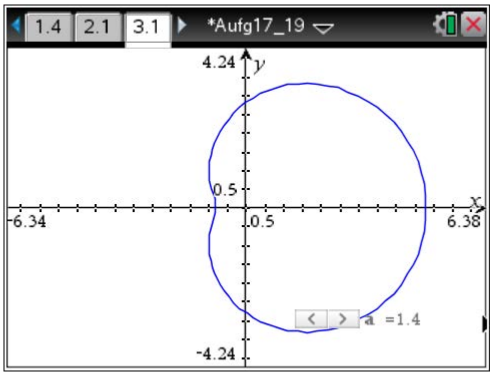
Diese Kurve kannst du täglich in der Kaffee- (oder Tee-, usw.)tasse beobachten. Trink zuerst die Tasse leer und halt sie dann in einem Abstand von ca 30 cm von einer Lichtquelle. Der Lichtschein soll über den Rand der Tasse in diese hinein fallen. Die Kurve, die das Licht in der Tasse erzeugt ist eine Kardioide (siehe auch Aufgabe 47).

Aufgabe 20 *Die Kardioide mit Schieberegler*

Führe einen Schieberegler für die Variable a ($0 \le a \le 3$) ein und ändere den Funktionsterm auf $r = 2(a + \cos\theta)$. Beschreibe den Einfluss von a auf die Form der Kurve. (Für die Animation des Schiebereglers siehe Abschnitt 5.2.)

Lösung 20





Aufgabe 21 *Die kartesische Form der Kardioide*

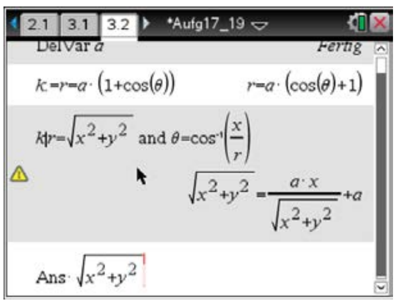
Die kartesische Form von $r = a(1+\cos\theta)$ lautet $(x^2+y^2)^2 - 2a\,x(x^2+y^2) = a^2y^2$. Versuche, diese Form zuerst händisch, dann mit Hilfe des CAS zu gewinnen.

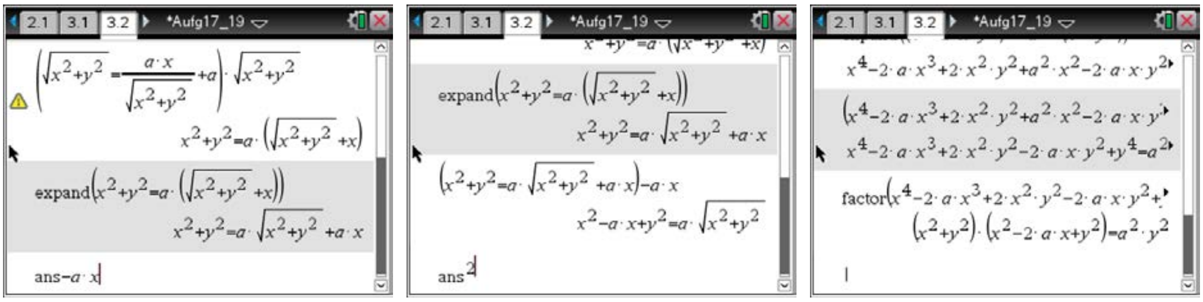
Lösung 21 Der erste Schritt besteht darin, den Term $\cos\theta$ in der Kardiodengleichung durch $\cos\theta$ aus $x = r\cos\theta$ zu ersetzen. Da $r = \sqrt{x^2+y^2}$ substituieren wir gleichzeitig für r

Wenn wir kein neues Problem eröffnen, müssen wir die Variable a (vom Schieberegler) erst löschen mit DelVar a.

Wir ersetzen $\theta = \arccos\dfrac{x}{r}$. und $r = \sqrt{x^2+y^2}$.

Das Ergebnis wird durch Multiplikation mit dem Wurzelausdruck bruchfrei gemacht.





Wir isolieren den Wurzelterm auf der rechten Seite, um durch Quadrieren des ganzen Ausdrucks die Wurzel zu entfernen. Damit sieht der Term ja schon recht ordentlich aus. Der Ausdruck wird ausmultipliziert (Expand) und a^2x^2 von beiden Seiten subtrahiert. Wenn wir nun noch faktorisieren sind wir dem Wunschergebnis schon sehr nahe. Daher geben wir den letzten Schliff mit Handarbeit:

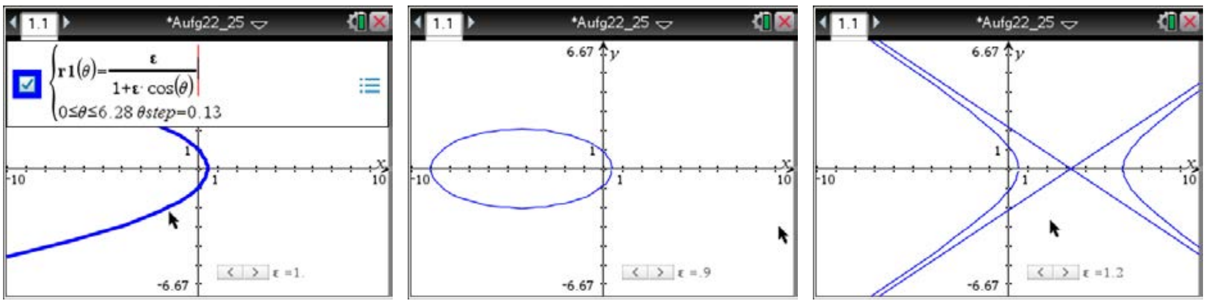
$$(x^2+y^2)(x^2-2a\,x+y^2)=a^2y^2$$
$$(x^2+y^2)((x^2+y^2)-2ax)=a^2y^2$$
$$(x^2+y^2)^2-2ax(x^2+y^2)=a^2y^2$$

Das ist reine Geschmacksache, welche Form der Formel „schöner“ ist.

Aufgabe 22 *Die Kegelschnitte in Polarkoordinaten*

Untersuche die Kurven, die durch die Polarform $r=\dfrac{\varepsilon}{1+\varepsilon\cdot\cos\theta}$ gegeben sind. Welchen Einfluss auf die Kurvenform hat der Parameter ε? (Diese Aufgabe eignet sich als Wettbewerb für Gruppen.)

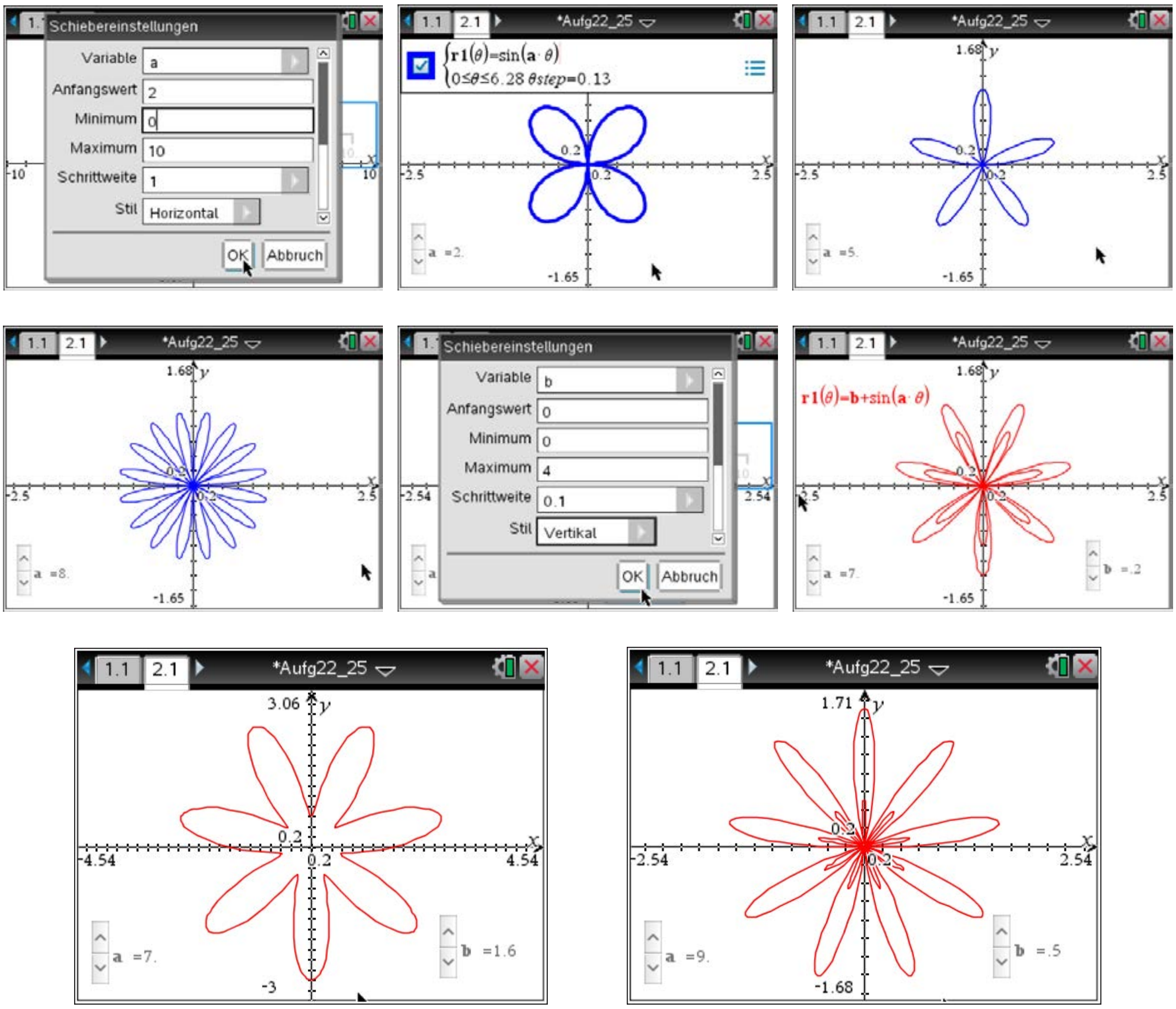
Lösung 22 Die gefragten Kurven sind alle Arten von Kegelschnitten. ε wird als Exzentrizität bezeichnet. Für $0<\varepsilon<1$ ergeben sich Ellipsen, für $\varepsilon=1$ erhält man eine Parabel und in allen anderen Fällen erscheint eine Hyperbel. Ein Schieberegler für ε drängt sich auf.



Auch hier erweist sich eine Animation des Schiebereglers als sehr informativ.

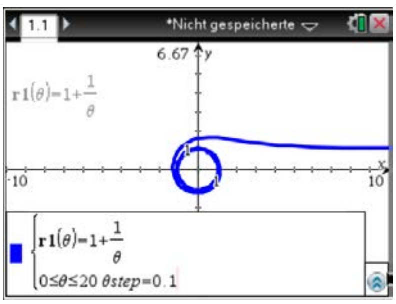
Aufgabe 23 *Blumen blühen im Grafikfenster*
Untersuche die Kurven, die durch die Polarform $r = \sin(a\,\theta)$ gegeben sind. Untersuche den Einfluss des Parameters $a \in \mathbb{N}_0$. Die Gleichung lässt sich noch verallgemeinern zu $r = b + \sin(a\,\theta)$.

Lösung 23 Für ungerade *a* haben die entstehenden „Blumen“ *a* „Blütenblätter“, für gerade Werte von *a* hingegen 2*a* – aber nur dann, wenn *b* < 1. Für *b* > 1 entsteht ein „Kern“ oder es gibt verschieden große Blütenblätter. Das Experimentieren mit Schiebereglern für *a* und *b* ist sehr reizvoll.



Aufgabe 24 *Polarkoordinaten und Asymptoten*
Erzeuge den Graph von $r(\theta) = 1 + \frac{1}{\theta}$ *mit* $\theta \in\,]0, +\infty[$. Überlege das mögliche asymptotische Verhalten und überprüfe dann mathematisch.

Lösung 24 Wir erzeugen den Graph und beachten dabei den Definitionsbereich für *θ*.



Die Schülerinnen und Schüler werden wohl die waagrechte Asymptote $y = 1$ erkennen. Es gibt aber außerdem noch einen „asymptotischen Kreis“.

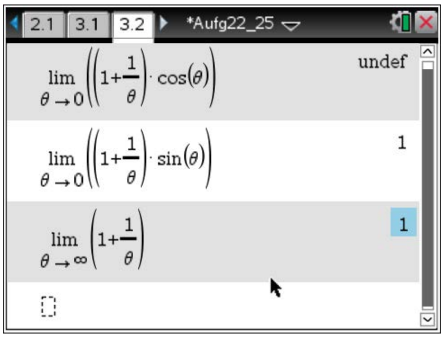
Die waagrechte Asymptote wird durch die nachfolgenden Rechnungen leicht bestätigt:

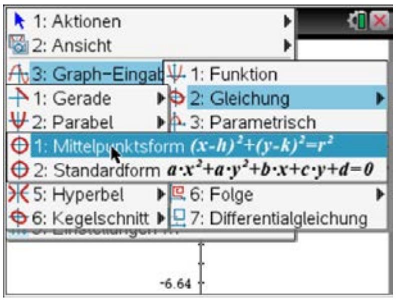
$$\lim_{\theta\to 0} x = \lim_{\theta\to 0} r\cos\theta = \lim_{\theta\to 0}\left(1+\frac{1}{\theta}\right)\cos\theta = +\infty$$

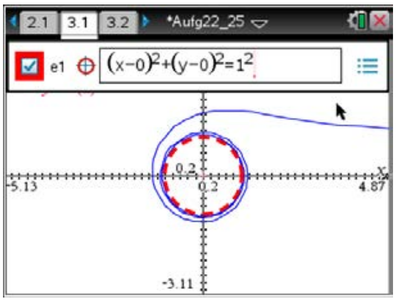
$$\lim_{\theta\to 0} y = \lim_{\theta\to 0} r\sin\theta = \lim_{\theta\to 0}\left(1+\frac{1}{\theta}\right)\sin\theta = \lim_{\theta\to 0}\sin\theta + \lim_{\theta\to 0}\frac{\sin\theta}{\theta} = 1$$

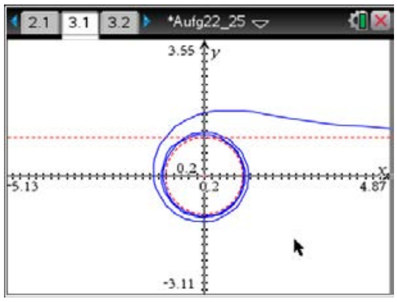
Der „asymptotische Kreis“ ergibt sich aus dem Umstand, dass sich der Punkt (r, θ) für $\theta \to \infty$ um den Koordinatenursprung dreht. Man kann den Grenzwert berechnen:

$$\lim_{\theta\to\infty} r = \lim_{\theta\to\infty} 1+\frac{1}{\theta} = 1.$$









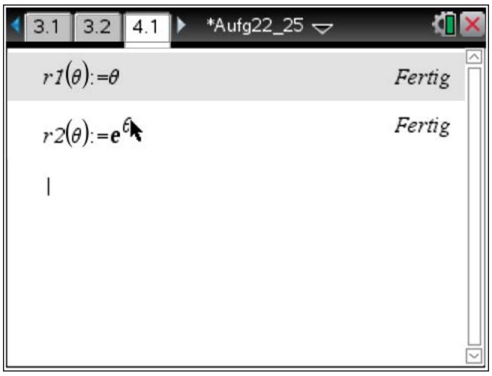
Da sieht man deutlich, dass asymptotisches Verhalten nicht an eine Gerade gebunden sein muss. (Siehe auch Aufgabe 14 d.) Ein weiteres Beispiel ist die Funktion $f(x) = x^2 + \frac{1}{x^2}$. Hier tritt eine Parabel als Asymptote auf.

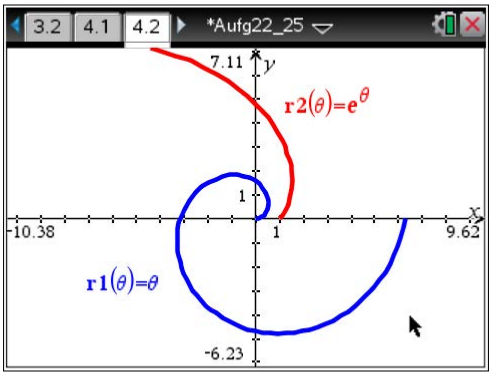
Aufgabe 25 *Verschiedene Spiralen*

Untersuche die Graphen von $r = \theta$ und $r = e^{\theta}$. Was haben sie gemeinsam und worin unterscheiden sie sich? Untersuche auch die Graphen von $r = \theta^n$. Suche im Internet nach weiteren Arten von Spiralen[*]. Warum sind für Kurven dieser Art Polarkoordinaten besser geeignet? Welche Arten von Spiralen kannst du unterscheiden? Was lässt sich über das asymptotische Verhalten von Spiralen sagen?

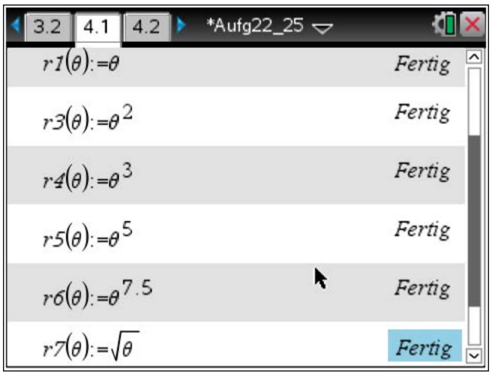
Lösung 25 Für die beiden erstgenannten Spiralen lässt sich sagen, dass die erste im Koordinatenursprung beginnt, ganz im Gegensatz zur zweiten. Die zweite entfernt sich auch wesentlich rascher vom Koordinatenursprung. Da für die Variable θ standardmäßig das Intervall [0, 2π] angenommen wird, endet die Spirale nach einer vollständigen Umdrehung. Das lässt sich aber natürlich ändern und anpassen.

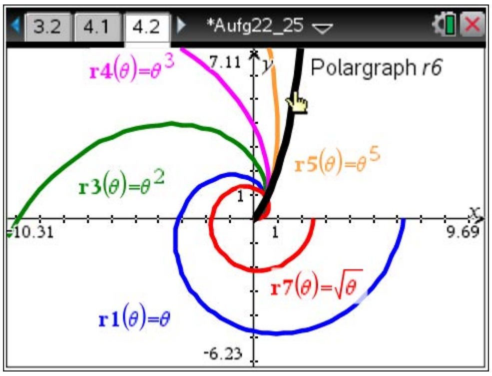
[*] z.B. in http://mathworld.wolfram.com/search/?q=spirals
oder http://www.mathe.tu-freiberg.de/~hebisch/spiralen/spiralen.html



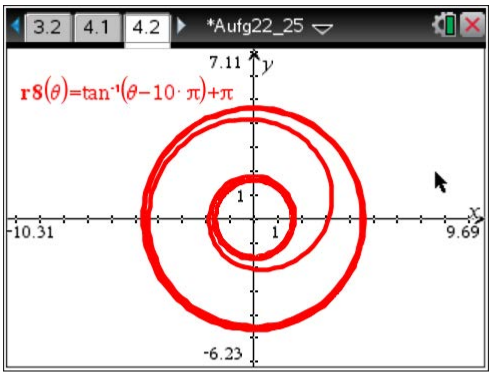


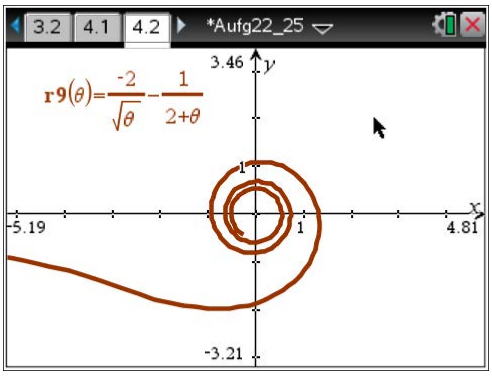
Eine ganze Familie von Spiralen der Form $r=\theta^n$:





- Eine **Auswärtsspirale** ist der Graph einer streng monoton steigenden Funktion in Polarkoordinaten, wie z.B. $f(\theta)=\theta$.
- Eine **Einwärtsspirale** ist der Graph einer streng monoton fallenden Funktion in Polarkoordinaten,wie z. B. $f(\theta)=\dfrac{1}{\sqrt{1+\theta}}$.
- Eine Spirale kann **begrenzt** oder **unbegrenzt** sein, wie z. B. $f(\theta)=\dfrac{1}{1+\theta}$ oder $f(\theta)=\sqrt{\theta}$.
- Spiralen können Gerade und/oder Kreise als Asymptoten haben. Siehe Aufgabe 24 aber auch $f(\theta)=\dfrac{1}{\sqrt{\theta}}$, $f(\theta)=\dfrac{1}{\theta}$, $f(\theta)=2+\dfrac{1}{\theta}$ oder auch $f(\theta)=\arctan(\theta-10\pi)+\pi$.





- ...

Parameterdarstellung

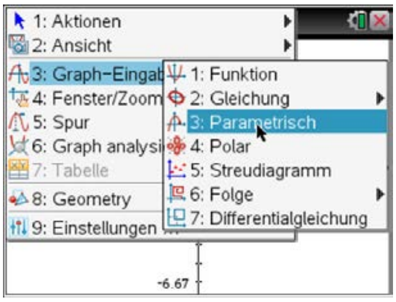
Für viele Graphen werden die Koordinaten der Kurvenpunkte in der Ebene als Funktionen eines weiteren Parameters (oft der Zeit t) gegeben: $(x(t), y(t))$. Dann wird die Kurve als ein System von Parametergleichungen festgelegt.

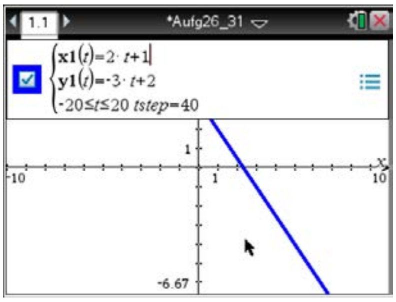
$$\begin{cases} x = f(t) \\ y = g(t) \end{cases}$$

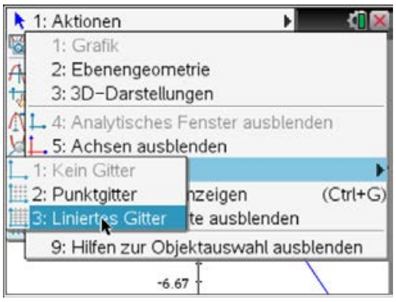
Aufgabe 26 *Eine Gerade in Parameterdarstellung*

Erzeuge den Graph von $\left(x = 2t + 1, y = -3t + 2\right)$. Es sollte eine Gerade entstehen. Versuche aus dem Graph die explizite Form $y = f(x)$ der linearen Funktion zu ermitteln. Rechnerisch gelangt man zur expliziten Form, indem man den Parameter t aus den beiden Gleichungen eliminiert, so dass nur mehr eine Gleichung in x und y übrig bleibt. Führe die entsprechende Rechnung händisch und mit TI-Nspire™ CX CAS durch und erzeuge auch den Graph der linearen Funktion $y = y(x)$.

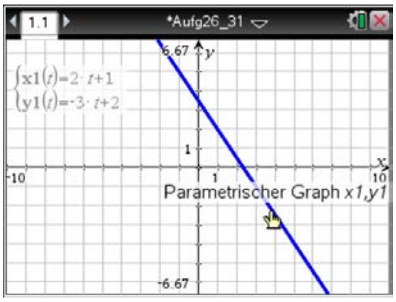
Lösung 26 Wir gelangen zur Eingabe der Parameterdarstellung über menu > 3:Graph-Eingabe/Bearbeitung > 3:Parametrisch. Da wir eine Gerade erwarten kann *tstep* ruhig größer sein. Um die Geradengleichung leichter finden zu können fügen wir Gitterlinien ein: menu > 2:Ansicht > 6:Gitter > 3:Liniertes Gitter.

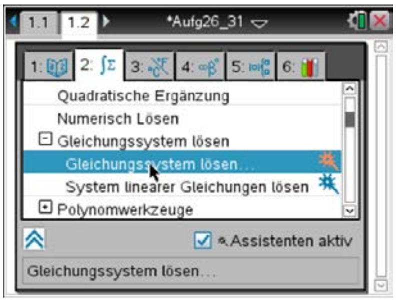


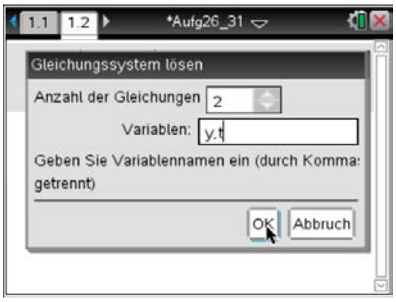




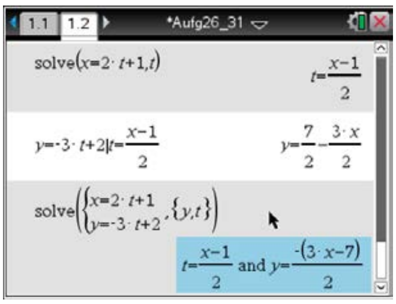
Aus dem Graph vermuten wir, dass die Geradengleichung $y = -\dfrac{3x}{2} + \dfrac{7}{2}$ lauten könnte.

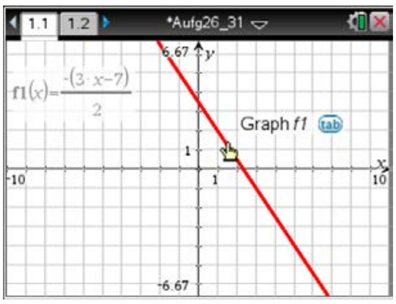






Die Rechnung kann entweder schrittweise durch Elimination des Parameters t erfolgen oder man wendet einen kleinen Trick für die Lösung des Gleichungssystems an, wie daneben gezeigt wird.

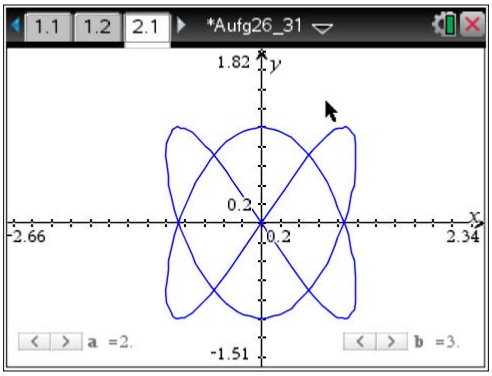


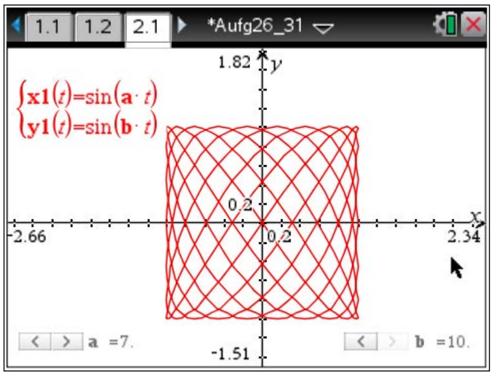


Aufgabe 27 *Lissajous Kurven*

Eine Lissajous Kurve ist gegeben durch eine Kombination von zwei aufeinander senkrecht stehenden Schwingungen: $\begin{cases} x = \sin(at) \\ y = \sin(bt) \end{cases}$. Untersuche diese Kurven. Untersuche im Besonderen wie die Gestalt der Graphen vom Verhältnis a:b abhängt. Führe Schieberegler für die Parameter a und b ein.

Lösung 27 Installiere zuerst die Schieberegler wie schon früher gezeigt und trage dann die beiden Parametergleichungen ein. Die Schieberegler können natürlich animiert werden.





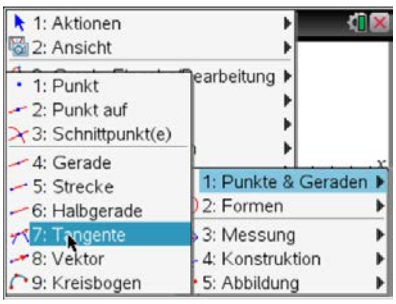
3.3 Ganz einfach mit TI-Nspire CX CAS!

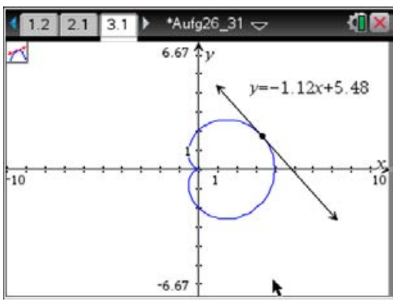
In diesem Abschnitt werden ein paar Aufgaben behandelt, die – obwohl mit TI-Nspire™ CX CAS rasch erledigt – doch eine intensivere mathematische Behandlung erfordern. Die Lösungen der Aufgaben sind auch etwas „technischer" als die vorhergehenden. Sie könnten daher von manchen als mühsam empfunden werden.

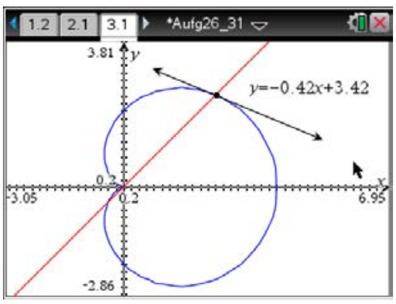
Aufgabe 28 *Tangente an die Kardioide*

Gegeben sei eine Kurve in Polarkoordinaten, wie z.B. die Kardiode aus Aufgabe 19 (ohne Schieberegler). Wie können wir die Tangente an die Kurve für $\theta = \frac{\pi}{4}$ finden?

Lösung 28 Zeichne im Grafikfenster die Kardiode wie in Aufgabe 19. Über menu > 8:Geometry > 1:Punkte & Geraden > 7:Tangente kannst du in einem beliebigen Punkt die Kurventangente zeichnen. Der Punkt kann längs der Kurve verzogen werden. Ihre – kartesische – Gleichung erhältst du dann über menu > 1:Aktionen > 8:Koordinaten/Gleichungen. Es ist aber nicht möglich, den Kurvenpunkt für $\theta = \frac{\pi}{4}$ genau anzugeben.

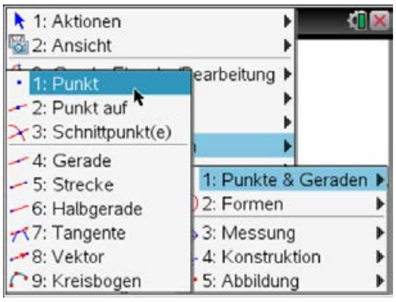


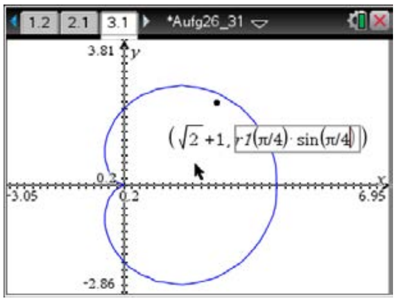




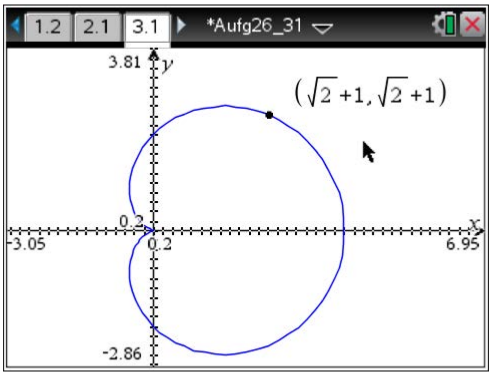
Du kannst dir mit einem Trick behelfen: Zeichne die Gerade $y = \tan(\frac{\pi}{4})x = x$ und verziehe den Punkt mitsamt Tangente möglichst genau in diesen Punkt.

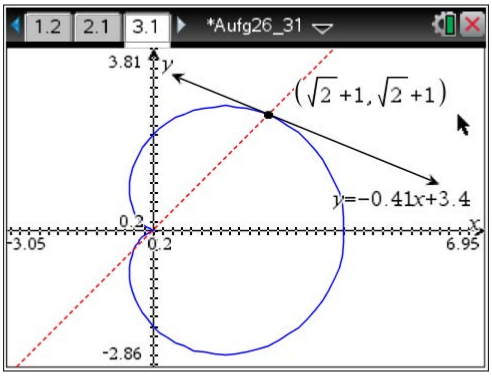
Es gibt aber noch eine Möglichkeit: zeichne zuerst wieder die Kardiode und lege dann über das Geometry –Menü einen beliebigen Punkt in der Ebene fest.





Überschreibe die Koordinaten des Punkts nach den Transformationsformeln auf Seite 23 mit ($r1(\theta)$*$\cos(\theta)$, $r1(\theta)$*$\sin(\theta)$). Damit erhältst du genau den gewünschten Punkt. Nun rufst du wie vorhin das Tangentenwerkzeug auf und erzeugst in diesem Punkt die Tangente. Ich habe dann noch die Gerade $y = x$ hinzugefügt.





Das hat aber noch immer den Makel, dass wir hier wohl den Kurvenpunkt mit seinen exakten kartesischen Koordinaten sehen, die Tangentengleichung bleibt aber numerisch – doch wiederum nur näherungsweise. Mit zusätzlichen mathematischen Kenntnissen wollen wir nun die exakte Tangentengleichung ermitteln.

Dazu greifen wir vorerst wieder auf die Transformationsformeln zurück:

$$\begin{cases} x = r\cos\theta = 2(1+\cos\theta)\cos\theta \\ y = r\sin\theta = 2(1+\cos\theta)\sin\theta \end{cases}$$

Die kartesischen Koordinaten des Berührungspunktes sind dann

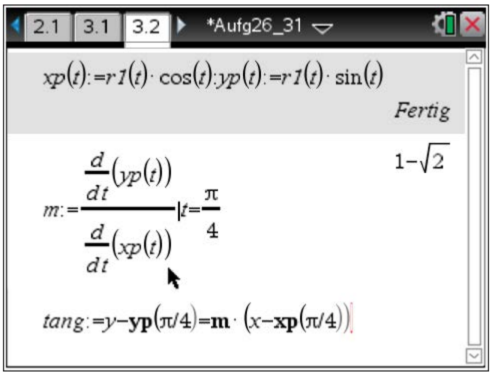
$$\begin{cases} x_0 = 2(1+\cos\frac{\pi}{4})\cos\frac{\pi}{4} = 2(1+\frac{\sqrt{2}}{2})\frac{\sqrt{2}}{2} = 1+\sqrt{2} \\ y = 2(1+\cos\frac{\pi}{4})\sin\frac{\pi}{4} = 2(1+\frac{\sqrt{2}}{2})\frac{\sqrt{2}}{2} = 1+\sqrt{2} \end{cases}$$

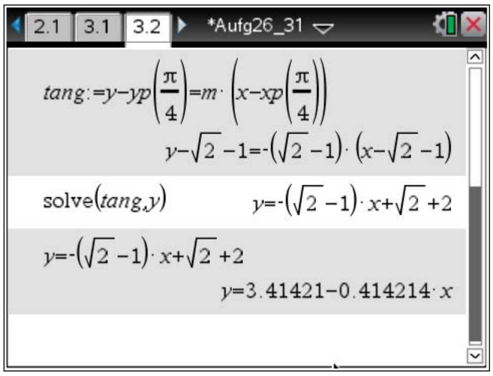
Für die Gleichung der Tangente brauchen wir noch deren Steigung und das ist der Differentialquotient

$$k = \frac{\mathrm{d}y}{\mathrm{d}x} = \frac{\frac{\mathrm{d}y}{\mathrm{d}\theta}}{\frac{\mathrm{d}x}{\mathrm{d}\theta}} = \frac{2\cos\theta + 2\cos^2\theta - 2\sin^2\theta}{-2\sin\theta - 4\cos\theta\sin\theta} = 1-\sqrt{2}\,.$$

Damit ergibt sich schlussendlich die Tangentengleichung mit $y = (1-\sqrt{2})(x-1-\sqrt{2})+1+\sqrt{2}$.

Für diese mühsamen Manipulationen mit Winkelfunktionen und Wurzeln ist das CAS des TI-Nspire™ CX CAS wie geschaffen.



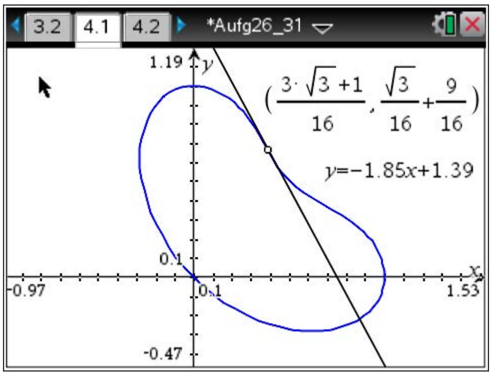


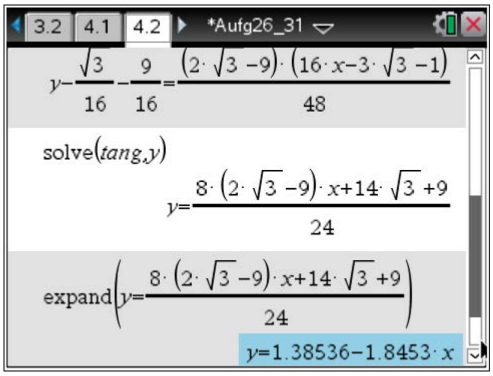
Wir müssen erkennen, dass die rein zeichnerische Lösung gar nicht so übel war.

Aufgabe 29 *Tangente an eine „Nierenkurve"*

Eine Kurve ist in Polarkoordinaten gegeben mit $r = \sin^3(\theta) + \cos^3(\theta)$. Zeichne den Graph mit der Tangente für $\theta = \frac{\pi}{3}$.

Lösung 29 Gehe vor wie in Aufgabe 28. Führe die Rechnung mit Unterstützung des CAS durch.





Aufgabe 30 *Tangenten für Parameterdarstellung*

Die Parameterdarstellung einer Ellipse in Ursprungslage lautet: $\begin{cases} x = a\cos t \\ y = b\sin t \end{cases}$. Bestimme die Gleichung der Tangente im allgemeinen Punkt $t = t_0$. Setze dann speziell $a = 3$, $b = 2$ und $t_0 = \frac{\pi}{4}$. Zeichne die Ellipse gemeinsam mit Tangente und Berührungspunkt.

Lösung 30 In den Aufgaben 28 und 29 hatten wir θ als Parameter, hier ist es t. Es wird wieder die Steigung k (oder m)[(*)] berechnet und dann in die Tangentengleichung eingesetzt:

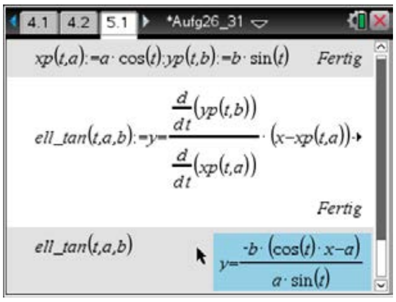
$$k = m = \frac{\mathrm{d}y}{\mathrm{d}x} = \frac{b\cos t\,\mathrm{d}t}{-a\sin t\,\mathrm{d}t} = -\frac{b}{a}\cot t$$

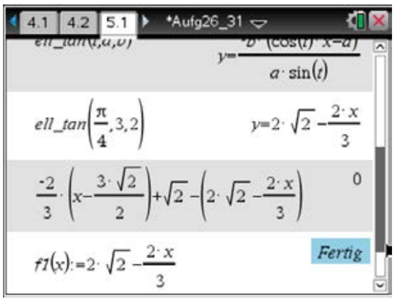
Tangentengleichung: $y = -\frac{b}{a}\cot t_0 \cdot (x - a\cos t_0) + b\sin t_0$.

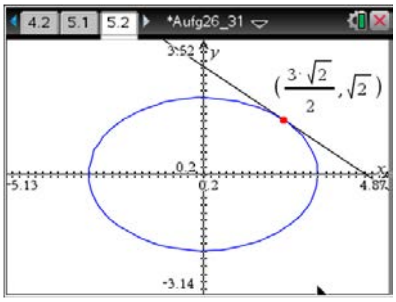
Für die speziellen Werte ergibt sich demnach: $y = -\frac{2}{3}\left(x - \frac{3\sqrt{2}}{2}\right) + \sqrt{2}$.

(*) In Österreich wird für die Steigung oft die Bezeichnung k verwendet. In anderen Ländern ist dies m.

Die Rechnung wurde hier möglichst rasch und effizient durchgeführt. *ell_tan(t,a,b)* ist eine Funktion mit der sich jede Ellipsentangente sofort bestimmen lässt. Man könnte natürlich auch mit dem $|$-Operator arbeiten. Mit der Funktion ist es aber doch eleganter. Die vorletzte Zeile im zweiten *Calculator*-Schirm zeigt die Identität der Ergebnisse von händischer und CAS-Rechnung.

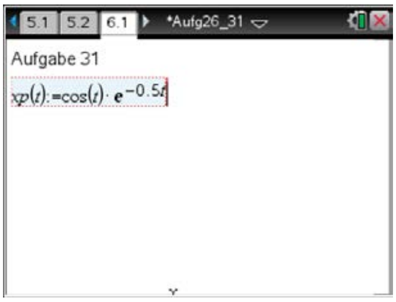


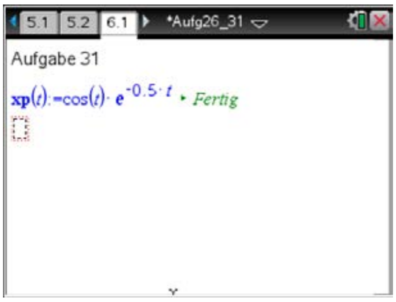


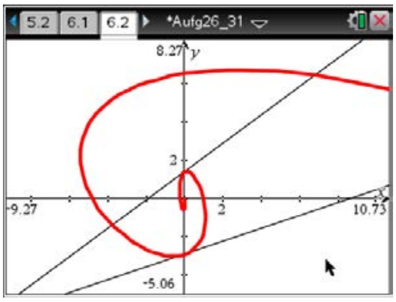


Aufgabe 31 *Tangenten für Parameterdarstellung*

Die Parameterdarstellung einer Kurve ist gegeben mit $\begin{cases} x = e^{-0.5t} \cdot \cos t \\ y = 2e^{-0.25t} \cdot \sin t \end{cases}$, $(-2\pi \le t \le 2\pi)$. Bestimme die Gleichung der Tangente an der Stelle $t = \frac{\pi}{2}$.







$\mathbf{xp}(t):=\cos(t)\cdot \mathbf{e}^{-0.5\cdot t}$ ▸ *Fertig*

$\mathbf{yp}(t):=2\cdot \sin(t)\cdot \mathbf{e}^{-0.25\cdot t}$ ▸ *Fertig*

die Tangentengleichung

$\mathbf{tang}(t):=y=\dfrac{\frac{d}{dt}(\mathbf{yp}(t))}{\frac{d}{dt}(\mathbf{xp}(t))}\cdot (x-\mathbf{xp}(t))+\mathbf{yp}(t)$ ▸ *Fertig*

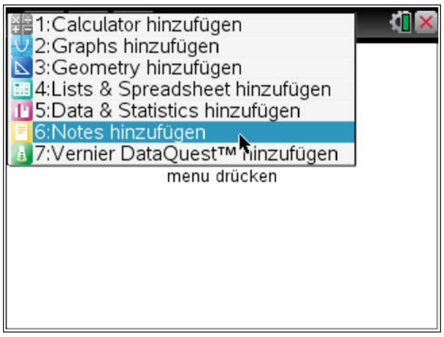
$\mathbf{tang}(t)$ ▸ $y=\dfrac{(0.778801)^{t}\cdot\left(\cos(t)\cdot\left(\sin(t)-4.\cdot \mathbf{e}^{\frac{t}{2}}\cdot x\right)+\sin(t)\cdot \mathbf{e}^{\frac{t}{2}}\cdot x+4.\right)}{\cos(t)+2.\cdot \sin(t)}$

$\mathbf{tang}\left(\frac{\pi}{2}\right)$ ▸ $y=0.740486\cdot x+1.35046$

$\mathbf{f1}(x):=\text{right}\left(\mathbf{tang}\left(\frac{\pi}{2}\right)\right):\mathbf{f2}(x):=\text{right}\left(\mathbf{tang}\left(\frac{-\pi}{2}\right)\right)$ ▸ *Fertig*

$\mathbf{x1}(t):=\mathbf{xp}(t):\mathbf{y1}(t):=\mathbf{yp}(t)$ ▸ *Fertig*

Lösung 31 Wenn man die gleiche Rechnung öfters durchführen will (z.B. Tangenten für verschiedene Kurven berechnen), dann muss man im *Calculator* immer wieder die Berechnung wiederholen. Hier bietet sich die Einführung einer *Notes*-Seite an.



Eine leere Seite wird geöffnet, auf die wir Text und mathematische Ausdrücke schreiben können.

Für einen mathematischen Ausdruck, der berechnet werden soll wird eine „Math Box" mit ctrl+M geöffnet, die durch einen roten Rand gekennzeichnet ist (wie im den Bildern auf der vorigen Seite zu erkennen ist). Der Auftrag wird hinein geschrieben und mit enter abgeschlossen.

Die kompletten *Notes* könnten für unsere Aufgabe dann so aussehen, wie auf der vorigen Seite abgebildet. Hier wurde die Computeransicht gewählt, dass alles auf einen Schirm passt. Der Graph ist dann leicht zu erstellen – inklusive einer weiteren Tangente.

Die „power" einer Notes-Seite kommt hier so richtig zum Tragen, wenn wir eine andere Funktion behandeln wollen, wie z.B.

$$\mathbf{xp}(t):=\cos(t)\cdot \mathbf{e}^{-0.5\cdot t} \blacktriangleright \textit{Fertig}$$

$$\mathbf{yp}(t):=3\cdot(\sin(t))^2\cdot \mathbf{e}^{-0.25\cdot t} \blacktriangleright \textit{Fertig}$$

die Tangentengleichung

$$\mathbf{tang}(t):=y=\frac{\frac{d}{dt}(\mathbf{yp}(t))}{\frac{d}{dt}(\mathbf{xp}(t))}\cdot(x-\mathbf{xp}(t))+\mathbf{yp}(t) \blacktriangleright \textit{Fertig}$$

$$\mathbf{tang}(t)$$

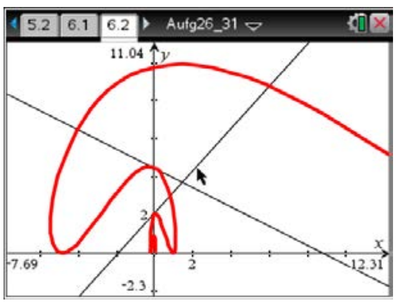
$$\blacktriangleright\ y=\frac{6.\cdot\sin(t)\cdot(0.778801)^t\cdot\left(0.25\cdot\left(\sin(t)\cdot \mathbf{e}^{\frac{t}{2}}\cdot x+4.\right)+0.25\cdot\cos(t)\cdot\left(\sin(t)-8.\cdot \mathbf{e}^{\frac{t}{2}}\cdot x\right)+(\cos(t))^2\right)}{\cos(t)+2.\cdot\sin(t)}$$

$$\mathbf{tang}\left(\frac{\pi}{2}\right) \blacktriangleright y=1.11073\cdot x+2.0257$$

$$\mathbf{f1}(x):=\text{right}\left(\mathbf{tang}\left(\frac{\pi}{2}\right)\right):\mathbf{f2}(x):=\text{right}\left(\mathbf{tang}\left(\frac{-\pi}{2}\right)\right) \blacktriangleright \textit{Fertig}$$

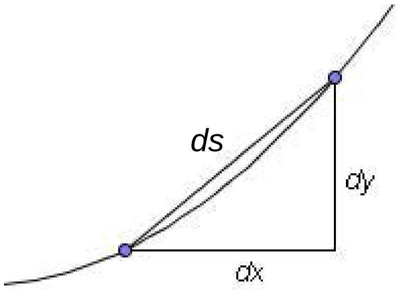
$$\mathbf{x1}(t):=\mathbf{xp}(t):\mathbf{y1}(t):=\mathbf{yp}(t) \blacktriangleright \textit{Fertig}$$

Wenn wir die Terme für xp(t) und yp(t) – hier nur für yp(t) - in den ersten beiden Zeilen überschreiben werden alle Berechnungen sofort angepasst und es ergibt sich unmittelbar die den neuen Daten entsprechende graphische Darstellung.



Aufgabe 32 *Die Bogenlänge in kartesischen Koordinaten*
Man weiß, dass die Länge eines (infinitesimal) kleinen Bogenstückchens einer Kurve durch die Länge der Hypotenuse eines rechtwinkligen Dreiecks mit den Katheten dx und dy angenähert werden kann.

Die Länge dieser Hypotenuse beträgt $\mathrm{d}s = \sqrt{(\mathrm{d}x)^2 + (\mathrm{d}y)^2} = \mathrm{d}x \cdot \sqrt{\left(\frac{\mathrm{d}y}{\mathrm{d}x}\right)^2 + 1} = \sqrt{f'(x)^2 + 1} \cdot \mathrm{d}x.$

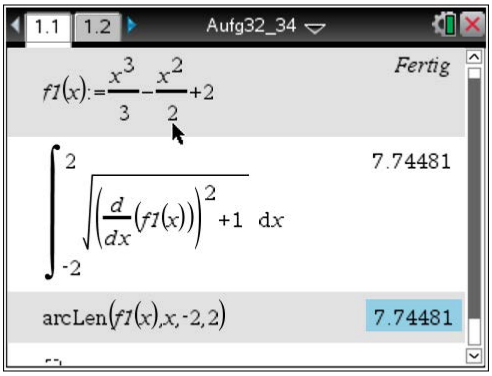


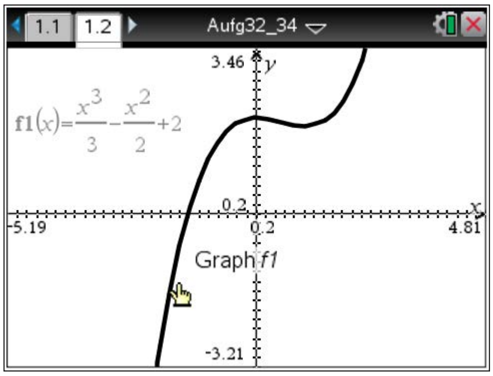
Um diese kleinen Wegstückchen zu summieren, bilden wir das bestimmte Integral vom Beginn des zu berechnenden Kurvenbogens bis zu seinem Ende:

$$s = \int_a^b \sqrt{f'(x)^2 + 1}\,\mathrm{d}x.$$

Berechne die Länge des Kurvenstücks von $f(x) = \frac{x^3}{3} - \frac{x^2}{2} + 2$ zwischen x = -2 und x = 2.

Lösung 32 Wir definieren die Funktion entweder in den *Graphs* oder im *Calculator*. Das entstehende Integral lässt sich nicht exakt berechnen.





Zur „Probe“ berechnen wir die Länge des Kurvenstücks nochmals mit der TI-Nspire™ CX CAS -Funktion arcLen().

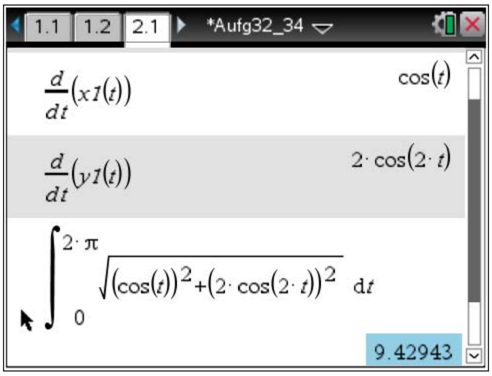
Aufgabe 33 *Die Bogenlänge für Parameterdarstellungen*

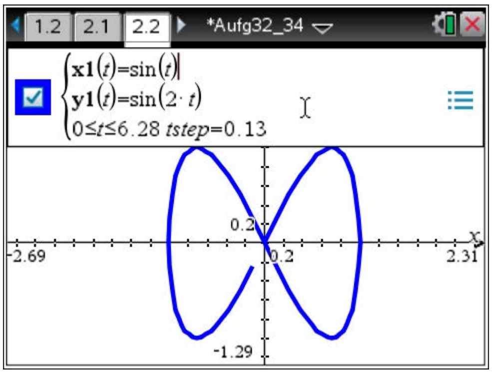
Berechne die Länge der Kurve $\begin{cases} x = \sin t \\ y = \sin 2t \end{cases}, \quad t \in [0, 2\pi[.$

Wende dieses Verfahren auch dazu an, den Umfang einer Ellipse zu berechnen.

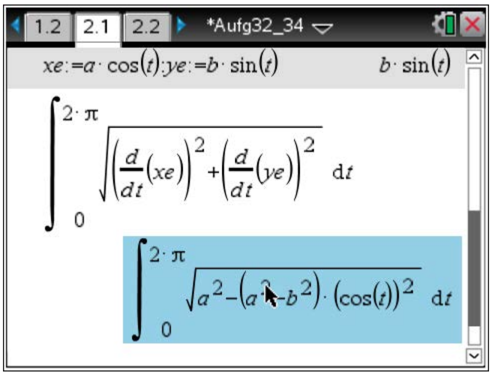
Lösung 33 Wir finden $\frac{\mathrm{d}x}{\mathrm{d}t} = \cos t$ und $\frac{\mathrm{d}y}{\mathrm{d}t} = 2\cos 2t,$ und weiters $\mathrm{d}s = \sqrt{(\cos t \cdot \mathrm{d}t)^2 + (2\cos 2t \cdot \mathrm{d}t)^2} =$

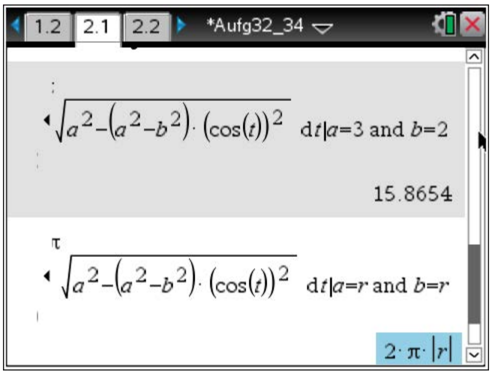
$= \sqrt{\cos^2 t + 4\cos^2 2t} \cdot \mathrm{d}t.$ Auch diese Funktion kann nicht exakt integriert werden.





Für die Ellipse und den Kreis (spezielle Ellipse mit $a = b = r$) erhalten wir die folgenden Ergebnisse:





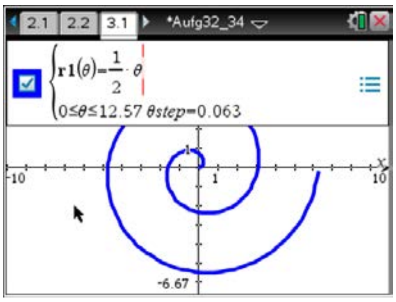
Während sich der Umfang der Ellipse nicht exakt angeben lässt („elliptisches Integral"), erhalten wir für den Kreis die erwartete Formel für den Umfang.

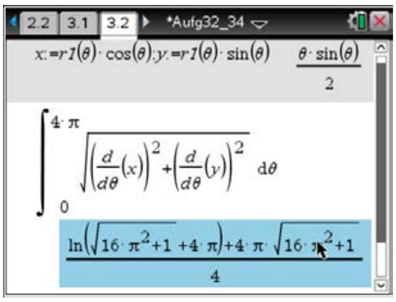
Nun fehlt noch die Berechnung der Bogenlänge in Polarkoordinaten!

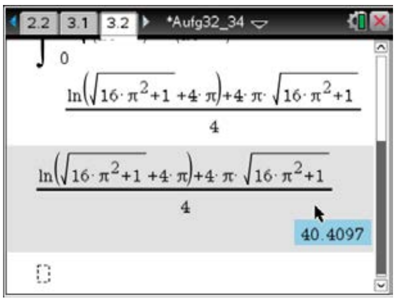
Aufgabe 34 *Die Bogenlänge in Polarkoordinaten*

Berechne die Länge von zwei Umläufen der Spirale $r = \frac{1}{2}\theta$.

Lösung 34 Nach Anwendung der Transformationsgleichungen polar → kartesisch können wir mit einer Parameterdarstellung – mit θ als Parameter – arbeiten : $\begin{cases} x = \frac{1}{2}\theta \cdot \cos\theta \\ y = \frac{1}{2}\theta \cdot \sin\theta \end{cases}$.

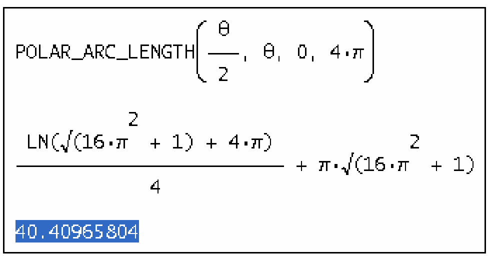






Das Ergebnis zeigt, dass dieses Integral offensichtlich exakt auswertbar ist.

Mit Hilfe eines anderen CAS (*DERIVE*) , das eine fertige Funktion für die Bogenlänge in Polarkoordinaten zur Verfügung stellt, können wir unser Ergebnis überprüfen.



4 Und noch ein wenig schwieriger ...

4.1 Wie arbeitet TI-Nspire™ CX CAS?

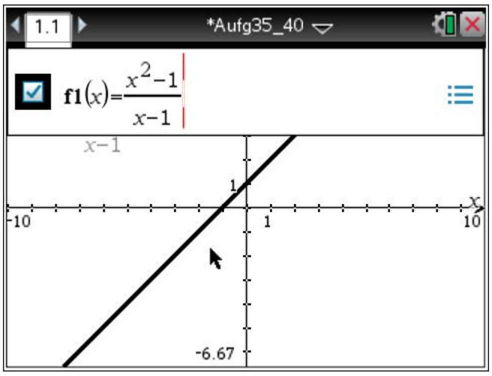
Wir beginnen mit einigen Beispielen, wie TI-Nspire™ CX CAS vorgeht, um bestimmte Aufgaben zu erledigen. Das soll uns helfen „falsche“ Antworten des Geräts zu entdecken und auch zu erklären.

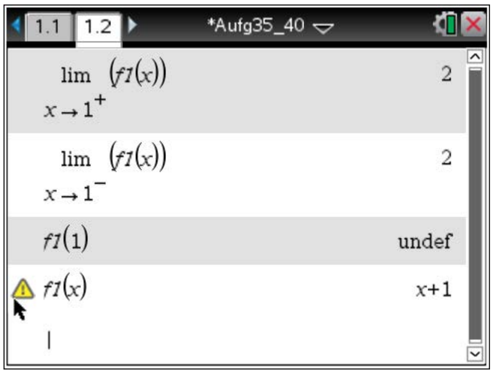
Wie erzeugt TI-Nspire™ CX CAS einen Graph?

Aufgabe 35 *Grenzwerte und Stetigkeit*

Erzeuge den Graph von $f(x) = \dfrac{x^2 - 1}{x - 1}$. Diskutiere die Stetigkeit und die Grenzwerte an den „heiklen“ Stellen. Erkläre den Fehler, den TI-Nspire™ CX CAS bei dieser Gelegenheit macht.

Lösung 35 Der Graph von $f(x)$ ist nicht korrekt, da die Unstetigkeitsstelle 1 nicht gezeigt wird. Man sieht einen stetigen Verlauf der Geraden.



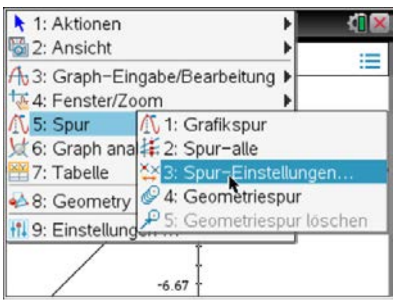


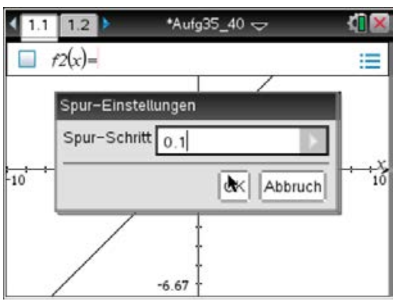
Erst wenn man den Funktionswert an der Stelle $x = 1$ ermittelt will, wird die Definitionslücke offenbar. Beachte die Ausgabe für $f1(x)$ und das „Warndreieck“!

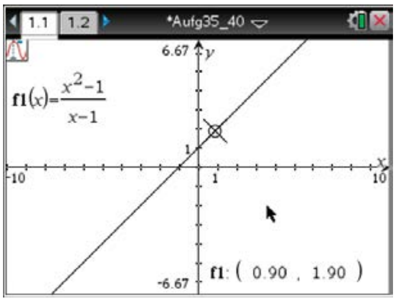
Erklärung: TI-Nspire™ CX CAS erzeugt den Graph, indem eine Anzahl von Punkten gleichmäßig im Intervall [xmin,xmax] gewählt wird. Für diese Punkte wird der Funktionswert berechnet und die entsprechen-den Punkte werden mit Liniensegmenten verbunden. Das hat zur Folge, dass die Unstetigkeit ver-borgen bleibt.

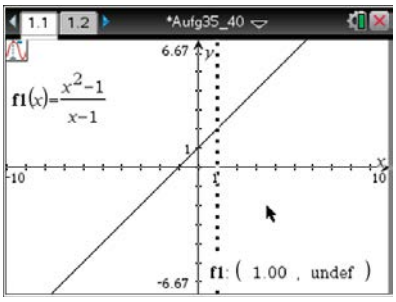
Mit einer weiteren Eigenschaft des TI-Nspire™ CX CAS kannst du aber diese Definitionslücke doch sichtbar machen: Im menu findest du die Option 5:Spur, deren Einstellungen geändert werden können. Mit dieser Spur kannst du den Verlauf des Graphen in „Spur-Schritten“ nach verfolgen.

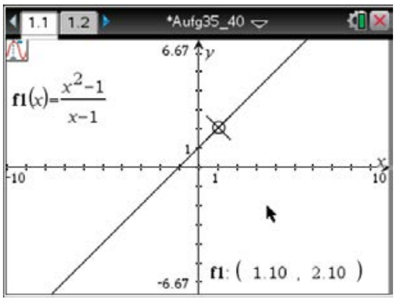
Aktiviere die Gerade und wähle menu > 5:Spur > 1:Grafikspur.









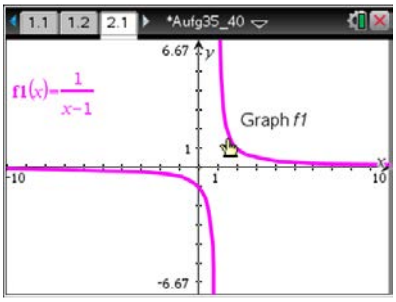


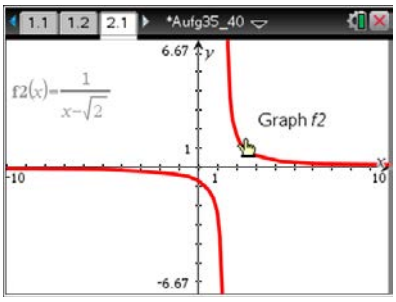
Mit dem Cursor (Pfeiltasten) gehe nun schrittweise die Gerade entlang. Unten rechts werden die Koordinaten des gerade besuchten Punkts angezeigt. Und was geschieht an der Stelle x = 1?

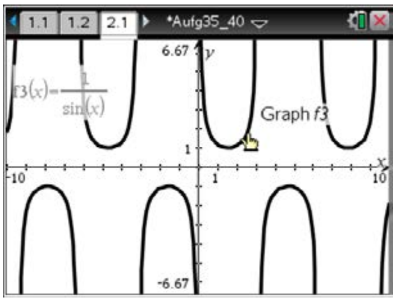
Aufgabe 36 *Asymptoten*

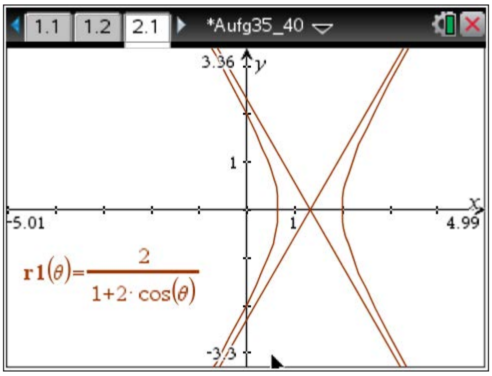
Erzeuge die Graphen von $f_1(x)=\dfrac{1}{x-1}$, $f_2(x)=\dfrac{1}{x-\sqrt{2}}$ und $f_3(x)=\dfrac{1}{\sin(x)}$. Werden die senkrechten Asymptoten gezeichnet oder nicht? Erzeuge dann auch die Graph von $r=\dfrac{2}{1+2\cos(\theta)}$ (eine Hyperbel) und $r=\dfrac{\cos\theta}{\sin\theta-2\cos\theta}$ in Polarkoordinaten. Was fällt nun auf?

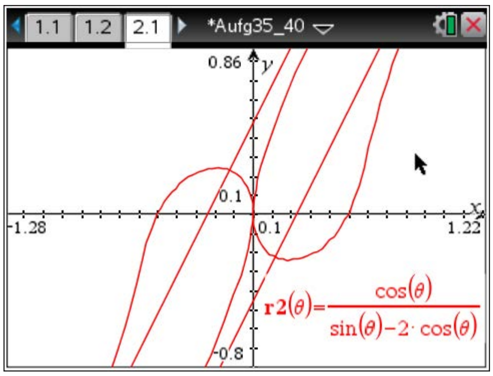
Lösung 36 Die Asymptoten werden abhängig vom gewählten Koordinatensystem einmal gezeichnet, ein anderes mal aber nicht.







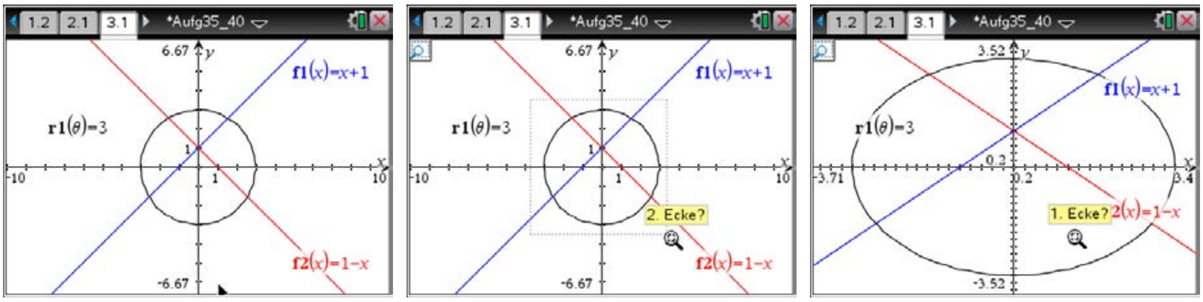




Erklärung: Die Erklärung für diesen Fehler ist sehr einfach. In kartesischen Koordinaten gibt es zum Zeichnen des Graphen einen zusätzlichen Algorithmus, der prüft, ob die Funktion sehr stark steigt oder fällt. Wenn dies der Fall ist, dann werden zwei aufeinander folgende Punkte nicht durch ein Liniensegment verbunden. Im Polarkoordinatensystem gibt es diesen Algorithmus nicht. Alle Punkte werden verbunden, daher erscheinen die Asymptoten.

Aufgabe 37 *Rechtwinkliges Koordinatensystem*
Zeichne die Graphen zweier aufeinander senkrecht stehender Geraden. Zeichne auch den Graph eines Kreises (in Polarkoordinaten). Verwende menu > 4:Fenster > 2:Zoom-Rahmen um einen Ausschnitt zu vergrößern. Was bemerkst du nun?

Lösung 37 Wir wählen die beiden Geraden $y = x + 1$ und $y = –x + 1$ und erhalten dann die folgenden Graphen:



Im ersten Fall stehen die beiden Geraden normal zu einander und der Kreis erscheint wie erwartet als Kreis senkrecht. Nach Vergrößerung des gewählten Ausschnitts schneiden die beiden Geraden einander nicht mehr unter einem rechten Winkel und der Kreis wurde zur Ellipse deformiert.

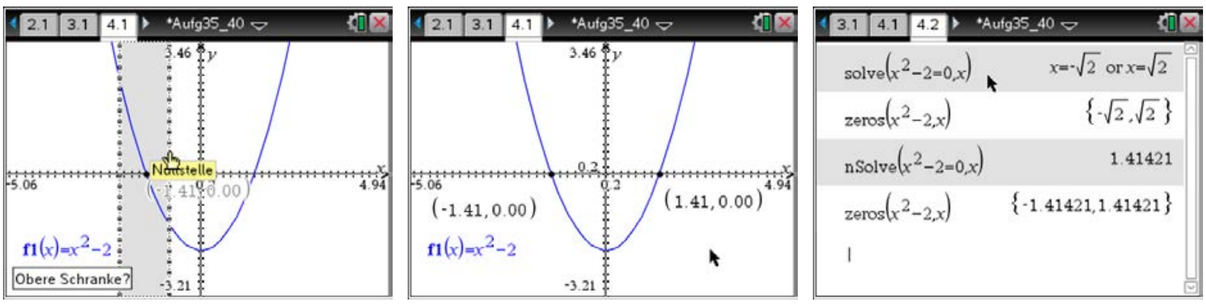
Erklärung: Der grundsätzliche Unterschied besteht darin, dass bei der Fenstereinstellung 5:Zoom-Standard ein Koordinatensystem verwendet wird, das winkeltreu ist, d.h., rechte Winkel erscheinen auch als rechte Winkel, Kreise erscheinen als Kreise etc., wobei wohl die Koordinatenachsen senkrecht zu einander stehen, auf den Achsen aber unterschiedliche Skalierungen angewendet werden. Damit kommt es zu einer Verzerrung. Aus der analytischen Geometrie wissen wir, dass alle Begriffe, die auf dem inneren Produkt basieren (wie z.B. Winkel, normale Lage zu einander, Normalabstand) ein System mit gleichen Skalierungen auf beiden Achsen benötigen.

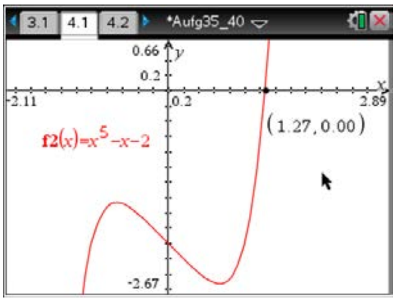
Nullstellen, Schnittpunkte und Extremwerte

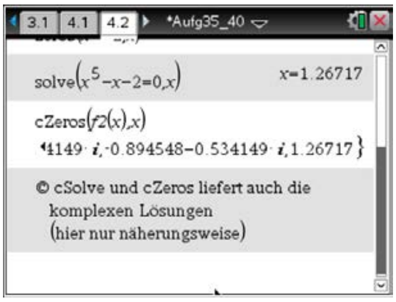
Aufgabe 38 *Nullstellen*

Bestimme im Grafikfenster die Nullstellen von $f(x) = x^2 – 2$ und $g(x) = x^5 – x – 2$. Suche diese Nullstellen anschließend auch im Rechenfenster.

Lösung 38 Im Grafikfenster werden Berechnungen immer nur numerisch durchgeführt, d.h., man erhält jeweils nur einen Näherungswert, während im Rechenfenster das CAS verwendet wird. Hier erhält man – wenn dies möglich ist – die exakten Werte. Die Nullstellen einer Polynomfunktion vom Grad 5 oder höher lassen sich auch von einem CAS i. A. nicht genau angeben.





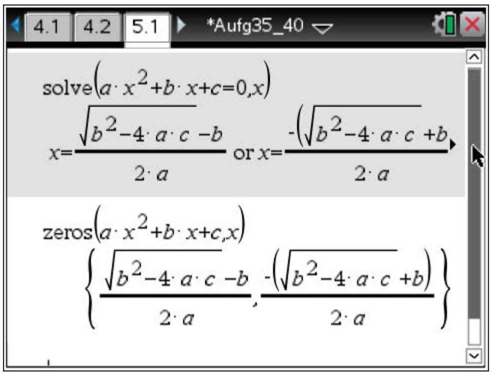


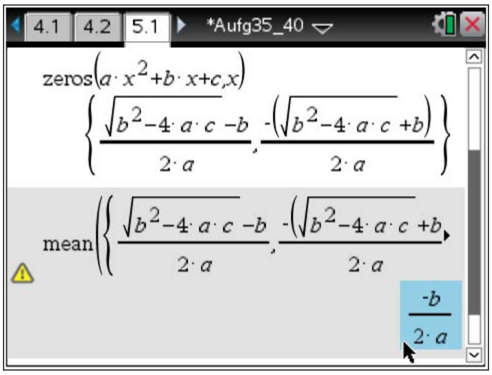
Erklärung: Die Erklärung dafür ist hier nicht technischer Natur, sie liegt innerhalb der Mathematik. Erstens wird im Grafikfenster ein anderer Algorithmus verwendet als im Rechenfenster (CAS). Dieser Algorithmus wird später im Programmierteil näher erläutert. Im Rechenfenster wird eine algebraische Methode angewendet. Allerdings gibt es einen wichtigen Lehrsatz der Mathematik, in dem bewiesen wird, dass Polynomgleichungen mit einem Grad höher als vier im Allgemeinen nicht exakt lösbar sind. Da kann auch das CAS nur numerische Näherungswerte angeben.

Aufgabe 39 *Schnittpunkte und Extremwerte*

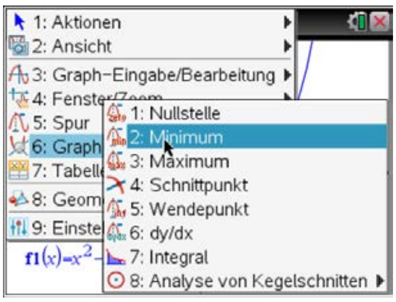
Du weißt nun, wie TI-Nspire™ CX CAS Nullstellen findet. Gib selbst zwei gute Beispiele zur Bestimmung von Extremwerten und Schnittpunkten an.

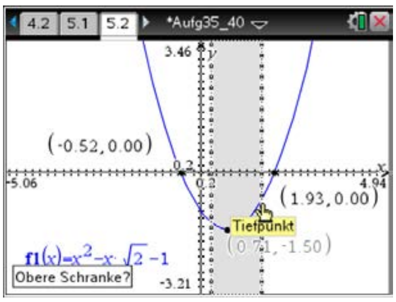
Lösung 39 Betrachte den Scheitel der Parabel $y = a\,x^2 + b\,x + c$. Der Scheitel liegt genau zwischen den beiden Nullstellen, sofern es solche gibt. Wir zeigen an einem konkreten Beispiel wie der Scheitel zu finden ist.

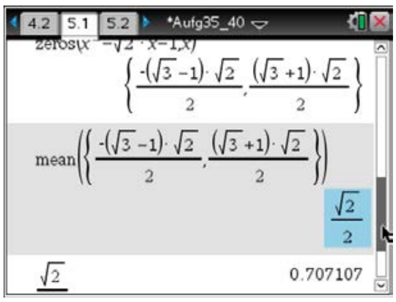




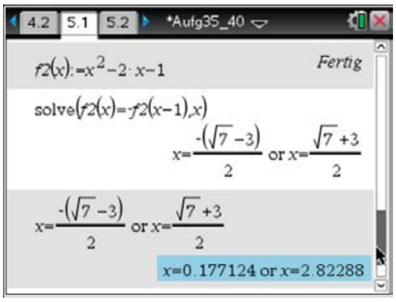
Mit mean(Liste) berechnen wir den Mittelwert der Elemente der Lösungsmenge. Wir erkennen aber auch, dass diese Formel auch für den Fall gilt, dass der Ausdruck unter der Wurzel – die Diskriminante – negativ ist und es dann keine reellen Nullstellen geben kann. Nun die grafische Lösung für die Parabel $y = x^2 - x \cdot \sqrt{2} - 1$. Beachte den Befehl zeros (engl. = Nullstellen) zur Berechnung der Nullstellen, die als Werteliste ausgegeben werden.

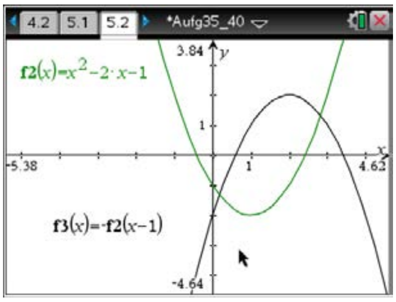


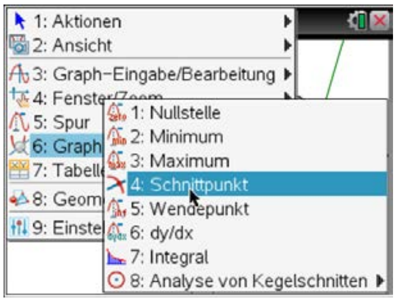


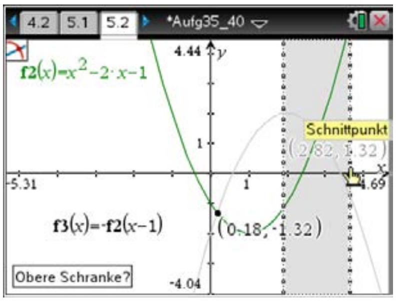


Für den Schnittpunkt zweier Kurven wählen wir die beiden Parabeln $f(x) = x^2 - 2x - 1$ und $g(x) = -f(x - 1)$. Das kannst du natürlich auch ohne Rechner durchführen.





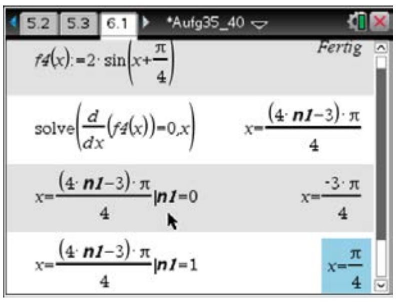


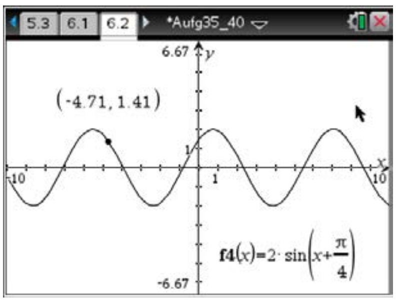


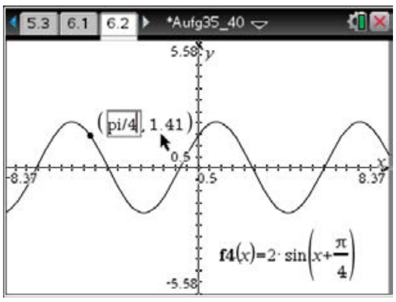
Aufgabe 40 *Minima und Maxima*

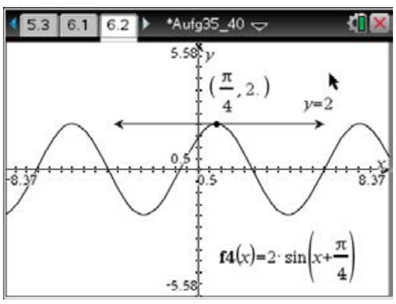
Berechne ein Maximum der Funktion $f(x) = 2\sin\left(x + \frac{\pi}{4}\right)$. Zeichne den Graph gemeinsam mit der Tangente in diesem Punkt.

Lösung 40 Wir beginnen mit der Berechnung des Maximums und wählen aus den unendlich vielen Lösungen jene mit dem Parameter ***n1*** = 0. Dann zeichnen wir den Graph von *f*(*x*) mit einem beliebigen Punkt auf dem Graph. Wir klicken auf die 1. Koordinate des Kurvenpunkts und überschreiben sie mit pi/4. Nun haben wir den gewünschten Hochpunkt gezeichnet. Über das Menu gelangen wir zum Zeichnen der Tangente in diesem Punkt.









4.2 Programmieren in TIBasic ist einfach

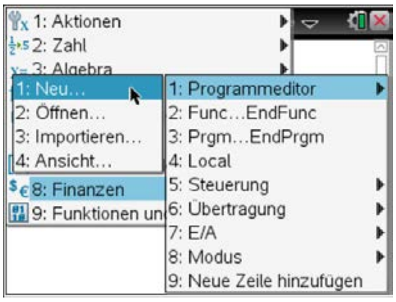
In diesem Kapitel wollen wir einige bekannte Algorithmen für TI-Nspire™ CX CAS implementieren. Das verlangt ein wenig Programmierarbeit.

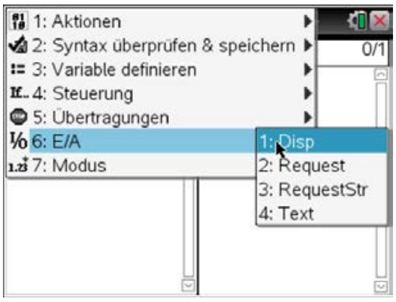
Die Programmiersprache des TI-Nspire™ CX CAS ist ein BASIC-Dialekt. BASIC ist jene Sprache, mit der Bill Gates (Microsoft) bekannt geworden ist. BASIC (Beginners All-Purpose Symbolic Instruction Code) wurde entworfen, dass auch breite PC-Nutzerkreise kleine Programme selbst schreiben können. TIBasic bleibt dieser Philosophie treu: es ist keine Programmiererfahrung nötig, um hier Programme zu schreiben. Ein wenig Beharrlichkeit und Selbstvertrauen sind allerdings erforderlich.

Hier wird mit der Programmierumgebung von TI-Nspire™ CX CAS Version 3.6 gearbeitet. Nachfolgende Versionen werden möglicherweise umfassender sein.

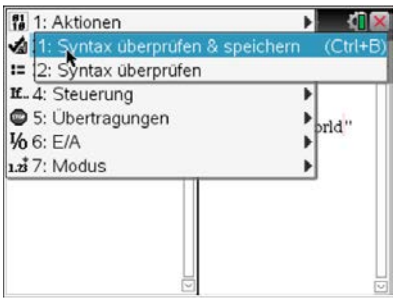
Um ein Programm zu schreiben muss vorerst ein Rechenfenster geöffnet werden. Über menu > 9:Funktionen und Programme > 1:Programmeditor > 1:Neu ... wirst du zur Angabe des Namens deines neuen Programms (oder der neuen Funktion) aufgefordert. Auf den Unterschied zwischen einem Programm und einer Funktion werden wir im Abschnitt 5.7 kurz eingehen.

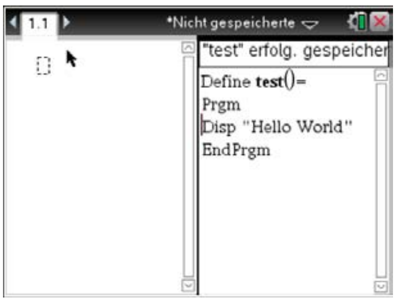
Wir wollen unser erstes Programm „test“ nennen. Damit wird der Schirm zweigeteilt. Links bleibt der bereits bekannte *Calculator* und rechts der Programmeditor. Hier haben wir Raum um das Programm zu schreiben. Mach die Editorhälfte aktiv und rufe das menu auf, dann wirst du hier alle Kommandos, die zum Programmieren verwendet werden können, finden.



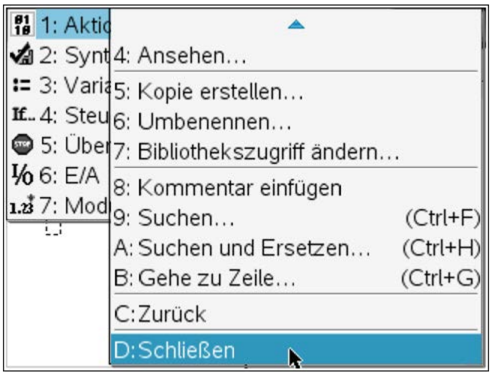


Unter 6:E/A werden alle verfügbaren Ein- und Ausgaberoutinen angezeigt. in fast allen Einführungen zu den Programmiersprachen ist das allererste Programm das „Hello World“-Programm:

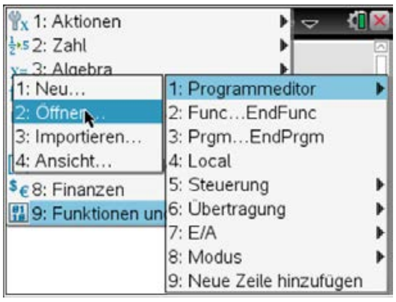




Es wird nur der Text „Hello World“ ausgegeben. Dann wollen wir das auch so halten. Der Stern beim Programmnamen im linken Bild zeigt, dass das Programm noch nicht gespeichert wurde. Über menu > 2:Syntax überprüfen & speichern (= ctrl + B) wird das vorliegende Programm auf syntaktische Fehler (aber auf keine logischen oder sonstigen Programmierfehler) geprüft und gespeichert. Erst jetzt kann die aktuelle Version auf der Rechenseite durchgeführt werden. Es wird aufgerufen mit test(), da keine weiteren Programmparameter vorliegen:

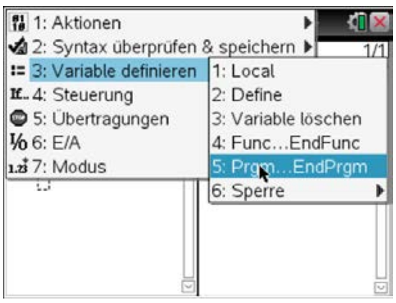


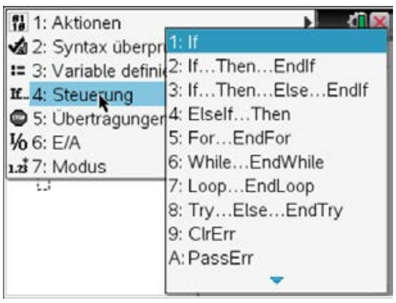
Nach erfolgreichem Test kann der Editor geschlossen werden und man findet sich wieder im Rechenfenster. Der Programmeditor kann bei jeder Gelegenheit aufgerufen werden, um das erstellte Programm in den Editor zu laden und eventuell zu verändern (verbessern, ergänzen, u.s.w.)

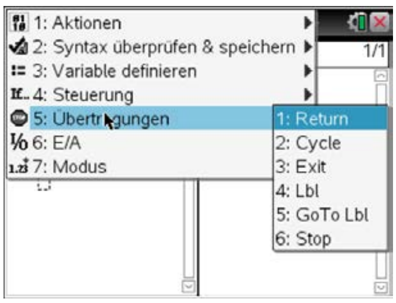


Es ist noch anzumerken, dass dieses Programm nur innerhalb des Dokuments in dem es gespeichert wurde verwendet werden kann. Programme, auf die von anderen Dokumenten auch zugegriffen werden kann, sind in *Bibliotheken* („*libraries*“) zusammengefasst. Deren Behandlung würde aber den Rahmen dieser Einführung sprengen. (Mehr übers Programmieren und auch über die Bibliotheken findet man in [*].)

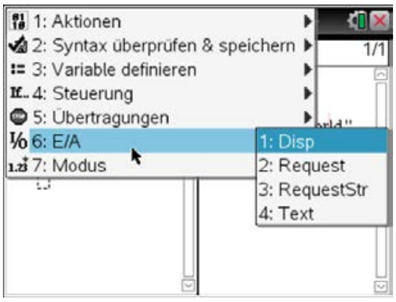
Wie schon früher gesagt, findet man die Programmierbefehle über das menu:

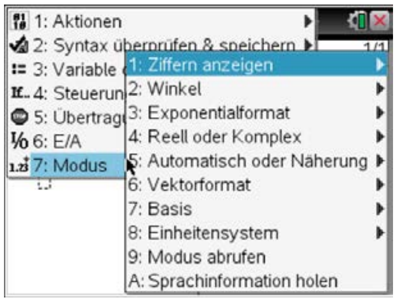






[*] Josef Böhm, *Programmieren mit TI-Nspire CAS*, bk teachware SR-60
http://www.acdca.ac.at/material/nspire/nspire_progkurz.htm
http://www.acdca.ac.at/material/nspire/nspire_trigonometrie.htm





Wir werden in dieser kurzen Anleitung nicht alle Befehle kennen lernen. Natürlich steht uns innerhalb eines Programms oder einer Funktion auch die volle Kraft des CAS zur Verfügung. Das konnte man mit dem „Ur-BASIC“ und vielen Nachfolgern nicht. Da musste man z.B. für das Differenzieren und Integrieren, ja selbst für das Lösen von einfachen Gleichungen z.T. recht aufwändige Programme schreiben.

Aufgabe 41 *Funktionsgraph*

Verwende eine For-EndFor-Schleife um den Graph einer Funktion (z.B. der Sinusfunktion) zu erzeugen.

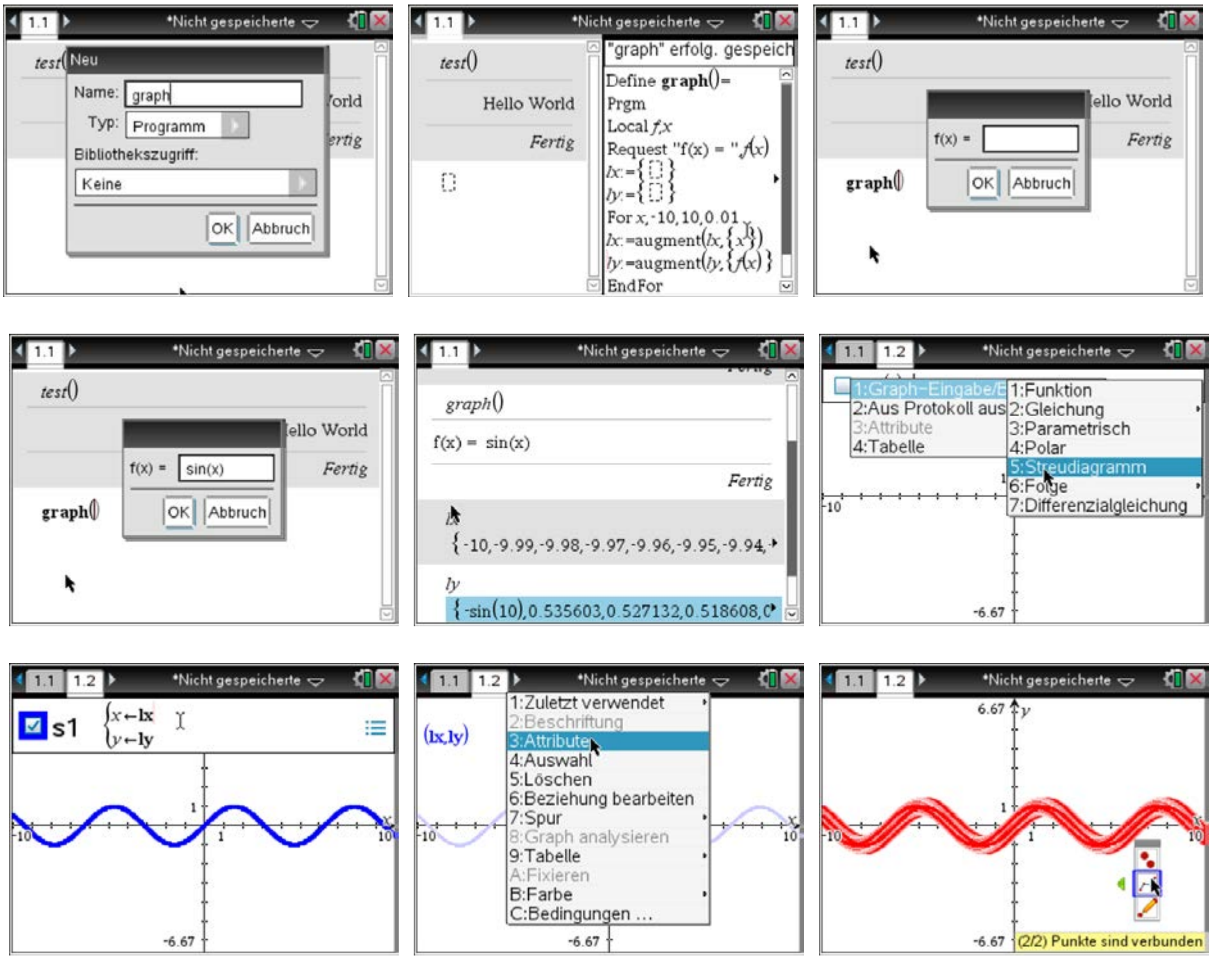
Lösung 41 Das geht mit TI-Nspire™ CX CAS leider (noch?) nicht so einfach, da die Programmiersprache noch keine Grafikkommandos bereit stellt wie z.B. der TI-92 und Voyage 200. Es ist zu hoffen, dass diese Lücke in späteren Versionen gefüllt wird. Wir können uns mit einem Trick behelfen, indem wir über erst zu erzeugende Listen von Argumenten und entsprechenden Funktionswerten das zugehörige Streudiagramm zeichnen.

Es muss deutlich zwischen *lokalen* und *globalen Variablen* unterschieden werden. Die Listen *lx* und *ly* werden nicht als Local definiert, d.h., sie sind dann auch außerhalb des Programms verfügbar.

```
1   Define graph()=
2   Prgm
3   :Local f,x,y
4   :Request "f(x) = ",f(x)
5   :lx:={}
6   :ly:={}
7   :For x,-10,10,0.01
8   :  lx:=augment(lx,{x})
9   :  ly:=augment(ly,{f(x)})
10  :EndFor
11  :EndPrgm
```

In den Programmzeilen 5 und 6 werden leere Listen *lx* und *ly* erzeugt. An diese Listen werden in der Schleife (Zeilen 8 und 9) immer wieder die nächsten Elemente angehängt und die alten Listen werden durch die neuen – nun verlängerten – Listen ersetzt. Die Koordinaten der erzeugten Kurvenpunkte sind zum Schluss den beiden Listen gespeichert.

In einem *Graphs*-fenster zeichnen wir abschließend das Streudiagramm.

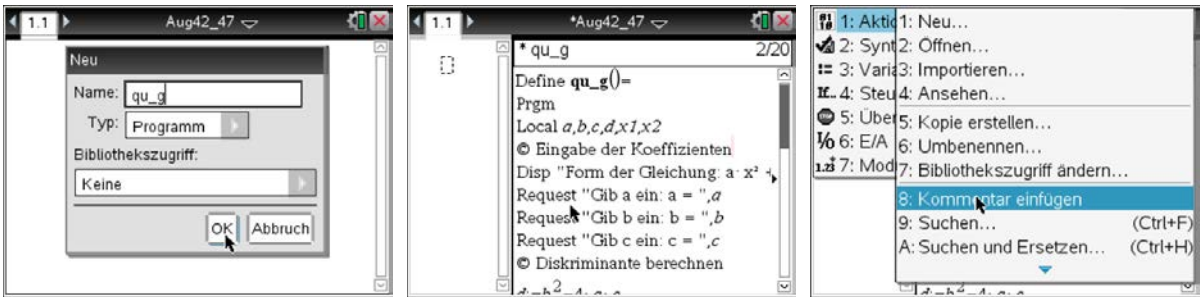


Über die 3:Attribute und B:Farbe lässt sich der Graph nach eigenen Vorstellungen „verschönern".

Beispielprogramme

Aufgabe 42 *Quadratische Gleichungen*
Schreibe ein Programm, das eine quadratische Gleichung löst.

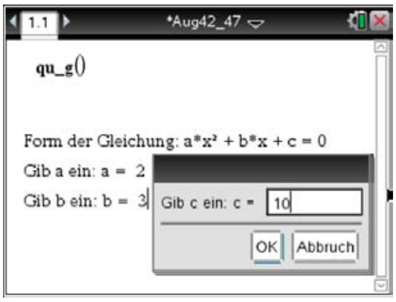
Lösung 42 Das Programm sieht aus wie folgt.

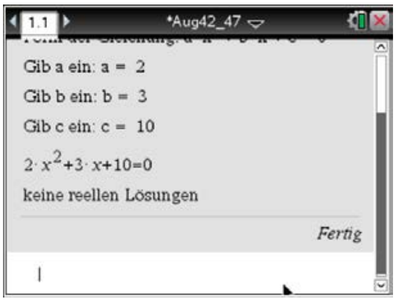


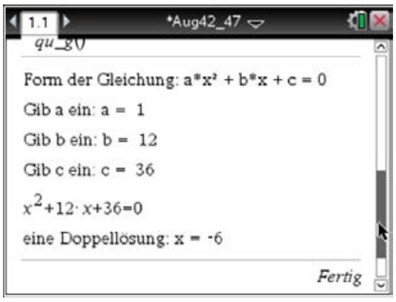
Einige Dinge sind beachtenswert: Füge an geeigneten Stellen Kommentare (©) ein. Sie erklären das Programm, dann kann jeder nachvollziehen, wie das Programm abläuft.

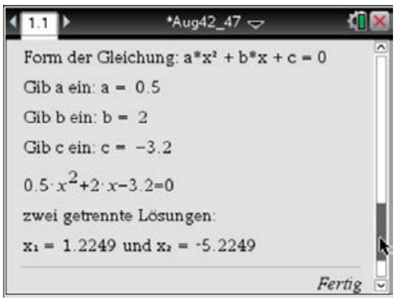
Die If-Anweisung kann aus nur einer Zeile bestehen, wenn im Fall des Eintretens der Bedingung nur eine weitere Anweisung auszuführen ist, anderenfalls muss man über If <Bedingung> Then <Anweisungsblock> EndIf einen Anweisungsblock (= mehrere Befehle) einfügen.

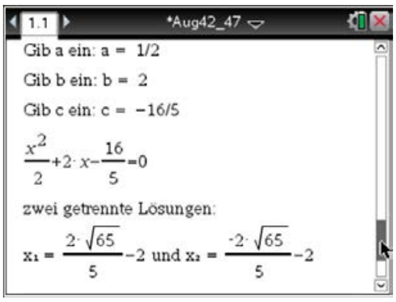
```
1   Define qu_g()=
2   Prgm
3   :Local a,b,c,d,x1,x2
4   :© Eingabe der Koeffizienten
5   :Disp "Form der Gleichung: a*x² + b*x + c = 0"
6   :Request "Gib a ein: a = ",a
7   :Request "Gib b ein: b = ",b
8   :Request "Gib c ein: c = ",c
9   :© Diskriminante berechnen
10  :d:=b^2−4*a*c
11  :Disp a*x^2+b*x+c=0
12  :© Fallunterscheidung
13  :If d<0: Disp "keine reellen Lösungen"
14  :If d=0: Disp "eine Doppellösung: x = ",-b/(2*a)
15  :If d>0 Then
16  :   x1:=(-b+√(d))/(2*a)
17  :   x2:=(-b−√(d))/(2*a)
18  :   Disp "zwei getrennte Lösungen:"
19  :   Disp "x1 = ",x1," und x2 = ",x2
20  :EndIf
21  :EndPrgm
```











Die unterschiedlichen If-Konstrukte kannst du in den Zeilen 13 und 14 bzw. 15 bis 20 erkennen. Die letzte Gleichung wurde einmal mit Dezimalzahlen und dann nochmals mit Brüchen als Koeffizienten eingegeben. Wenn die Dokumenteinstellungen für den Berechnungsmodus auf „Auto“ stehen, wird bei der Eingabe von zumindest einem Wert in Dezimalform immer die Dezimalausgabe mit der voreingestellten Anzahl von Dezimalstellen angewendet.

Aufgabe 42a *Quadratische Gleichungen erweitert*
Erweitere das Programm so, dass auch die konjugiert komplexen Lösungen ausgegeben werden.

Aufgabe 43 *Näherungsweise Nullstellenberechnung durch Intervallhalbierung*

Mit welchen Algorithmen lassen sich numerisch Nullstellen ermitteln? Das erste Verfahren, das wir behandeln wollen ist die Methode der Intervallhalbierung. Dazu sind zwei Lehrsätze nötig:

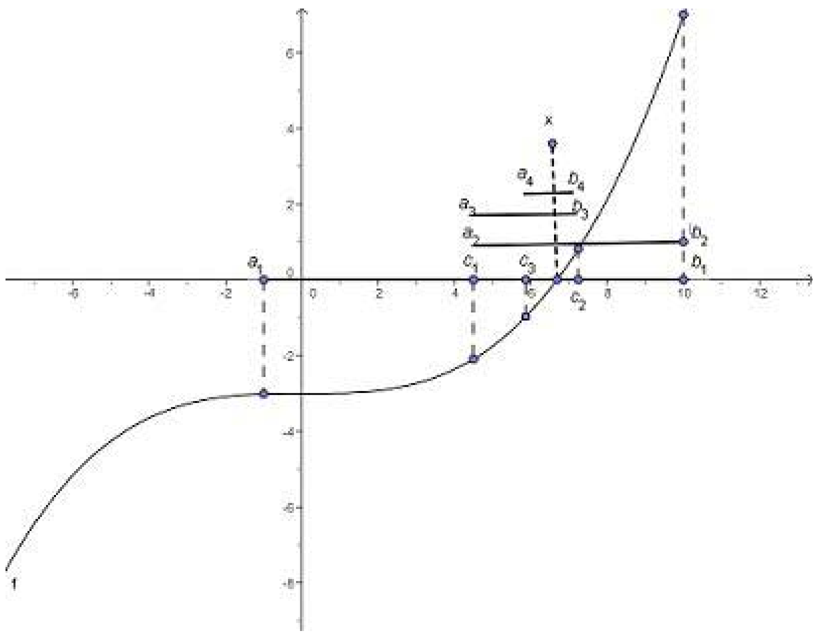
Satz 1 (Bolzano)
Es sei $f:\mathbb{R}\to\mathbb{R}$ eine auf dem Intervall $]a, b[$ stetige Funktion. Wenn $f(a) < f(b)$ (bzw. $f(b) < f(a)$), dann gibt es für jedes $c \in [f(a), f(b)]$ (bzw. $c \in [f(b), f(a)]$ ein $x \in [a,b]$ so dass $c = f(x)$.

Aus diesem Satz folgt unmittelbar der nächste

Satz 2
Es sei $f:\mathbb{R}\to\mathbb{R}$ eine auf dem Intervall $]a, b[$ stetige Funktion. Wenn $f(a)$ und $f(b)$ unterschiedliche Vorzeichen aufweisen, dann hat f eine Nullstelle in $[a, b]$.

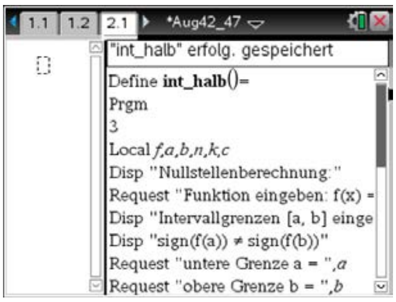
Der Nutzen dieses Satzes liegt vor allem in seinen vielen Anwendungen. Die Schüler und Schülerinnen können damit leicht zeigen, dass jede Polynomgleichung mit ungeradem Grad mindestens eine reelle Nullstelle hat (indem sie zwei Funktionswerte berechnen).

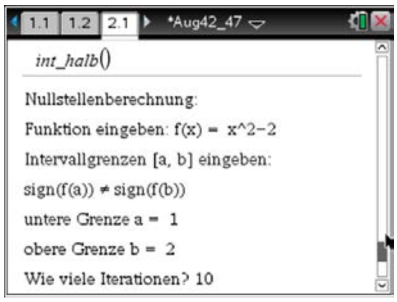
Als weitere Anwendung können wir näherungsweise Nullstellen berechnen, indem man diesen Satz iterativ einsetzt. Es sei $f:\mathbb{R}\to\mathbb{R}$ eine auf dem Intervall $]a_1, b_1[$ stetige Funktion. Wenn nun $f(a_1)$ und $f(b_1)$ verschiedene Vorzeichen haben, dann schneidet der Graph die x-Achse im Intervall $[a_1, b_1]$. Wir berechnen nun den Mittelpunkt $c_1 = \frac{a_1+b_1}{2}$. Wenn $f(c_1) = 0$ haben wir die Nullstelle bereits gefunden. Im Allgemeinen ist das aber nicht der Fall: Haben nun $f(c_1)$ und $f(a_1)$ verschiedene Vorzeichen, dann muss die Nullstelle in $[a_1, c_1]$ liegen (wir setzen $a_2 = a_1$ und $b_2 = c_1$); anderenfalls sind die Vorzeichen von $f(c_1)$ und $f(b_1)$ verschieden und wir setzen $a_2 = c_1$ und $b_2 = b_1$. Die Nullstelle liegt nun in $[a_2, b_2]$ und die Prozedur geht mit der Berechnung der nächsten Intervallmitte weiter. Wir erhalten somit eine Reihe von ineinander geschachtelten Intervallen, die die Nullstelle x immer besser annähern. Die Skizze verdeutlicht die Vorgangsweise.

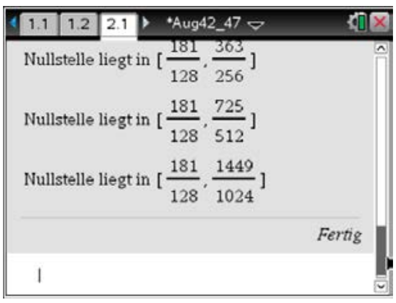


Lösung 43 Das Programm fragt zuerst um die Funktion $f(x)$ und anschließend um das Intervall $[a, b]$ in dem die Nullstelle gesucht werden soll, wobei $f(a)$ und $f(b)$ verschiedene Vorzeichen haben müssen. Dann wird noch um die Anzahl der durchzuführenden Iterationen (Intervallhalbierungen) gefragt. Im Programm müssen wir noch prüfen, ob $f(a)$ und $f(c)$ unterschiedliche Vorzeichen haben. Das geht am einfachsten, indem wir prüfen, ob $f(a) \cdot f(c) \leq 0$. Davon abhängig werden dann die Grenzen des nächsten Intervalls gesetzt.

```
1   Define int_halb()=
2   Prgm
3   :Local f,a,b,n,k,c
4   :Disp "Nullstellenberechnung:"
5   :Request "Funktion eingeben: f(x) = ",f(x)
6   :Disp "Intervallgrenzen [a, b] eingeben:"
7   :Disp "sign(f(a)) ≠ sign(f(b))"
8   :Request "untere Grenze a = ",a
9   :Request "obere Grenze b = ",b
10  :Request "Wie viele Iterationen?",n
11  :For k,1,n
12  :  c:=(a+b)/2
13  :  If f(a)*f(c)<0 Then
14  :            b:=c
15  :       Else
16  :            a:=c
17  :  EndIf
18  :Disp "Nullstelle liegt in [",a,",",b,"]"
19  :EndFor
20  :EndPrgm
```





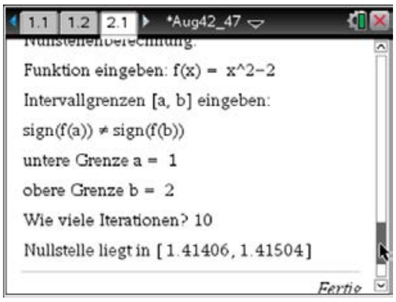


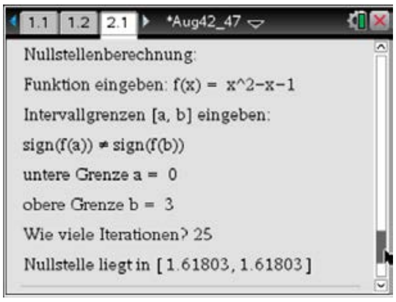
Es liegt auf der Hand, dass uns hier nur das letzte Intervall interessiert und dass wir das Ergebnis lieber als Dezimalzahl sehen würden.

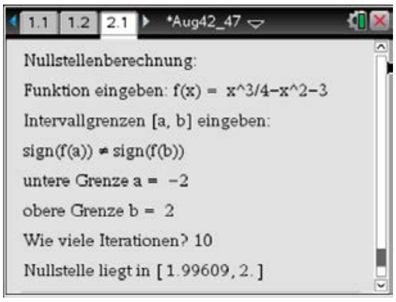
Daher passen wir die Zeilen 18 und 19 entsprechend an:

```
…
:  EndIf
:EndFor
:Disp "Nullstelle liegt in [",a*1.,",",b*1.,"]"
```

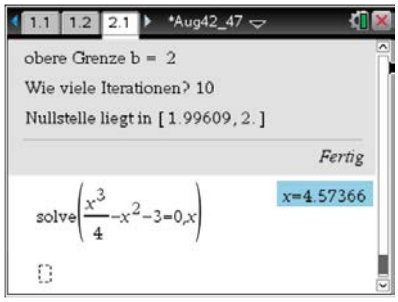
Die Multiplikation mit 1.0 erzwingt die dezimale Ausgabe (approx(a) hätte das auch erreicht.) Wir zeigen noch den Näherungswert für den *goldenen Schnitt* und dann noch ein Beispiel mit einer fehlerhaften Ausgabe – die aber durch eine falsche Eingabe verursacht wurde.







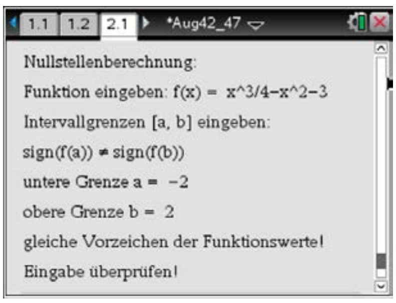
Ein richtig sauberes Programm würde hier noch intern abfragen, ob tatsächlich verschiedene Vorzeichen an den Intervallgrenzen vorliegen und gegebenenfalls eine entsprechende Meldung ausgeben.

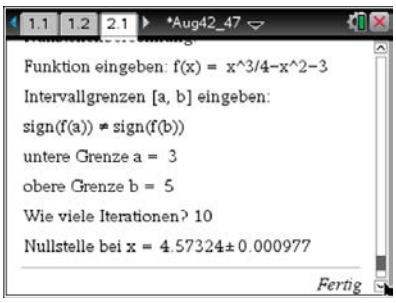


Aufgabe 43a *Programm mit Fehlerabfrage und anderer Ausgabe*

Baue die Überprüfung einer fehlerhaften Eingabe ein und gib das Ergebnis als Mitte des letzten Intervalls ± halber Intervallbreite aus.

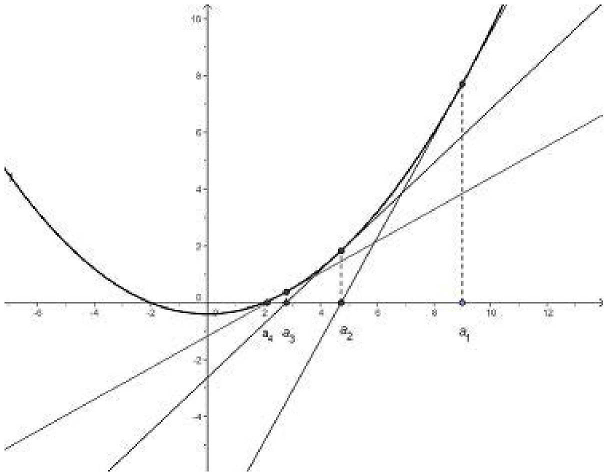
```
…
:Request "obere Grenze b = ",b
:If f(a)*f(b)>0 Then
:     Disp "gleiche Vorzeichen der Funktionswerte!"
:     Disp "Eingabe überprüfen!"
:     Goto ende
:EndIf
:Request "Wie viele Iterationen?",n
…
Disp "Nullstelle bei x = ",((a+b)/(2.)),"±",((abs(b−a))/(2.))
:Lbl ende
:EndPrgm
```





Aufgabe 44 *Newtonsche Näherungsmethode*

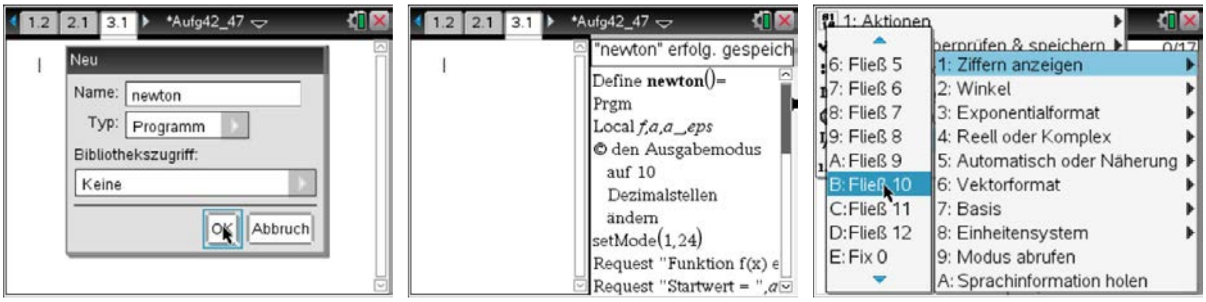
Diese Methode wird in vielen Texten auch als *Newton-Raphson-Methode* bezeichnet. Um die Nullstellen näherungsweise zu berechnen, wird die Kurventangente verwendet. Wenn a_1 der erste Näherungswert für eine Nullstelle von $f(x)$ ist, dann können wir die Tangente an der Stelle $x = a_1$ angeben: $y = f'(a_1)(x - a_1) + f(a_1)$. Der Schnittpunkt der Tangente mit der x-Achse ergibt sich dann als $x = a_1 - \dfrac{f(a_1)}{f'(a_1)} = a_2$. a_2 ist der nächste Näherungswert. Somit erhalten wir eine Folge a_1, a_2, a_3, ..., die in vielen Fällen zur gewünschten Nullstelle – manchmal aber auch zu einer anderen – konvergiert. Es ist zu bedenken, dass nicht alle Startwerte a_1 zum gewünschten Erfolg führen. Dazu gibt es viele Informationen in einschlägigen Quellen.



Lösung 44 Das Programm ist nicht allzu schwierig:

```
Define newton()=
Prgm
:Local f,a,a_,eps
:© den Ausgabemodus auf 10 Dezimalstellen ändern
:setMode(1,24)
:Request "Funktion f(x) eingeben: ",f(x)
:Request "Startwert = ",a
:Request "Genauigkeit = ",eps
:© Beginn der Iterationsschleife
:Loop
  :a_:=x−f(x)/d(f(x),x)|x=a
  :© Genauigkeitsabfrage
  :If abs(a−a_)<eps:Goto ende
  :a:=a_
:EndLoop
:Lbl ende
:Disp approx(a_)
:© den Ausgabemodus auf 6 Stellen rücksetzen
:setMode(1,18)
:EndPrgm
```

Wir haben hier einige zusätzliche Möglichkeiten eingebaut. Die Kommentare sollten schon ausreichend erklären. In den Programmzeilen 4 und 18 steuern wir die Anzahl der Dezimalstellen. Es lassen sich auch andere Einstellungen programmintern ändern. In das entsprechende Menü kommst du über menu > 7:modus > 1:Ziffern anzeigen > ... In unserem Lösungsvorschlag haben wir keine For – EndFor-Schleife mit einer vorgegebenen Anzahl von Iterationen verwendet, sondern wir wollen bis zu einer vom Benutzer angegebenen Genauigkeit ε rechnen. Dazu setzen wir eine Loop - EndLoop-Schleife ein, die bei erreichter Genauigkeit – zwei aufeinander folgende Näherungswerte haben einen Abstand < ε – verlassen wird.



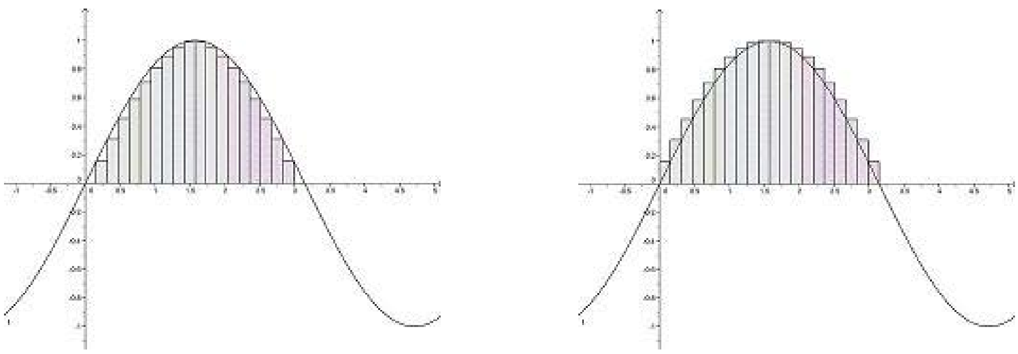
Wir berechnen die Nullstellen von $y = x^2 - 3$ und von $y = x^2$.

(Führe diese Rechnungen auch mit int_halb() aus und vergleiche die Ergebnisse! Welches Verfahren konvergiert rascher?)

Mit einer einzigen Programmzeile mehr können wir die Ausgabe der Folge von Näherungswerten veranlassen. Führe diese kleine Erweiterung selbst durch.

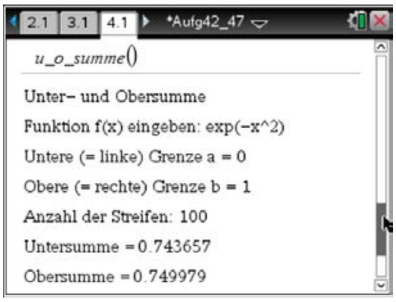
Aufgabe 45 *Numerische Integration – Ober- und Untersummen*

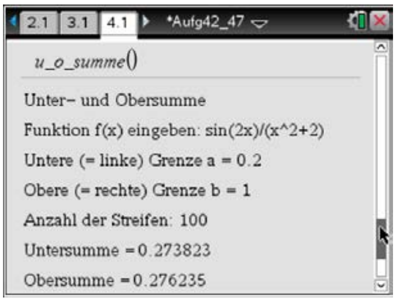
Den Wert eines bestimmten Integrals kann man näherungsweise durch eine *Untersumme* und *Obersumme* berechnen. Der tatsächliche Wert liegt dann zwischen diesen beiden Summen (siehe Abbildung).

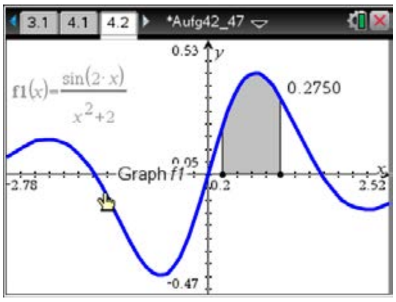


Schreibe ein kurzes Programm, das Ober- und Untersummen bildet und berechne so z.B. $\int_0^{\pi} \sin x \, dx$.

Lösung 45 Wir fragen wieder zuerst um die Funktion und anschließend um die Integrationsgrenzen *a* und *b*. Das Integrationsintervall wird in *n* gleiche Teile zerlegt. Damit entsteht eine Folge von Stützpunkten $x_1, x_2, \ldots, x_{n-1}$ mit dem Abstand $h = \frac{b-a}{n}$. Für die Untersumme bilden wir das Minimum zweier aufeinander folgender Funktionswerte und für die Obersumme das Maximum. (Wenn wir ganz penibel sind, finden wir heraus, dass damit nicht unbedingt die wirkliche Unter- bzw. Obersumme entstehen muss. Wenn z.B. eine lokales Maximum zwischen zwei Stützstellen liegt, dann wird nicht das umschreibende Rechteck für die Summe berücksichtigt, sondern ein etwas kleineres. Dieser „Fehler“ wirkt sich aber bei unseren Näherungswerten überhaupt nicht entscheidend aus.)



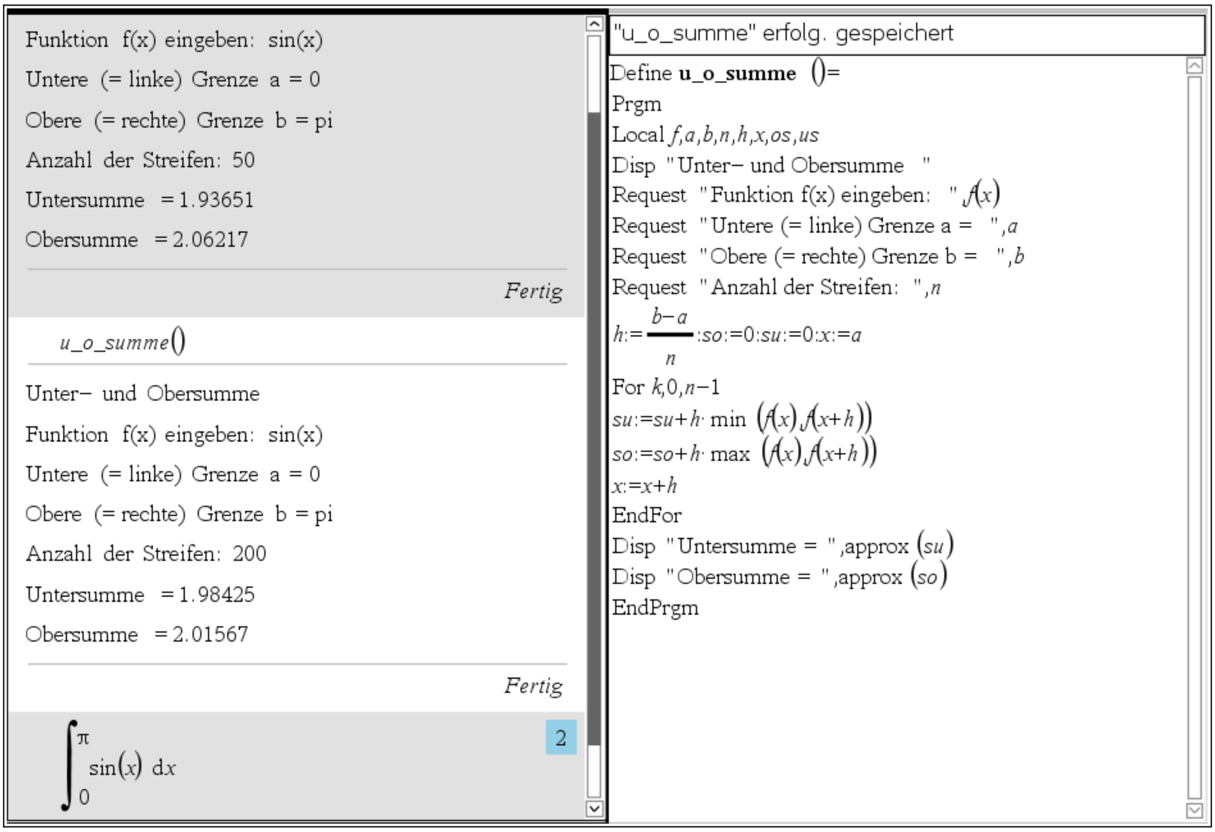




Die beiden letzten Integrale lassen sich sicherlich nicht in geschlossener Form berechnen. TI-Nspire[TM] CX CAS hat ein noch effizienteres numerisches Verfahren zur Berechnung derartiger Integrale implementiert. Das Rechenfenster liefert als Ergebnis den Wert

$$\int_{0.2}^{1} f1(x)\,dx \qquad 0.275036$$

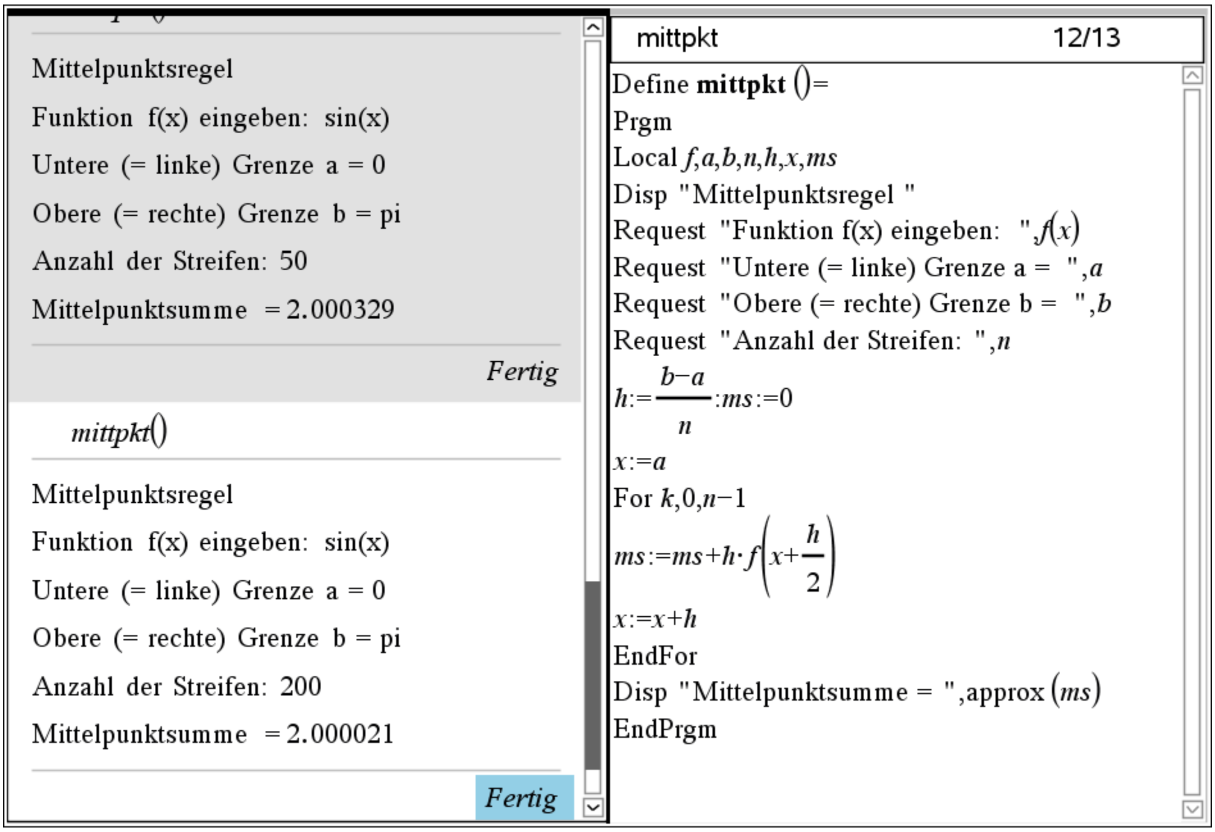
Wir zeigen den screen shot von der PC-Version (Programm + Testläufe):

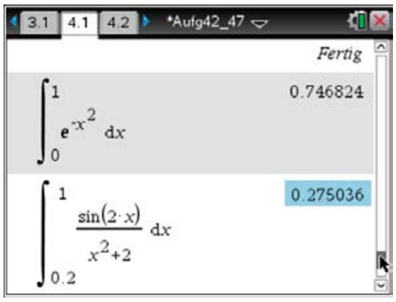


Aufgabe 46 *Numerische Integration – Mittelpunktsregel*

In dieser Methode wird das Intervall [a, b] wieder in n gleiche Teile zerlegt. Die Höhe eines jeden Teilrechtecks ist dann der Funktionswert in der Mitte des zugehörigen Streifens. Berechne auch auf diese Weise die Integrale von Aufgabe 45 und zeige, dass hier – bei gleicher Anzahl der Streifen – ein besserer Näherungswert gebildet wird

Lösung 46 Das Programm sollte sich von selbst erklären





Aufgabe 46a *Numerische Integration – Trapezregel*

Nach der Trapezregel werden keine Rechtecke ein- bzw. umschrieben, sondern Trapeze. Das k-te Trapez hat den Flächeninhalt $h \cdot \dfrac{f(x_k) + f(x_k + h)}{2}$. Berechne die bestimmten Integrale von vorhin auch auf diese Weise. Kannst du einen Zusammenhang zwischen Trapezregel und Unter- bzw. Obersumme erkennen? Nütze diesen Zusammenhang und baue die Trapezregel in das Programm aus Aufgabe 44 ein.

Aufgabe 47 *Flächenberechnung in Polarkoordinaten*

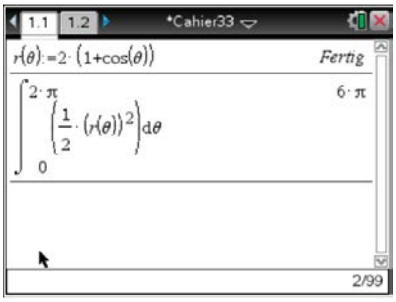
Eine Kurve liegt vor in Polarkoordinatendarstellung: $r = f(\theta)$. Die Fläche, die von dieser Kurve bei einem Umlauf von r begrenzt wird, ist gegeben durch

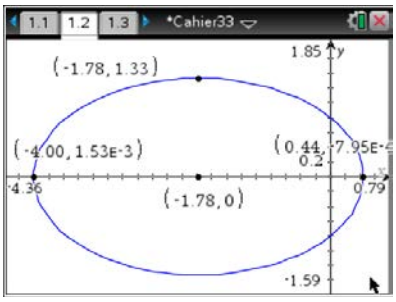
$$A = \int_0^{2\pi} \tfrac{1}{2} r^2 \, \mathrm{d}\theta.$$

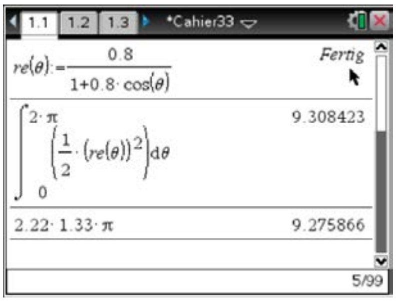
Warum?

Lösung 47 Wir wissen, dass die Kreisfläche $r^2\pi = \frac{1}{2} r^2 \cdot 2\pi$ beträgt. Der Flächeninhalt eines Halbkreises ist demnach $\frac{r^2\pi}{2} = \frac{1}{2} r^2 \pi$. Ein Kreissektor mit dem Öffnungswinkel θ hat den Flächeninhalt $\frac{1}{2} r^2 \theta$. Es ist einsichtig, dass ein *infinitesimaler* Kreissektor den Inhalt $\frac{1}{2} r^2 \, \mathrm{d}\theta$ aufweist. Wenn wir nun über den ganzen Winkel $\theta = 2\pi$ aufsummieren, ergibt sich die obige Formel. Für manche in Polarkoordinaten gegebene Kurven brauchen wir da – im Gegensatz zur Berechnung mit einem graphischen Taschenrechner wie z.B. dem TI-84 – kein Programm sondern wir berechnen sofort:

zuerst die Fläche begrenzt von der Kardiode aus Aufgabe 19, dann die Fläche einer Ellipse.

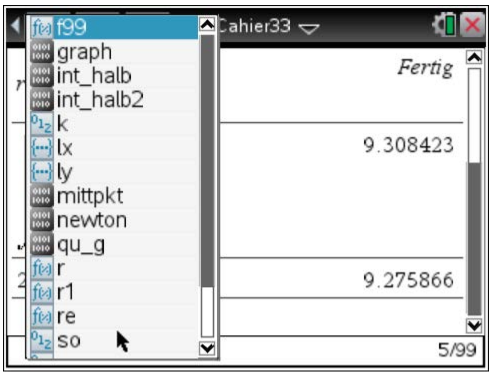


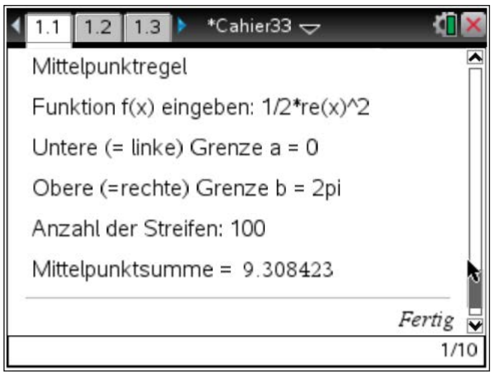




Dieses Integral (Ellipse) kann auch anders näherungsweise berechnet werden. Aus dem Graph der Ellipse lesen wir möglichst genau die Länge der Halbachsen a und b ab und berechnen $A = a \cdot b \cdot \pi$. Da wollen wir doch gleich unser Programm zur Mittelpunktsregel testen:

Über die Taste var kommen wir zur Übersicht über alle in diesem Dokument verfügbaren Variablen, Funktionen und Programme. Ein Klick auf mittpkt bringt den Programmaufruf in die Eingabezeile.





Mit dem Ergebnis können wir wohl zufrieden sein.

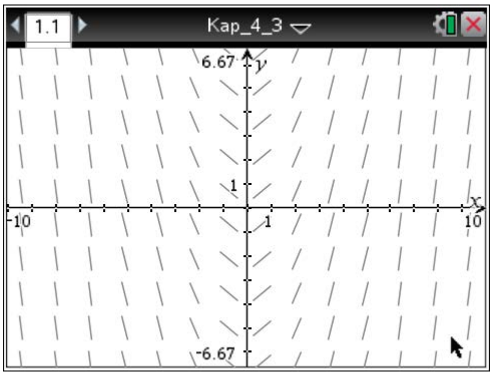
Achtung: Diese Flächenberechnung – in Polarkoordinaten – gilt nur für Kurven ohne Schleifen, die ein bestimmtes Gebiet G umgrenzen, wo mit jedem Punkt $P \in G$ auch der Strahl OP vollständig in G enthalten ist. Anderenfalls muss der Flächeninhalt in entsprechende Teilintegrale zerlegt werden.

4.3 Jetzt ohne Programmieren ...

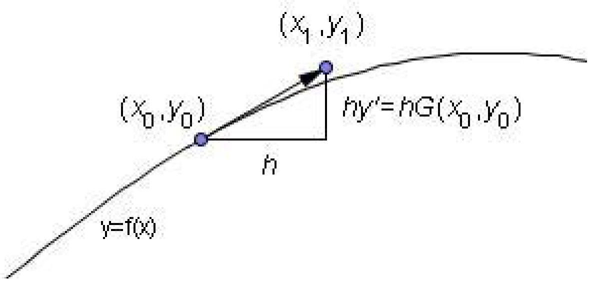
Im Folgenden wollen wir zeigen, wofür TI-Nspire™ CX CAS im Bereich der Analysis noch eingesetzt werden kann. Von besonderem Interesse sind die analytischen und numerischen Möglichkeiten einfache Differentialgleichungen zu lösen.

Die Grundaufgabe zur Differentialgleichung ist das Auffinden der Funktion $y = f(x)$ aus ihrer Ableitung $y' = G(x, y)$. Einige Typen von Differentialgleichungen lassen sich durch Integrieren – und einige Tricks – lösen, aber bei weitem nicht alle. Doch lässt sich der Graph von $f(x)$ mit TI-Nspire™ CX CAS auf einfache Weise – unter Anwendung der eulerschen Methode oder des komplizierteren Runge-Kutta-Verfahrens – finden.

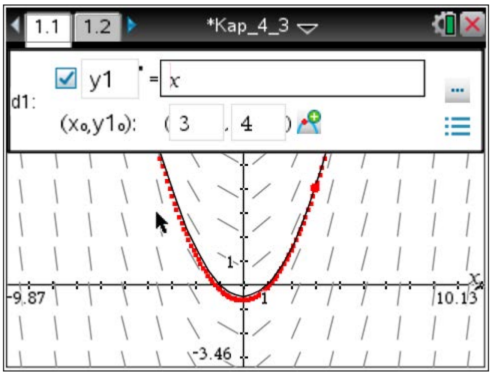
Mit $y' = f'(x) = G(x, y)$ ist eigentlich in jedem Punkt (x, y) der Ebene ein Richtungsvektor (eine Tangente) gegeben. Wir könnten in jedem Punkt ein kleines Stückchen Tangente zeichnen. Das führt zu einem *Richtungsfeld*. Wenn wir z.B. das Richtungsfeld zur DGL $y' = x$ zeichnen, kann dies zunächst so aussehen:

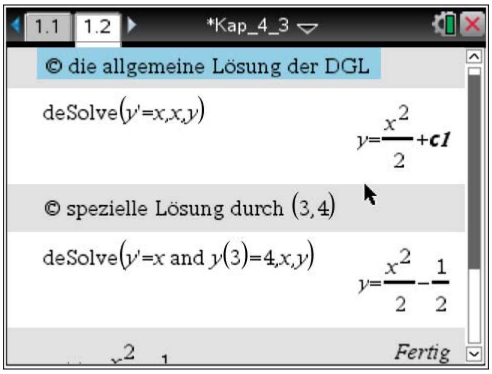


Wie gelangen wir nun zum Graph von f? Nimm an, dass wir von einem Punkt (x_0, y_0) verlangen, dass er auf der gesuchten Kurve liegt. In diesem Punkt kennen wir den Richtungsvektor $\left(1, y' = G(x_0, y_0)\right)$. Wenn wir nun einen kleinen Schritt $h > 0$ nach rechts (in positiver x-Richtung) und gleichzeitig einen Schritt $h \cdot G(x_0, y_0)$ in y-Richtung gehen, folgen wir ein kurzes Stück Weg längs der Tangente und gelangen zu einem nächsten Punkt $(x_1, y_1) = (x_0 + h, h \cdot G(x_0, y_0))$, der wohl sehr nahe bei der gesuchten Kurve liegt. In diesem Punkt führen wir die gleiche Prozedur durch und gelangen zum Punkt (x_2, y_2) u.s.w. So folgen wir Schritt für Schritt – aber doch nur näherungsweise – dem richtigen Graph. Es ist klar, dass die Qualität des Verfahrens von der gewählten Schrittweite h abhängt.



Wenn wir die so gewonnenen Punkte verbinden, erhalten wir schon eine recht gute Annäherung zur gewünschten Lösungs-(Integral-)kurve $y = f(x)$ der DGL. Wir haben aber eine freie Wahl: der erste Punkt (x_0, y_0) kann frei gewählt werden und er liegt dann immer auf der Lösungskurve. In unserem Beispiel scheint die Lösung die Form einer Parabel zu haben.





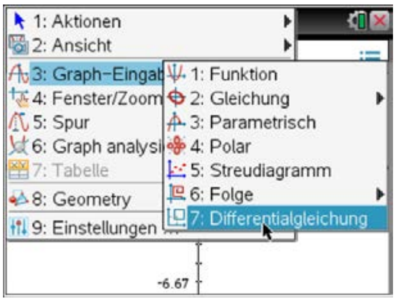
In diesem Fall hätten wir die Lösungskurve auch durch Integration mit $y=\dfrac{x^2}{2}+c$ mit einem geeigneten Wert für c finden können. TI-Nspire™ CX CAS macht auch dies für uns. Der rote Graph ist die exakte Lösung der Differentialgleichung mit der gegebenen *Anfangsbedingung*.

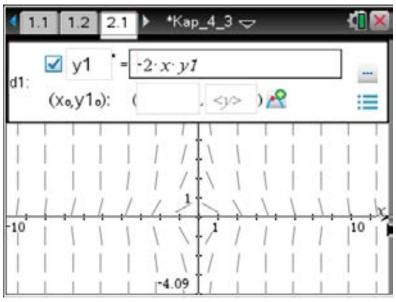
Wie bringen wir das nun alles in den TI-Nspire™ CX CAS?

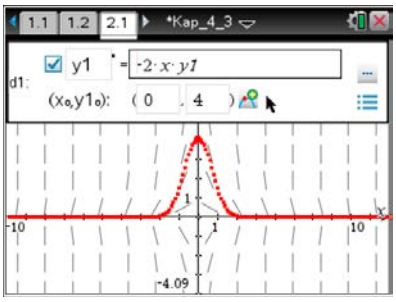
Aufgabe 48 *Differentialgleichungen – Richtungsfeld – eulersches Verfahren*

Erzeuge das Richtungsfeld zur DGL $y' = -2x\,y$. Zeige, dass TI-Nspire™ CX CAS die Methode von Euler verwendet. Bestimme auch die allgemeine Lösung der DGL. Löse dann auch die DGL $y' = e^{-x^2}$.

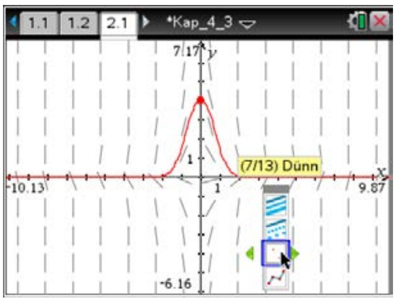
Lösung 48 Wenn wir das Richtungsfeld zeichnen wollen, müssen wir in einem Grafikfenster beginnen.

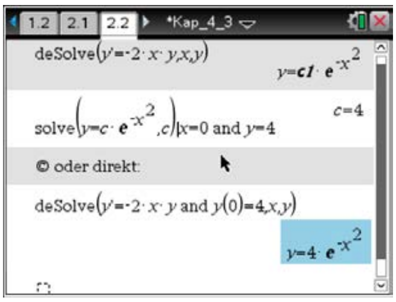






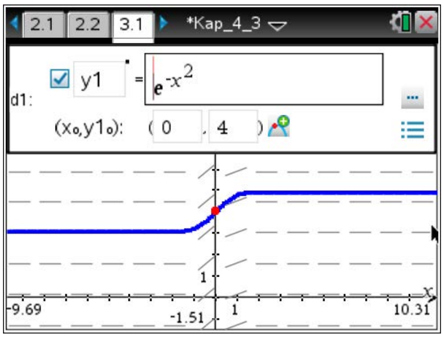
Ohne Anfangsbedingung gibt es keinen Graph! Über die Attribute zeichnen wir die Punkte klein und verbinden sie, um eine Kurve zu erhalten. Weitere Einstellungen zur graphischen Lösung der DGL erhalten wir über die [...] neben der Eingabezeile. Hier finden wir u.a. die Angabe des Lösungsverfahrens.

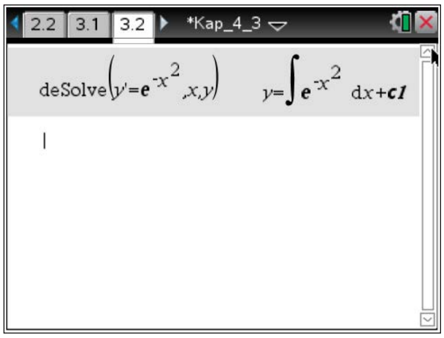




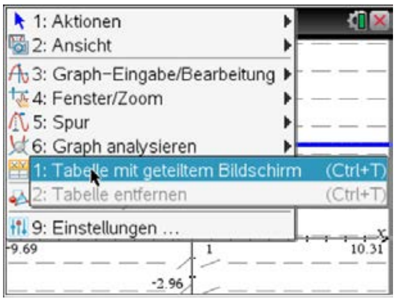
Die CAS-Lösung ist leicht gefunden.

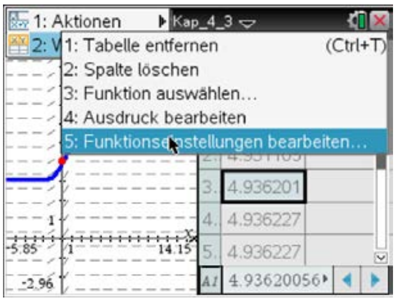
Bei der zweiten DGL finden wir die graphische Lösung ebenso einfach. Die exakte Lösung kann hier nicht gefunden werden, da die Funktion nicht integrierbar ist.

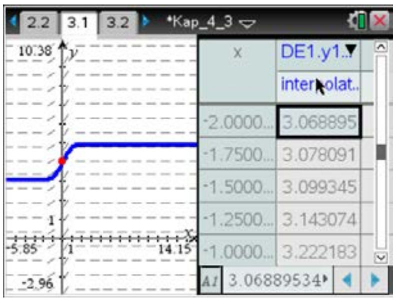




Haben wir hier überhaupt keinen Zugang zur Lösungskurve? Oh, doch, denn es ist ein Leichtes, eine Wertetabelle zu erstellen. menu > 7:Tabelle > 1:Tabelle mit ... zeigt uns sofort die Wertetabelle der Integralkurve. Wir können – müssen aber nicht - die linke Hälfte löschen und über menu > Wertetabelle > Funktionseinstellungen bearbeiten ... den Tabellenanfang, aber vor allem die Schrittweite unseren Wünschen anpassen.





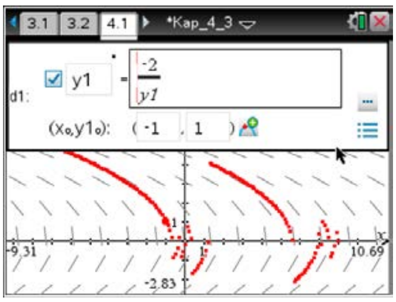


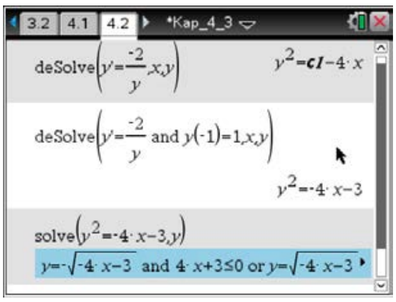
Aufgabe 49 *Differentialgleichungen – Richtungsfeld – eulersches Verfahren*

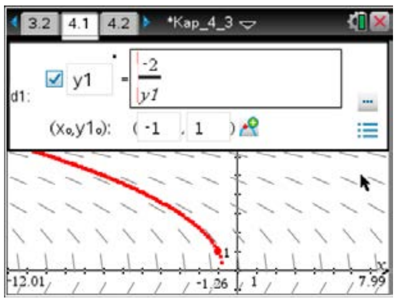
Das Verfahren von Euler ist nicht besonders gut. Es kommt leicht zu „merkwürdigen“ Ergebnissen.

Sieh dir z.B. die graphische Lösung von $y' = -\frac{2}{y}$ mit $y_0(x_0 = -1) = 1$ an.

Lösung 49



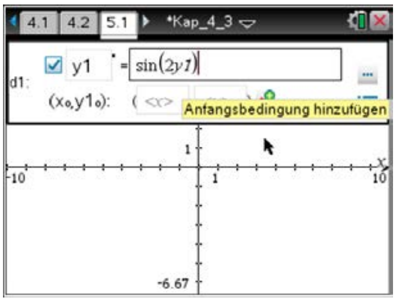


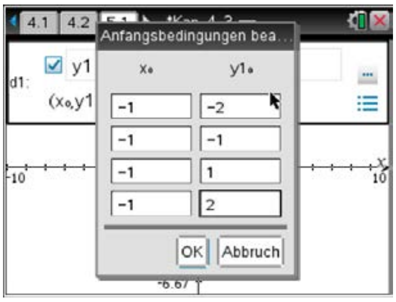


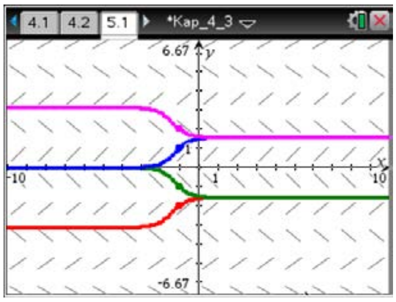
Eine kurze Rechnung zeigt, dass die Lösung eine Parabelhälfte ist. In der Nähe ihres Scheitels ist die Tangente fast senkrecht. In derartigen Punkten ist die eulersche Methode sehr sensibel bezüglich kleinster Abweichungen und kann daher nicht mehr zu einer vernünftigen Näherungskurve führen. Ganz anders verhält sich die Runge-Kutta-Methode, welche die Kurve recht ordentlich erzeugt (rechtes Bild).

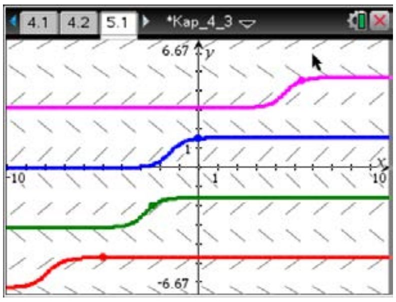
Aufgabe 50 *Differentialgleichungen – Richtungsfeld – eulersches Verfahren*
Wenn wir die *x*-Achse als Zeitachse interpretieren, dann wird eine horizontale Asymptote den nach längerer Zeit erreichbaren stabilen Zustand darstellen. Betrachte die DGL $y' = \sin(2y)$. Wie hängt ihr langfristiges Verhalten vom Anfangswert ab?

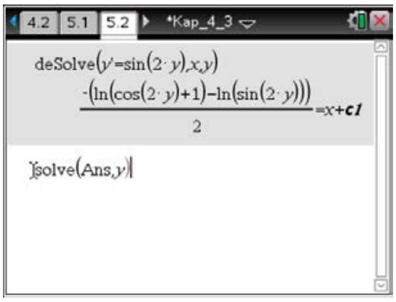
Lösung 50 Wir können den kleinen Button in der letzten Zeile (weißes Plus im grünen Kreis) verwenden, um eine Tabelle zu öffnen, in die wir mehrere Anfangsbedingungen eintragen können.

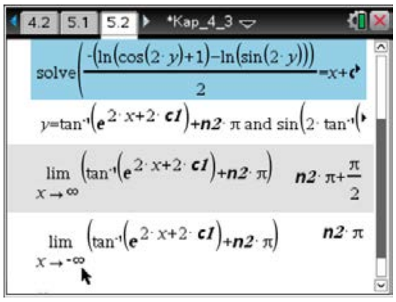










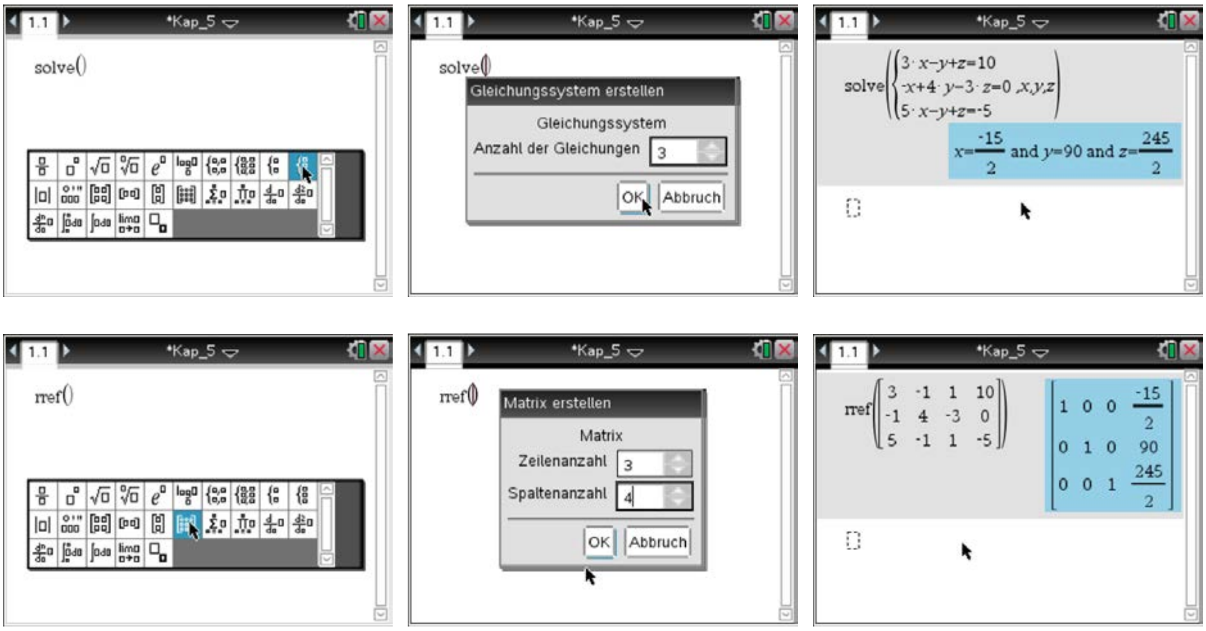


Für die vier Anfangsbedingungen erhalten wir die Gleichgewichtslagen bei *y* = $y=\pm\frac{\pi}{2}$. Doch es geht noch weiter. Die Punkte der Anfangsbedingungen lassen sich mit dem Cursor verziehen und da merken wir, dass die waagrechten Asymptoten geradezu springen und zwar nach beiden Richtungen. Neugierig geworden lassen wir unser CAS arbeiten: zuerst wird die DGL gelöst. Für dieses Ergebnis hätte wir händisch sicherlich länger gebraucht. Die implizite Lösung lässt sich sogar explizit machen – und dann bilden wir die Grenzwerte für $x \to \pm\infty$. Die Kurven gehen in positiver Richtung gegen $k \cdot \pi + \frac{\pi}{2}$ und in negativer Richtung gegen Vielfache von π. Das ist sicherlich ein unerwartetes Ergebnis, zu dem wir mit Hilfe des CAS rasch und mühelos gelangen. Warum das nun so ist, sagt uns das CAS nicht, da müssen wir selbst unsere mathematischen Kenntnisse einbringen.

5 Einige weitere Möglichkeiten von TI-Nspire™ CX CAS

5.1 Lösen von Gleichungssystemen

Mit Hilfe des CAS lassen sich – lineare – Gleichungssysteme mühelos lösen. Dafür stehen sowohl der Befehl solve als auch rref zur Verfügung. rref stützt sich auf die gaußsche Eliminationsmethode



Für beide Methoden verwenden wir die Vorlagen. Wenn wir mit rref() von der Koeffizientenmatrix ausgehen, können wir in der letzten Spalte die Lösungen ablesen. Dieses Beispiel ist der Idealfall mit einer eindeutigen Lösung. Wie TI-Nspire™ CX CAS in den anderen Fällen reagiert, wollen wir in den Aufgaben erkunden.

Aufgabe 51 Verwende eine *Vandermonde*-Matrix um eine Polynomfunktion vom Grad $n-1$ aus n gegebenen Punkten des Graphs zu bestimmen.

Lösung 51 Bei gegebenen n Punkten können wir davon ausgehen, dass die Polynomfunktion vom Grad $n-1$ ist und die folgende Form hat:

$$f(x)=a_0+a_1x+a_2x^2+a_3x^3+\ldots+a_{n-1}x^{n-1}.$$

Gesucht sind die Koeffizienten a_0 bis a_{n-1}. Der Funktionsgraph enthält die Punkte (x_1,y_1), (x_2,y_2), ..., (x_n,y_n). Mit diesem Wissen ausgestattet lässt sich ein lineares Gleichungssystem aufstellen:

$$\begin{aligned}
f(x_1) &= a_0+a_1x_1+a_2x_1^2+\ldots+a_{n-1}x_1^{n-1} &= y_1\\
f(x_2) &= a_0+a_1x_2+a_2x_2^2+\ldots+a_{n-1}x_2^{n-1} &= y_2\\
&\vdots\\
f(x_n) &= a_0+a_1x_n+a_2x_n^2+\ldots+a_{n-1}x_n^{n-1} &= y_n
\end{aligned}$$

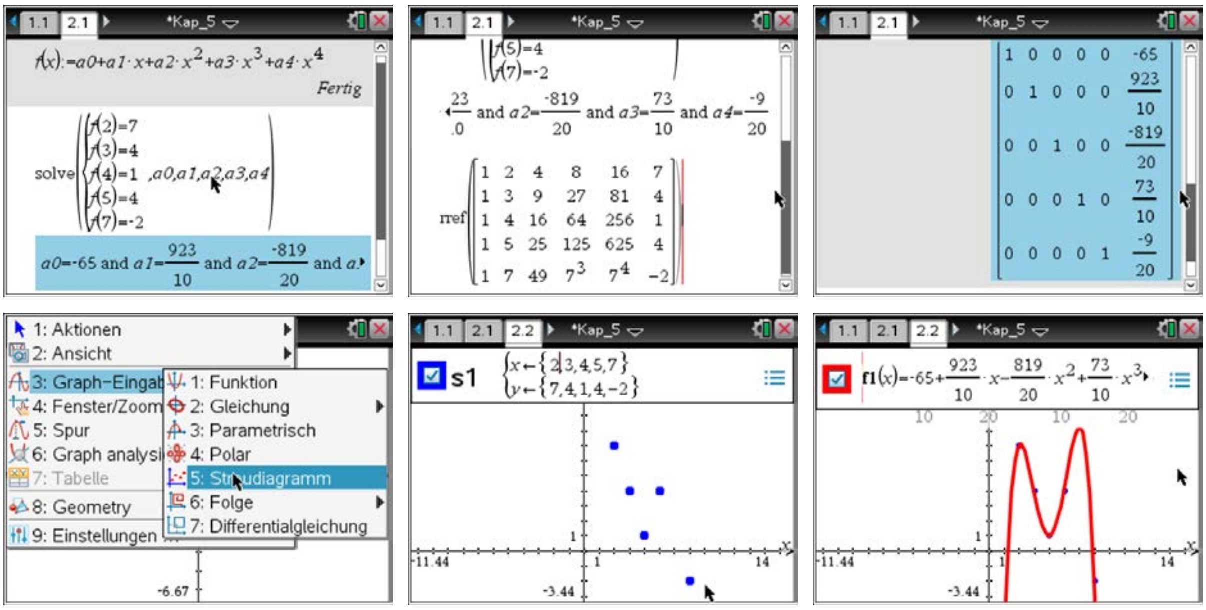
Das Gleichungssystem stellt sich in Matrixschreibweise so dar:

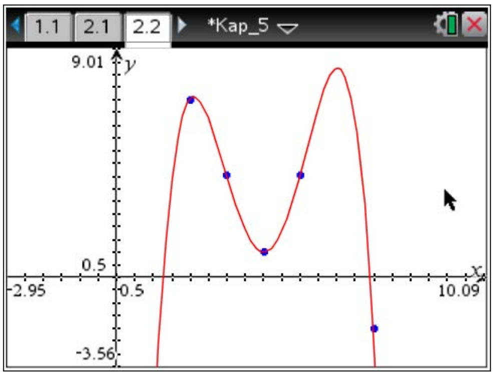
$$\begin{bmatrix}1 & x_1 & x_1^2 & \ldots & x_1^{n-1}\\ 1 & x_2 & x_2^2 & \ldots & x_2^{n-1}\\ \vdots & \vdots & & \vdots & \\ 1 & x_n & x_n^2 & \ldots & x_n^{n-1}\end{bmatrix}\begin{bmatrix}a_0\\ a_1\\ \vdots\\ a_{n-1}\end{bmatrix}=\begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_n\end{bmatrix}$$

Diese Matrix der Potenzen der x-Werte wird *Vandermonde*-Matrix genannt. Mit Hilfe der um die y-Werte erweiterten Matrix wird dieses System mit der gaußschen Methode gelöst.

$$\left[\begin{array}{ccccc|c} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} & y_1 \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} & y_2 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^{n-1} & y_n \end{array}\right]$$

Nimm an, dass die Punkte (2,7), (3,4), (4,1), (5,4) und (7,–2) gegeben sind. Wie lautet die Funktion 4. Grades, deren Graph durch diese Punkte verläuft. Stelle auch den Ergebnisgraph dar.





Aufgabe 52 Löse die folgenden beiden Gleichungssysteme auf beide Arten. Interpretiere die Ergebnisse, die mit der rref-Methode ausgegeben werden.

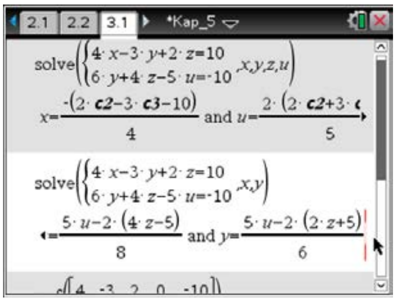
a)

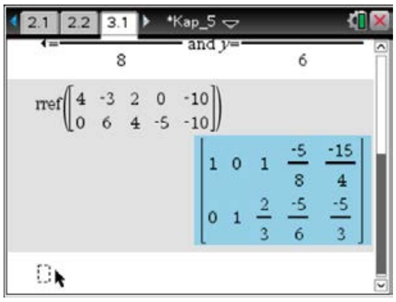
$$\begin{aligned} 4x - 3y + 2z &= 10 \\ 6y + 4z - 5u &= -10 \end{aligned}$$

b)

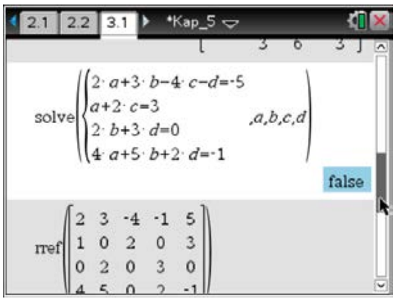
$$\begin{aligned} 2a + 3b - 4c - d &= -5 \\ a + 2c &= 3 \\ 2b + 3d &= 0 \\ 4a + 5b + 2d &= -1 \end{aligned}$$

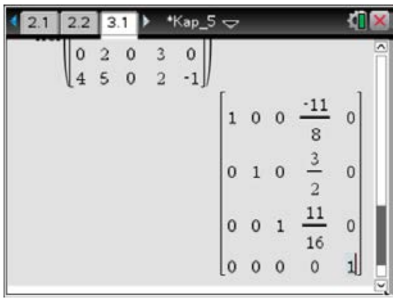
Lösung 52 Die Bildschirme zeigen die Vorgangsweise.





Bei Aufgabe a) gibt es zwei Parameter, die TI-Nspire™ CX CAS mit ***c2*** und ***c3*** bezeichnet. Vom Programm aus werden *x* und *u* in Abhängigkeit von diesen Parametern dargestellt. Wir können die Verteilung der Parameter steuern, indem wir gezielt z.B. nach *x* und *y* lösen lassen. Das Herauslesen der Lösung aus der mit rref umgeformten Matrix erfordert zusätzliche Rechenarbeit. Die zweite Zeile wird interpretiert als $y+\frac{2}{3}z-\frac{5}{6}u=-\frac{5}{3}$. Nach *y* aufgelöst finden wir den gleichen Term wie im linken Bild. Der Wert für *y* ist in die erste Gleichung einzusetzen und diese nach *x* aufzulösen.





In Aufgabe b) ist in der letzten Zeile ein Widerspruch zu sehen (0 = 1), daher gibt es keine Lösung.

Aufgabe 53 Mit TI-Nspire™ CX CAS lassen sich auch nichtlineare Gleichungssysteme lösen. Löse die beiden Systeme

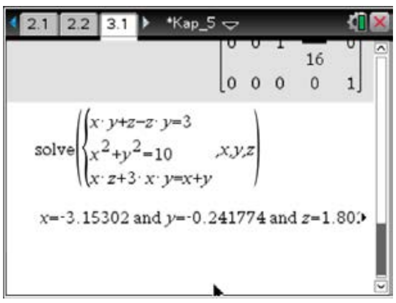
a)

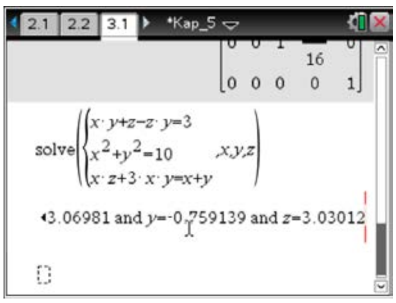
$$\begin{aligned} x\,y+z-z\,y&=3\\ x^2+y^2&=10\\ x\,z+3x\,y&=x+y \end{aligned}$$

b)

$$\begin{aligned} \sin(x)+y&=2\\ x+\cos(y)&=2 \end{aligned}$$

Lösung 53 TI-Nspire™ CX CAS hat – wie „große“ CAS-Programme – einen Algorithmus zur Lösung von polynomialen Gleichungssystemen implementiert (*Gröbner Basen*). Allerdings kann die Berechnung länger dauern oder bei zu komplexen Aufgaben überhaupt verweigert werden. Aufgabe 53 a) führt zu insgesamt sechs Lösungen, die numerisch ausgegeben werden:





Wir lassen dieses System – zur Kontrolle – auch den Vorgänger von TI-Nspire™ CX CAS (DERIVE) lösen:

$$\mathrm{SOLVE}\left(\left[x\cdot y + z - z\cdot y = 3,\ x^2 + y^2 = 10,\ x\cdot z + 3\cdot y\cdot x = x + y\right],\ [x,\ y,\ z]\right) = \left[x = 3 \wedge y = 1 \wedge z = -\frac{5}{3},\ x = \right.$$

$$\frac{59850990\cdot z^5 + 265403184\cdot z^4 - 1032200331\cdot z^3 - 1080487627\cdot z^2 + 871790297\cdot z + 1149427175}{917323160} \wedge y =$$

$$-\frac{73902690\cdot z^5 + 269396004\cdot z^4 - 1686149601\cdot z^3 - 933032267\cdot z^2 + 4651141587\cdot z + 491809435}{917323160} \wedge$$

$$\left. 30\cdot z^5 + 48\cdot z^4 - 827\cdot z^3 + 1181\cdot z^2 + 429\cdot z = 845\right]$$

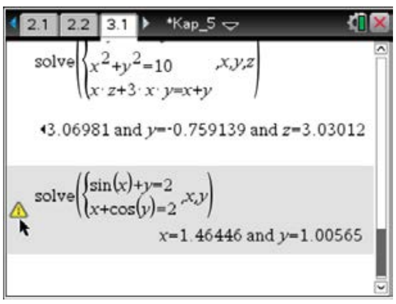
$$\mathrm{SOLUTIONS}(30\cdot z^5 + 48\cdot z^4 - 827\cdot z^3 + 1181\cdot z^2 + 429\cdot z = 845,\ z)$$

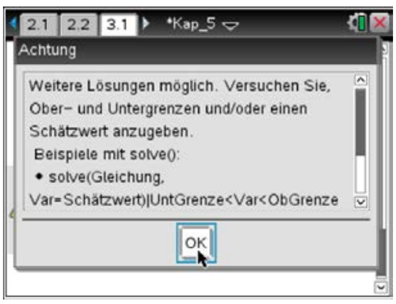
$$[-6.608494773,\ 3.030124048,\ 1.802002828,\ 0.9760779182,\ -0.7997100224]$$

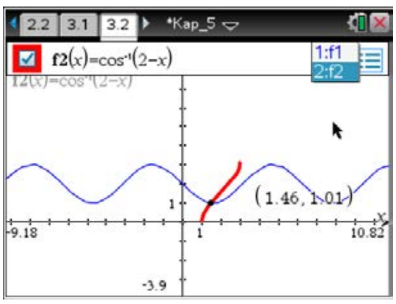
$$\begin{bmatrix} -2.261478391 & y = 2.210365424 \\ 3.069805913 & y = -0.7591387038 \\ -3.153021599 & y = -0.2417742253 \\ 0.3324902500 & y = -3.144749629 \\ 0.4122038355 & y = 3.135297114 \end{bmatrix}$$

Interessanterweise liefert uns TI-Nspire™ CX CAS gleich auf Anhieb alle 6 Lösungen, während uns DERIVE hier nur den Weg dazu weist. Das Gleichungssystem wird umgeformt: eine Lösung besteht aus rationalen Zahlen, alle anderen sind in der letzten Gleichung für *z* „versteckt". Die Gleichung 5. Grades kann nur mehr numerisch gelöst werden. Wenn wir nun alle 5 (hier zufälligerweise reellen) Lösungen in die Terme für *x* und *y* einsetzen, erhalten wir auch die vollständige Lösung.

Aufgabe b) liefert sofort eine Lösung mit dem Hinweis, dass es weitere Lösungen geben könnte. Um diese eventuell zu finden zeichnen wir die Graphen zu den beiden Gleichungen und sehen, dass es wirklich nur einen Schnittpunkt – und damit auch nur eine Lösung des Gleichungssystems gibt:





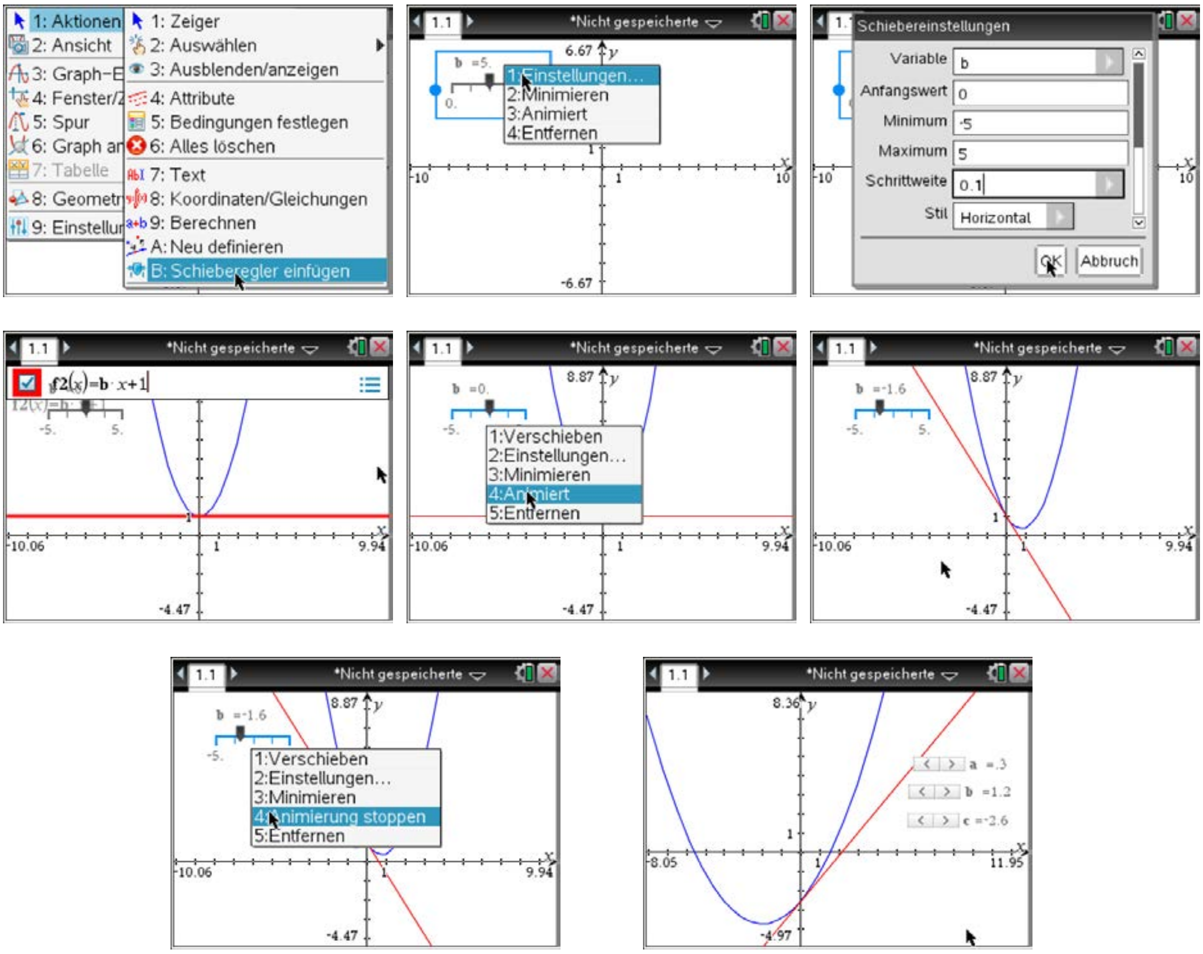


5.2 Animationen

Auf den Seiten 20, 22, 26 und anderen wurde schon auf die Möglichkeit hingewiesen, mit Hilfe des Schiebereglers eine Animation durchzuführen. Dieses wertvolle Werkzeug wird hier nochmals an Hand einer Aufgabe beschrieben.

Aufgabe 54 Untersuche mit einer Animation die Bedeutung des Parameters *b* für den Graph der Parabel $y = a\,x^2 + b\,x + c$. Zeichne auch die Gerade $y = b\,x + c$. Wähle geeignete Werte für *a* und *c*.

Lösung 54 Wir wählen *a* = –1 und *c* = 1. Für *b* installieren wir einen Schieberegler.



Es stellt sich heraus, dass der Parameter *b* der Anstieg der Kurventangente im Schnittpunkt mit der *y*-Achse ist.

Mit einem Klick auf den Schieberegler wird die Animation gestoppt.

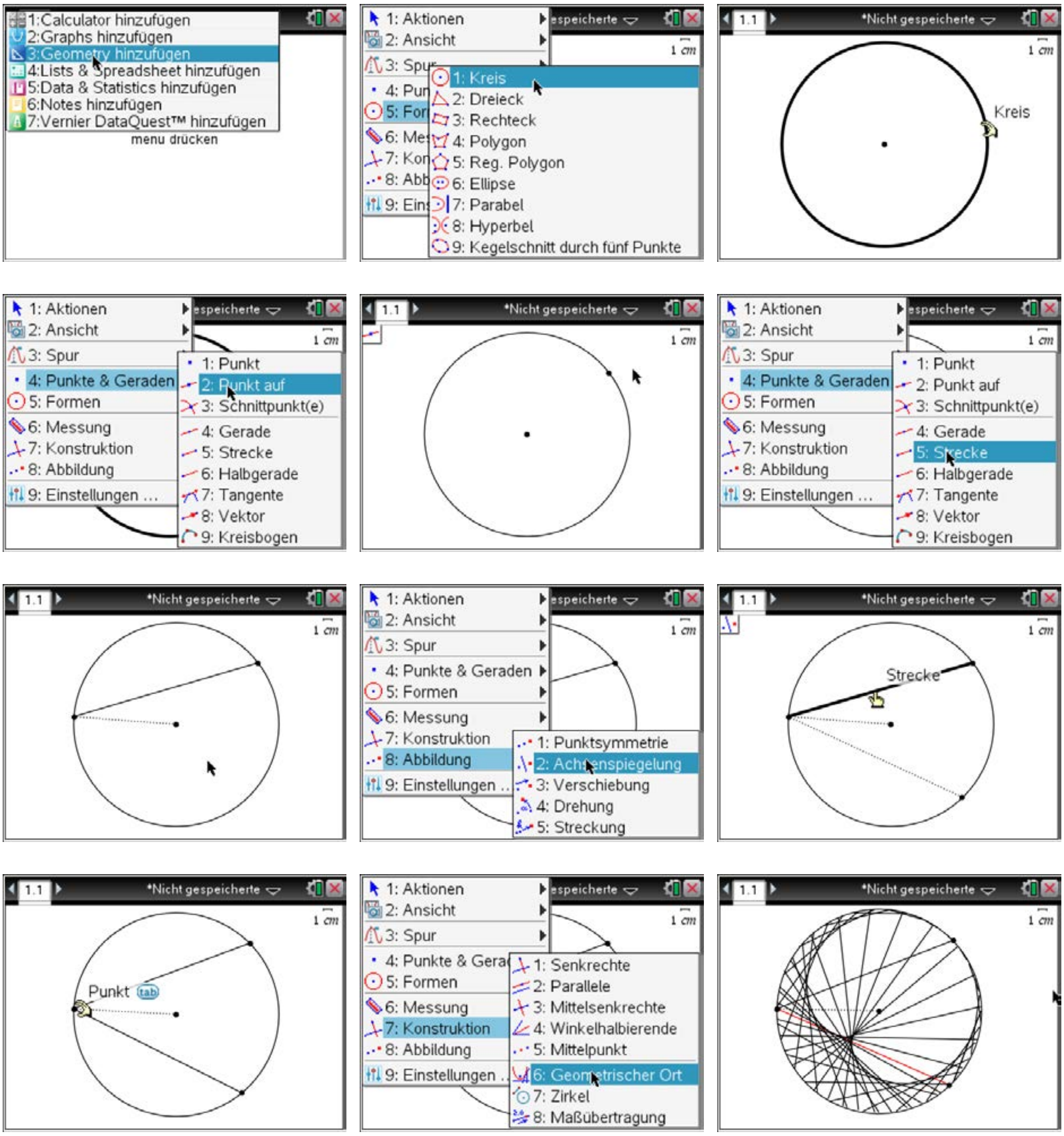
Durch Einführung von Schiebereglern für die Koeffizienten *a* und *c* wird weiter verallgemeinert. Eine gleichzeitige Animation von mehreren Reglern ist wohl möglich, bringt aber hier keinen Nutzen. Wenn mehrere Schieberegler verwendet werden ist es – vor allem am Handheld – sinnvoll, diese zu minimieren.

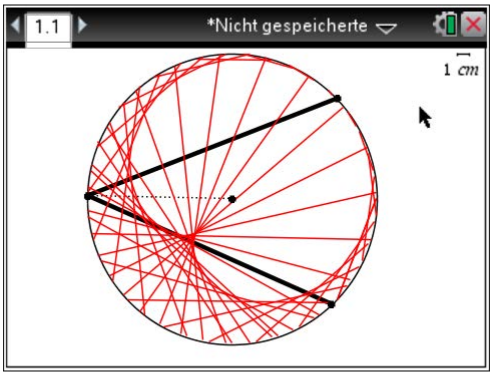
5.3 Hüllkurven und Ortslinien

Mit dem Werkzeug Geometrischer Ort auf einer *Geometry*-Seite kann man die Ortlinie eines Punktes oder die Hüllkurve einer Geraden oder eines Kreises in Abhängigkeit von einem bewegten Objekt (Punkt) erzeugen. Dafür wollen wir drei Beispiele zeigen.

Aufgabe 55 Erzeuge den Einfall eines Lichtstrahls über einen Tassenrand (siehe Aufgabe 19).

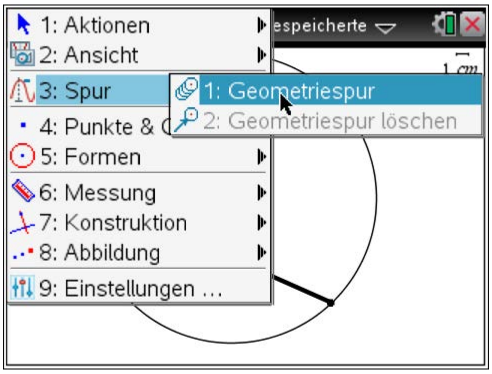
Lösung 55 Öffne eine *Geometry*-Seite, zeichne einen Kreis und auf diesem einen Punkt, der fest bleiben soll. Dieser Punkt symbolisiert die Lichtquelle. Von diesem Punkt gehen die Lichtstrahlen aus und werden am Kreis gespiegelt. Zeichne einen derartigen Strahl und spiegle ihn am Kreis mittels einer Achsenspiegelung (= Spiegelung an einer Geraden). Wenn der Auftreffpunkt des Strahls am Kreis bewegt wird, ergibt sich eine *Kardiode* als Hüllkurve der reflektierten Strahlen.

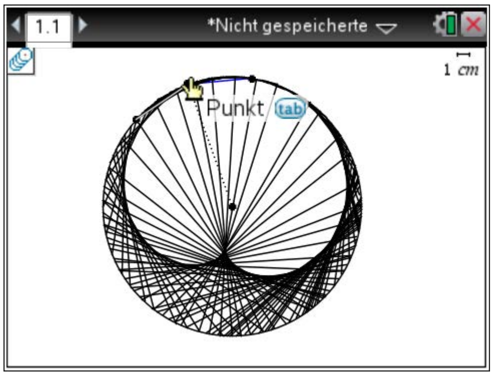




Die Farben können frei gewählt werden und über die Attribute lassen sich weitere „Verschönerungen“ ausführen.

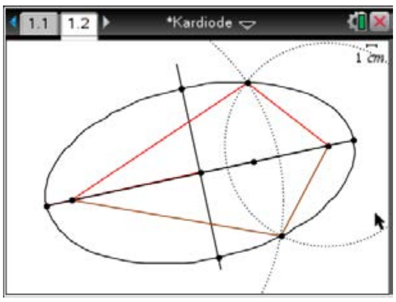
Willst du die Entstehung der Kardiode verfolgen, dann lösche zuerst die Hüllkurve und wähle im menu > 3:Spur > 1:Geometriespur und bewege den Punkt am Kreis. Der Punkt muss mit der tab-Taste angesprochen werden.

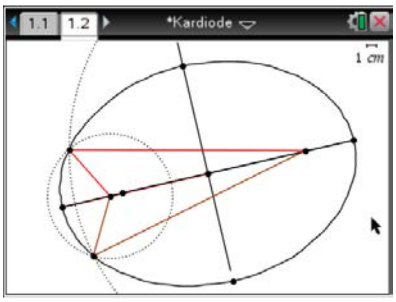


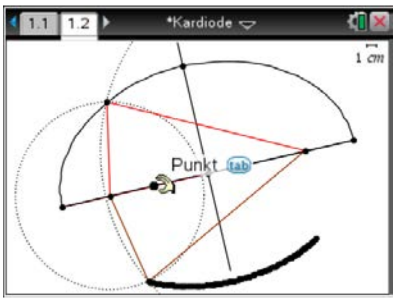


Aufgabe 56 *Die klassische Ellipsenkonstruktion*

Lösung 56 Wir setzen die Kenntnis der klassischen Brennpunktsdefinition der Ellipse voraus: sie ist die Menge aller Punkte, deren Summe der Abstände von zwei festen Punkten – den Brennpunkten – konstant $2a$ ist.







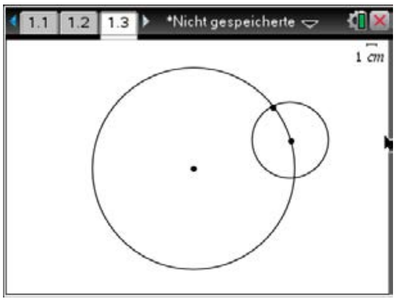
Versuche mit den Möglichkeiten, die im *Geometry*-Fenster geboten werden, diese Konstruktion durchzuführen. Die Achsenlängen lassen sich dann beliebig mit dem Cursor verändern. Der variable Punkt wurde auf der Hauptachse angenommen.

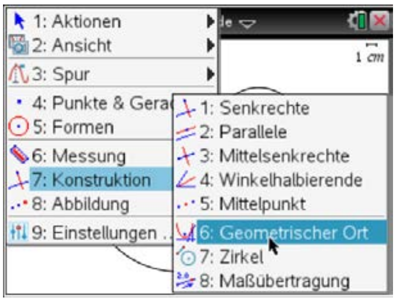
Im letzten Bild rechts wird in der oberen Hälfte die Ortslinie und in der unteren die Spur der Punkte gezeichnet. Elemente werden aus- und wieder eingeblendet über menu > 1:Aktionen > 3:Ausblenden/anzeigen.

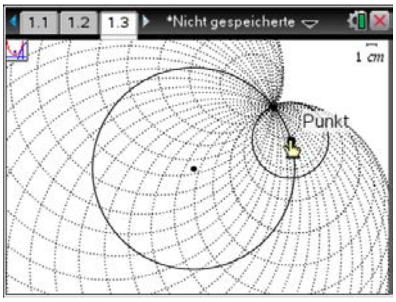
Aufgabe 56 *Und nochmals die Kardiode!*

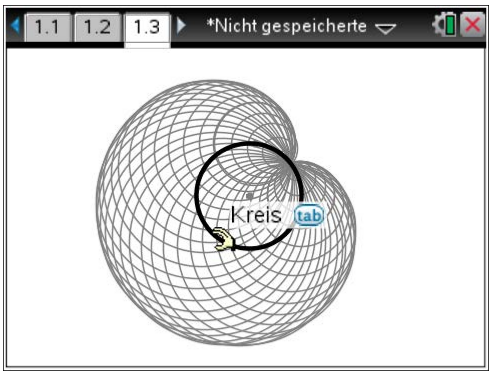
Zeichne einen Kreis und fixiere auf ihm einen festen Punkt. Betrachte nun alle weiteren Kreise, deren Mittelpunkt am gegebenen Kreis liegen und die durch den fixierten Punkt gehen. Erzeuge die Hüllkurve aller dieser Kreise!

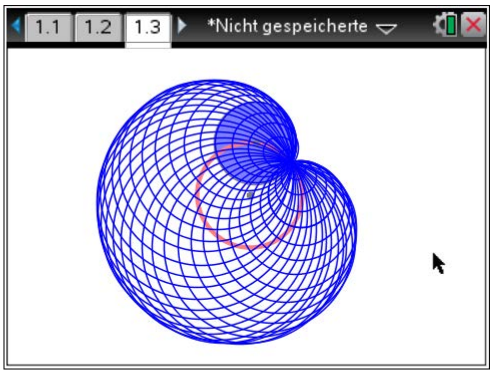
Lösung 5/











Wenn wir den Radius des festen ersten Kreises verkleinern, können wir die ganze Kardiode deutlich sehen. Für die graphische Gestaltung haben wir mehrere attraktive Möglichkeiten. (Tipp: wenn du die Kreise bearbeiten willst, dann blende zuerst die Kardioide aus.)

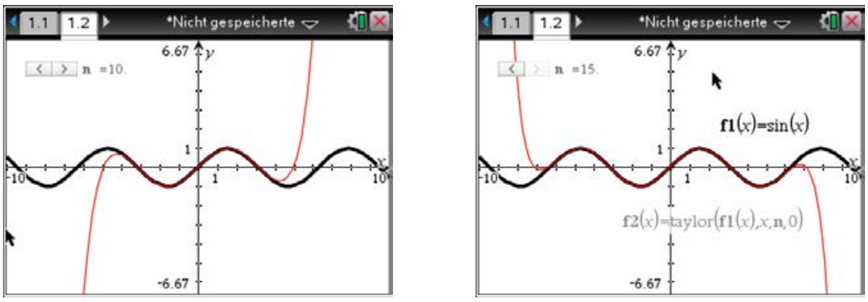
5.4 Taylorpolynome

Sei f eine Funktion, dann ist das Polynom $t_n(x,x_0)$ das Taylorpolynom n-ten Grades, das die Funktion f an der Stelle $x = x_0$ approximiert.

$$t_n(x,x_0) = f(x_0) + \frac{f'(x_0)}{1!}(x-x_0) + \frac{f''(x_0)}{2!}(x-x_0)^2 + \frac{f'''(x_0)}{3!}(x-x_0)^3 + \ldots + \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n$$

Aufgabe 55 *Taylorpolynom für den Sinus*
Verwende die Funktion taylor TI-Nspire™ CX CAS um sin(x) durch Taylorpolynome vom Grad n = 1 bis n = 1% an der Stelle x_0 = 0 anzunähern. Setze dazu einen Schieberegler ein. Animiere den Grad n.

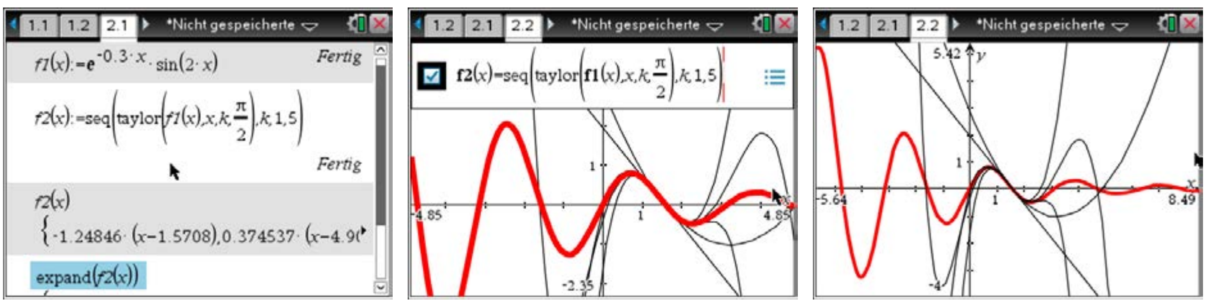
Lösung 55



Du kannst deutlich sehen, wie gut das Polynom 15. Grades die Sinusfunktion im Intervall [-π,+π] annähert. Alle trigonometrischen Funktionen werden programmintern durch Taylorpolynome auf die erforderliche Genauigkeit berechnet. Anlässlich der Animation fällt auf, dass für n = 3 und n = 4 das gleiche Polynom erscheint, dann für n = 5 und n = 6 wiederum das gleiche u.s.w. Ist das ein Fehler von TI-Nspire™ CX CAS oder hat das eine andere Ursache?

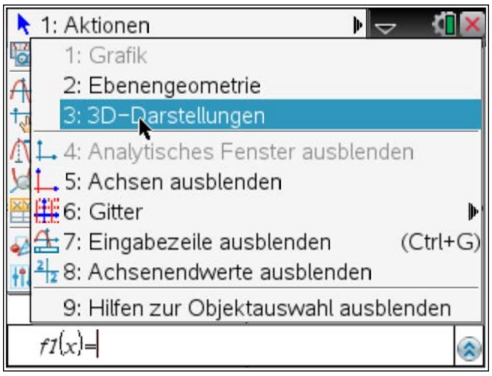
Aufgabe 56 Ermittle die Taylorpolynome vom Grad 1 bis 5 für $f(x) = e^{-0.3x} \cdot \cos(2x)$ and der Stelle $x_0 = \frac{3\pi}{2}$..

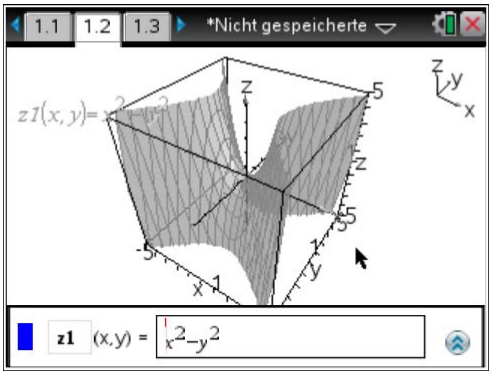
Lösung 56 Wir erzeugen alle Taylorpolynome als eine Folge.



5.5 3D-Grafik

Mit den aktuellen Versionen von TI-Nspire™ CX CAS lassen sich ordentliche 3D-Grafiken von Flächen und Raumkurven herstellen. Wir wollen von dieser attraktiven Anwendung drei Beispiele zeigen. Über die Aktionen rufen wir aus einem Grafikfenster die 3D-Darstellungen auf.



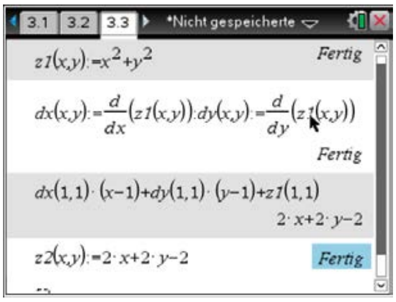


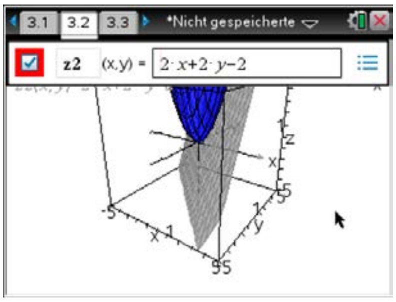
Aufgabe 57 *Tangentialebene an ein Paraboloid*
Die Gleichung der Tangentialebene an die Fläche $z = f(x,y)$ in (x_0, y_0) ist gegeben durch

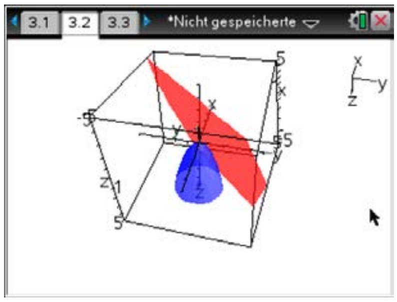
$$z = \partial_x f(x_0,y_0)(x - x_0) + \partial_y f(x_0,y_0)(y - y_0) + f(x_0,y_0).$$

Zeichne das Paraboloid $z = x^2 + y^2$ mit seiner Tangentialebene in (1,1).

Lösung 57





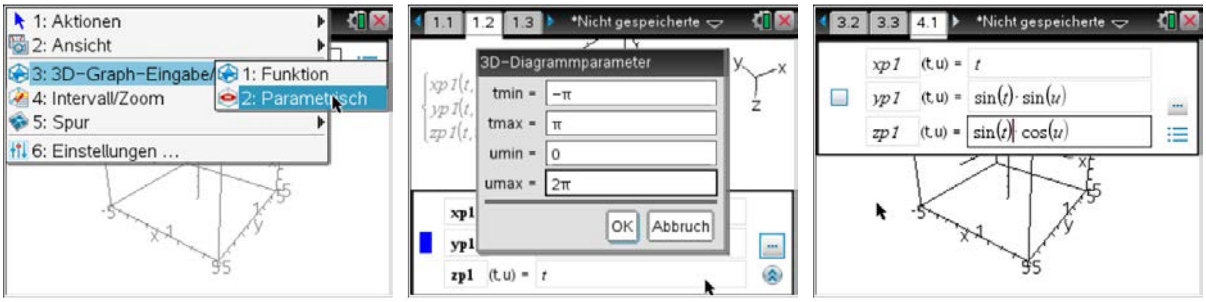


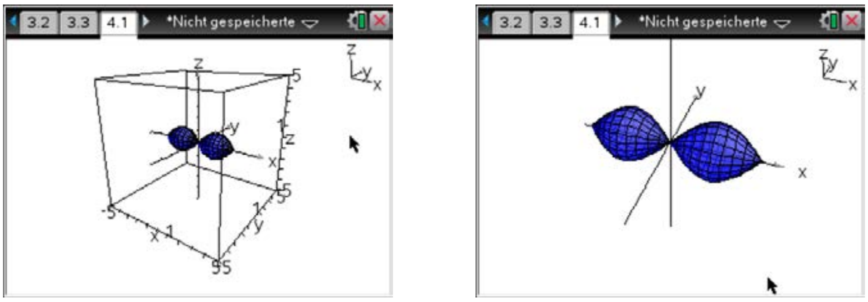
Aufgabe 58 *Darstellung einer Drehfläche in Parameterdarstellung*
Eine wichtige Anwendung der Integralrechnung ist die Berechnung des Volumens eines Körpers, der bei Drehung um eine Achse entsteht. Eine derartige Fläche soll nun dargestellt werden: die Sinusschwingung rotiert für $-\pi \le x \le +\pi$ um die x-Achse. Stelle den entstehenden Körper auf einer 3D-Seite dar.

Lösung 58 Wenn der Graph von f für $a \le x \le b$ um die x-Achse rotiert, dann kann der entstehende Körper in Parameterdarstellung beschrieben werden:

$$\begin{cases} t \\ f(t)\cdot\sin(u) \text{ , } a \le t \le b \text{ und } 0 \le u \le 2\pi. \\ f(t)\cdot\cos(u) \end{cases}$$

Bei der 3D-Eingabe wählen wir nun 2:Parametrisch und tragen die Parameterdarstellung wie oben gezeigt ein. Anschließend fixieren wir über den Button daneben die Parameterbereiche.



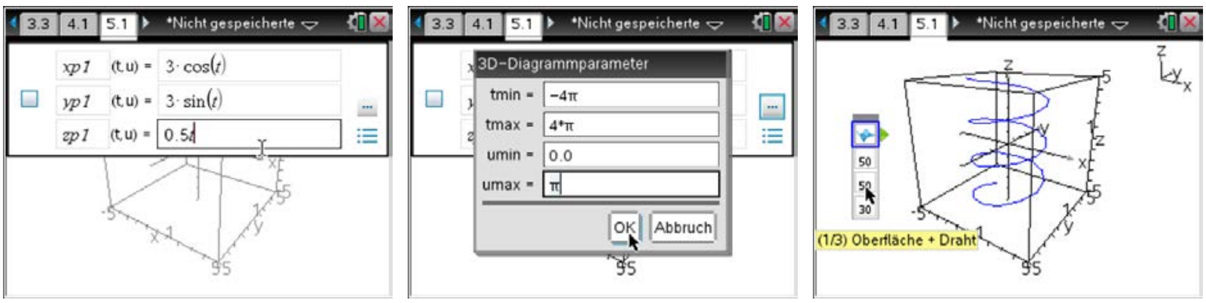


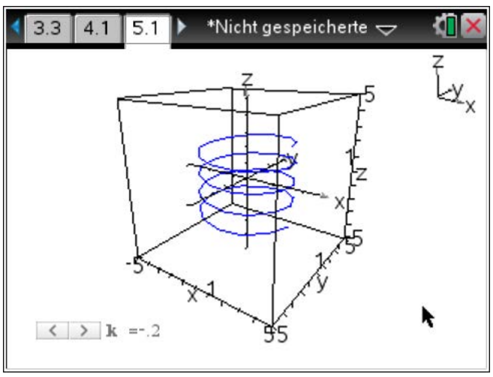
Auch hier können wir über die Attribute das Aussehen verändern, vergrößern oder verkleinern oder den Kasten bzw. die Achsen ausblenden, u.s.w.

Aufgabe 59 *Darstellung einer Raumkurve in Parameterdarstellung*
Eine Raumkurve hängt nur von einem Parameter ab. Denke an die Gleichung einer Geraden im Raum in Vektordarstellung. Wir wollen drei Windungen einer Schraublinie zeichnen.

Lösung 58

Die Parameterdarstellung einer Schraublinie lautet: $\begin{cases} 3\cdot\sin(t) \\ 3\cdot\cos(t) \\ 0{,}5\cdot t \end{cases}$ mit $0 \le t \le 2\pi$ für eine Windung.





Beachte die Attribute, sonst erhältst Du nur eine „eckige“ Schraublinie!

Anstelle von 0.5·t setzen wir nun k·t und führen für k einen Schieberregler ein, um den Einfluss von k auf die Gestalt der Schraublinie zu studieren. Lasse k auch negativ werden!

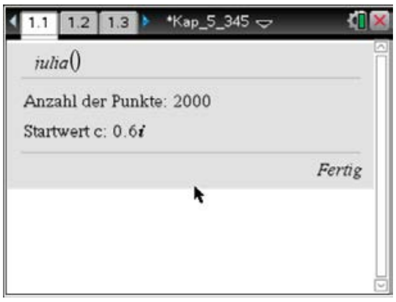
Welche Auswirkung auf die Kurve hat das?

5.6 Julia-Mengen

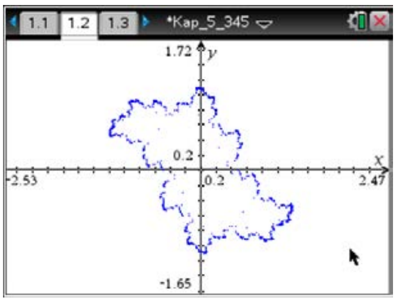
Über die so genannten „*Backtracking-Methode*“ [*] lassen sich mit einfachen Mitteln *Julia-Fraktale* erzeugen. Mit der Software-Version kann am PC sogar eine sehr große Anzahl von Punkten erzeugt werden. Auf dem Handheld scheint es geraten, nicht mehr als 1000 Punkte zeichnen zu lassen.

Hier folgt das Programm:

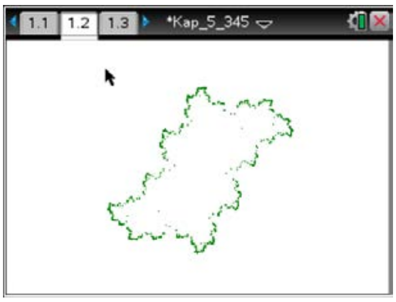
```
Define julia()=
Prgm
:Local c,z,k,n
:Request "Anzahl der Punkte:",n
:Request "Startwert c =",c
:lx:={}:ly:={}
:z:=0
:For k,1,n/2
:z:=(⁻1)^(randInt(0,1))*√(z−c)
:lx:=augment(lx,{real(z),⁻real(z)})
:ly:=augment(ly,{imag(z),⁻imag(z)})
:EndFor
:EndPrgm
```



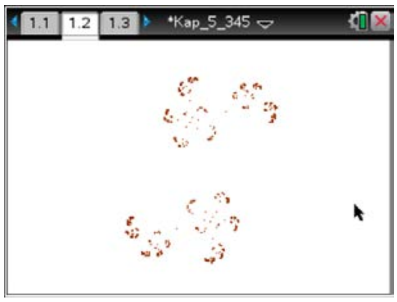
In den Dokumenteinstellungen muss das Kartesische Format und der Berechnungsmodus Approximiert eingestellt sein.



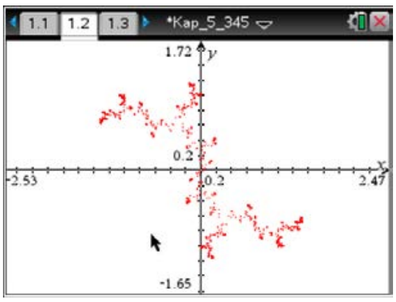
c = 0,6i



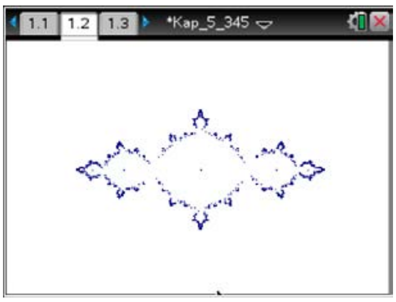
c = -0,6i



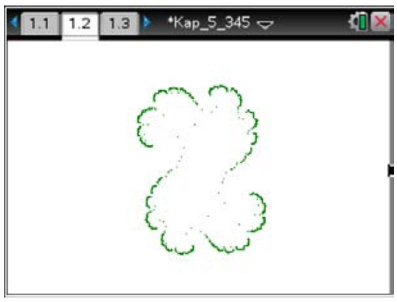
c = 0,5 – 0,5i



$c = i$



c = -1



c = 0,36 + 0,1i

Die Graphen wurden am PC hergestellt. Sie werden aber auch mit 1000 Punkten recht schön.

[*] Didier Deses, *Fractalen met de TI-84*, T3-Vlaanderen, http://www.t3vlaanderen.be/fileadmin/media/cahiers/pdf/cahier17.pdf

5.7 Programme – Funktionen

Wie schon im Abschnitt über das Programmieren angekündigt, wird hier kurz auf die wichtigsten Unterschiede zwischen selbst erzeugten Programmen und Funktionen eingegangen. Ein Beispiel – allerdings nicht aus dem Gebiet der Analysis – wird sowohl mit einem Programm als auch mit einer Funktion behandelt.

Eine Funktion erzeugt – wie erwartet – einen Funktionswert, der ausgegeben wird, der aber auch direkt von anderen Funktionen und auch Programmen übernommen werden kann. Bei einem Programm muss eine eventuelle Ausgabe über die disp-Anweisung erzwungen werden. Die erzeugten Werte können in (globalen) Variablen gespeichert werden. Diese Variablen muss man kennen, um sie weiter verwenden zu können. Denke an die Listennamen in unseren Programmierbeispielen. Entsprechende Hinweise sind in disp-Anweisungen sehr nützlich.

Funktionen kennen keine globalen Variablen. Die Eingabe der Parameter muss in Form von Funktionsargumenten erfolgen, da die request-Anweisung in Funktionen nicht anwendbar ist. Es gibt auch einige andere Anweisungen, die in Funktionen nicht verwendet werden können. Auf Details wollen wir hier nicht eingehen.

Aufgabe 60 *Teiler einer ganzen Zahl*
Schreibe zuerst ein Programm zur Ausgabe der Teilerliste einer beliebigen ganzen Zahl. Schreibe dann auch eine Funktion, die dasselbe leistet. Lasse dir dann für beide Fälle die Anzahl der Teiler und die Summe aller Teiler ausgeben.

Lösung 60 Für Teilbarkeitsprobleme ist die modulo-Funktion wie geschaffen. mod(a,b) liefert den ganzzahligen Rest der Division a:b. So ist z.B mod(35,4) = 3 und mod(110,15) = 5. Die Zahl x ist dann durch y restlos teilbar, wenn mod(x, y) = 0.

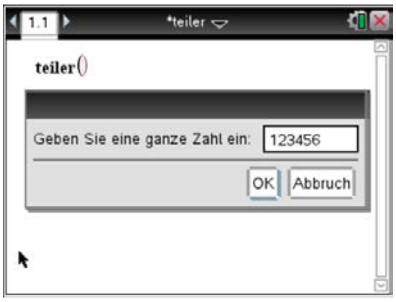
Das Programm teiler()

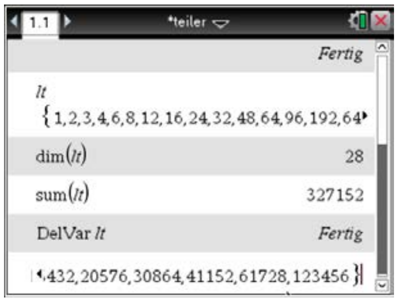
```
teiler
Define teiler()=
Prgm
Local z,k
Request "Geben Sie eine ganze Zahl ein: ",z
© lt is eine globale Variable
lt:={ }
For k,1,z/2
If mod(z,k)=0: lt:=augment(lt,{k})
EndFor
© der triviale Teiler wird angefügt
lt:=augment(lt,{z})
Disp "Teiler sind: ",lt
Disp "gespeichert in lt"
EndPrgm
```

Die Funktion divisors(Zahl)

```
divisors
Define divisors(z)=
Func
Local z,k,lt
© es gibt keine globalen Variablen
lt:={ }
For k,1,z/2
If mod(z,k)=0: lt:=augment(lt,{k})
EndFor
© der triviale Teiler wird angefügt
lt:=augment(lt,{z})
Return lt
EndFunc
```

Hier siehst du zuerst einen Programmlauf. Um die Teilerliste ansprechen zu können, musst du wissen, dass sie unter dem Namen *lt* gespeichert ist. Erst dann kannst du mit dem Ergebnis weiter arbeiten.





Wir löschen die Variable *lt* und lassen nun die Funktion divisors laufen. Der Benützer muss wissen, dass er die Zahl als Funktionsargument eingeben muss. Das „*Fertig*“ ist immer die letzte Ausgabe eines Programms. Wenn eine Funktion mehrere Argumente verlangt – was sehr häufig vorkommt, muss die Reihenfolge der Argumente bekannt sein.

Eine Funktion hingegen gibt einen Funktionswert aus. Die Funktion kann als Ganzes gleichsam für den Funktionswert in eine andere Funktion (TI-Nspire™ CX CAS - oder selbst geschrieben) oder auch in ein Programm übernommen werden.