

Folgen in GeoGebra

Im File **Folgen.ggb** ist die **Zahlenfolge** $a(k) = k^2/2^k$ und ein **Schieberegler** für **n** von **1 bis 10** gegeben. Öffne diese Datei, gib in der Eingabezeile den Befehl Folge ein und wähle

Folge[<Ausdruck>, <Variable>, <Startwert>, <Endwert>, <Schrittweite>]

Erkunde diesen Befehl z.B. durch folgende Eingaben:

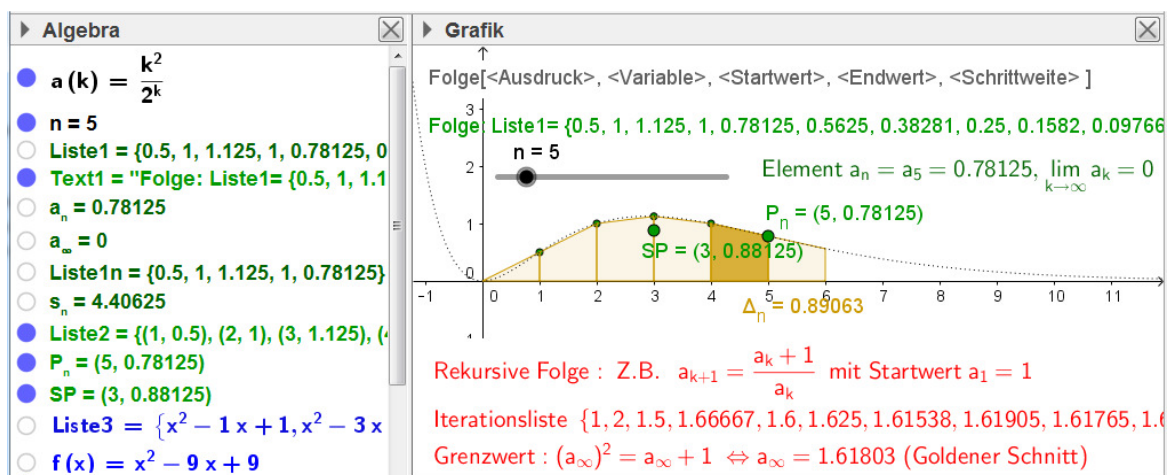
- Folge[a(k), k, 1, 10]** liefert in einer **Liste1** die ersten 10 Werte der **Zahlenfolge** $a(k)$, die als **Text[Liste1]** im Zeichenfenster ausgegeben werden kann. Teste auch den Befehl **TabellenText[Liste1, „v“]** oder „h“ statt „v“. Der Befehl **Element[Liste1, n]** liefert das n-te Element a_n abhängig vom Schieberegler n und **Grenzwert[a(x), ∞]** den Grenzwert von $a(k)$ für $k \rightarrow \infty$.

Durch Eingabe von

`Element \; a_n=a_{[n]}=a_n, \lim_{k \to \infty} a_k=a_\infty`

im Texteditor, kann man die Werte a_n in Abhängigkeit von n als LaTeX-Formel ausgeben.

- Summe[Liste1]** liefert die Summe der Elemente und damit die Partialsumme s_{10}
- Folge[(k, a(k)), k, 1, n]** liefert in einer **Liste2** die ersten n Punkte der **Punkt-folge** $(k, a(k))$, die direkt im Zeichenfenster ausgegeben wird.
Element[Liste2, n] liefert den n-ten Punkt P_n dieser Folge.
Summe[Liste2, n]/n liefert den Schwerpunkt dieser n Punkte.
- Folge[x^2 - k x + k, k, 1, 10, 2]** liefert in der **Liste 3** die ersten 5 Graphen der **Funktionsschar** $f_k(x) = x^2 - kx + k$ mit dem **Scharparameter** k und **Schrittweite** 2 , die direkt in der Zeichenebene ausgegeben wird. Ändere die Schrittweite auf 0.5. Warum liefert **Element[Liste3, n]** nicht den Graphen von $f_k(x)$ zum Parameter $k = n$?
- Folge[Vieleck[(k, 0), (k + 1, 0), (k+1, a(k+1)), (k, a(k))], k, 0, n]** liefert in der **Liste4** die ersten 10 Dreiecke der **Folge von Dreiecken** mit den Ecken $(k,0)$, $(k+1,0)$ auf der x-Achse und P_k aus Liste2, die direkt im Zeichenfenster ausgegeben wird.
Element[Liste2, n] liefert das n-te Dreieck Δ_n dieser Folge und **Summe[Liste4]** die Trapezsumme $S \rightarrow$ numerische Integration.



- Iterationsliste[(x + 1) / x, 1, 10]** liefert die **rekursive Folge** $a_{k+1} = \frac{a_k + 1}{a_k}$ mit dem **Startwert** $a_1 = 1$, deren Grenzwert a_∞ (im Falle der Konvergenz) die Gleichung $(a_\infty)^2 = a_\infty + 1$ erfüllt (Goldener Schnitt). (doppelt rekursive Folgen mit JavaScript)

Ändere nun den Term für $a(k)$ durch Doppelklick auf die Funktion oder generiere $N=100$ Folgenglieder. Wie sind entsprechend die Eigenschaften des Schiebereglers für n abzuändern. Spiele auch mit der Schrittweite.

Betrachte auch die GeoGebra-Figuren **Sunflower**, **Reihe** und **Fibonacci**.

Beispiele zu GeoGebra-Skript

1. GeoGebra-File Skript-SetzeWert

Betrachtet man die Exponentialfunktion a^x mit allgemeinem $a > 0$, so stellt man fest, dass diese für $a \approx 2.72$ mit ihrer Ableitung übereinstimmt. Am Schieberegler kann man aber den exakten Wert von $e = 2.71828\dots$ nicht einstellen, da man a nur pixelweise verschieben kann. Mit Hilfe einer **Schaltfläche** und **GeoGebra-Skript** kann man den Wert a exakt auf e setzen mit dem Skript-Befehl: **SetzeWert[<Objekt>, <Objekt>]** hier **SetzeWert[a , e]**.

Für $0 < a < 1$ ist der Graph monoton fallend und man vermutet, dass es jeweils einen Graphen für $a > 1$ gibt, der zu dem mit $0 < a < 1$ bezüglich der y -Achse symmetrisch ist.

Um dieses Paar a^x und b^x zu finden, kann man den Graphen $g(x)=b^x$ mit einem **Kontrollkästchen** (Name ww) sichtbar machen und setzt beim Anklicken von ww mit Hilfe von **GeoGebra-Skript** den Wert von b auf a mit dem Befehl: **SetzeWert[b , a]**

Schaltfläche
 Klick in die Zeichenfläche, um eine

Setze $a=e$ über GeoGebraSkript
 siehe Eigenschaften/Skripting/Bei Mausklick
 von Schaltfläche1 den Befehl: **SetzeWert[a,e]**
 Eingabe von e über Copy&Paste aus der Eingabezeile

$f(x) = 2^x$
 $f'(x) = 2^x \ln(2)$

Setze $a = e$

$a = 2$

Behalte Wert

Um den Graph von f zu duplizieren und den Wert a festzuhalten, gibt man eine Funktion $g(x)=b^x$ ein, die man mit einem Kontrollkästchen c sichtbar macht und dabei den Wert von b auf a setzt. Siehe Eigenschaften/Skripting vom Wahrheitswert/Kontrollkästchen c den Befehl: **SetzeWert[b,a]**

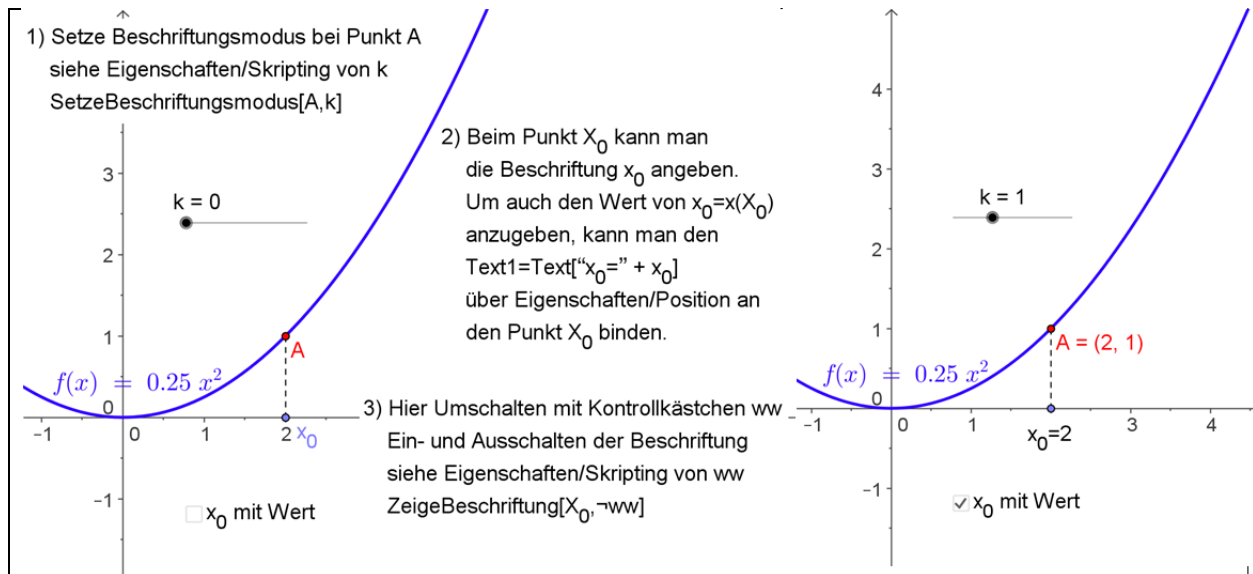
Eingabe:

Öffne die Datei **Skript-SetzeWert.ggb** und füge die Schaltfläche und das Kontrollkästchen ein, vgl. auch **Skript-SetzeWert-Lsg.ggb**.

2. GeoGebra-File Skript-Beschriftung

Bei GeoGebra kann man unter Eigenschaften -> Grundeinstellungen -> Beschriftung anzeigen wählen, ob beim Objekt nur der **Name**, **Name und Wert**, nur der **Wert** oder die selbst gewählte **Beschriftung** erscheint oder die Beschriftung ausgeblendet werden soll.

Dies kann man mit den Skript-Befehlen: **SetzeBeschriftungsmodus[<Objekt>, <Zahl>]** und **ZeigeBeschriftung[<Objekt>, <Wahrheitswert>]** mit Hilfe von Schaltflächen, Kontrollkästchen aber auch durch Anklicken von Objekten ändern.



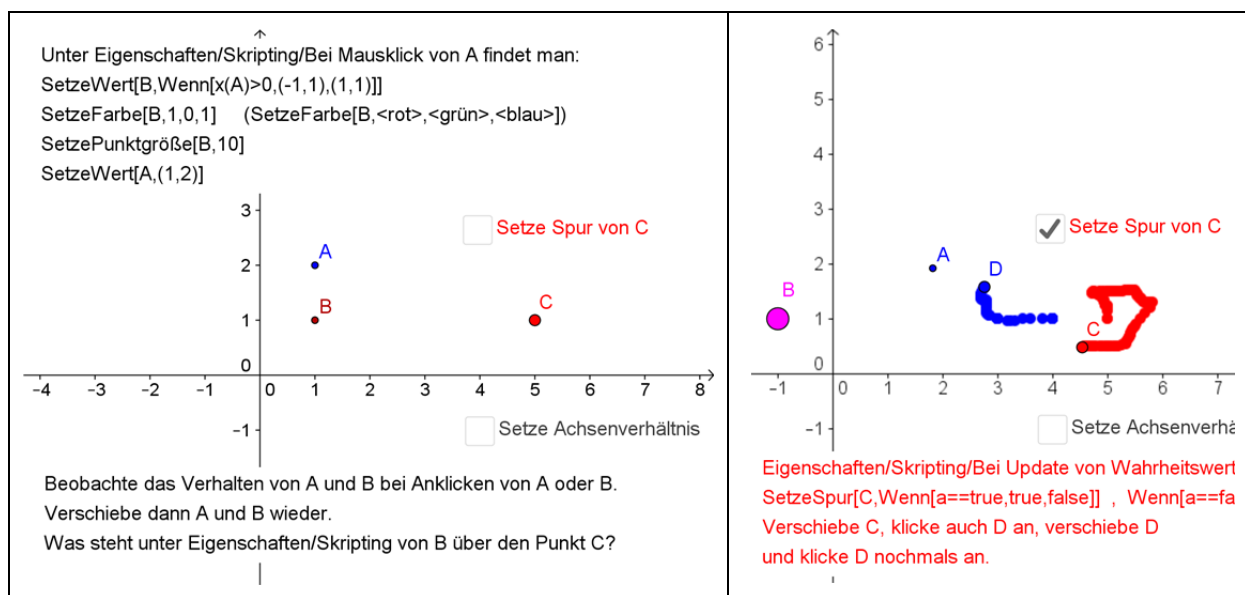
Dazu muss man diese Befehle unter Eigenschaften -> Skripting -> Bei Update bzw. bei Mausklick eintragen. Welcher Modus bei $k = 0, 1, 2, 3$ erscheint zeigt das File **Skript-Beschriftung-Lsg.ggb**. Mit **SetzeBeschriftung[<Objekt>, <Text>]** kann man auch die Beschriftung eines Objektes auf Knopfdruck ändern, vgl. z.B. Hoehenschnittpunkt-Lsg.ggb.

Obige Figur zeigt, wie man die Stelle x_0 auf der x-Achse so generiert, dass x_0 auf dieser verschiebbar ist aber x_0 gleichzeitig den Wert als Zahl angibt. Dazu wird der Punkt X_0 auf der x-Achse gewählt, dessen Beschriftung man auf x_0 setzen kann. Dann ist der Wert der Stelle durch $x_0 = x(X_0)$ gegeben. Leider kann man diesen Wert noch nicht mit der Beschriftung des Punktes X_0 ausgeben. Daher wird der **Text1=Text[„ $x_0 =$ „ + x_0]** an X_0 gebunden.

Öffne die Datei **Skript-Beschriftung.ggb**, setze die Beschriftung von A abhängig von k und trage für x_0 den Text mit dem Wert ein.

3. GeoGebra-File Skript-Befehle

Beobachte das Verhalten von A und B beim Anklicken von A und B, Klicke „Setze Spur von C“ an, verschiebe C, klicke auch D an, verschiebe D und klicke D wieder an. Betrachte die Einträge der Objekte unter **Eigenschaften -> Skripting -> Bei Mausklick** bzw. **Bei Update**.



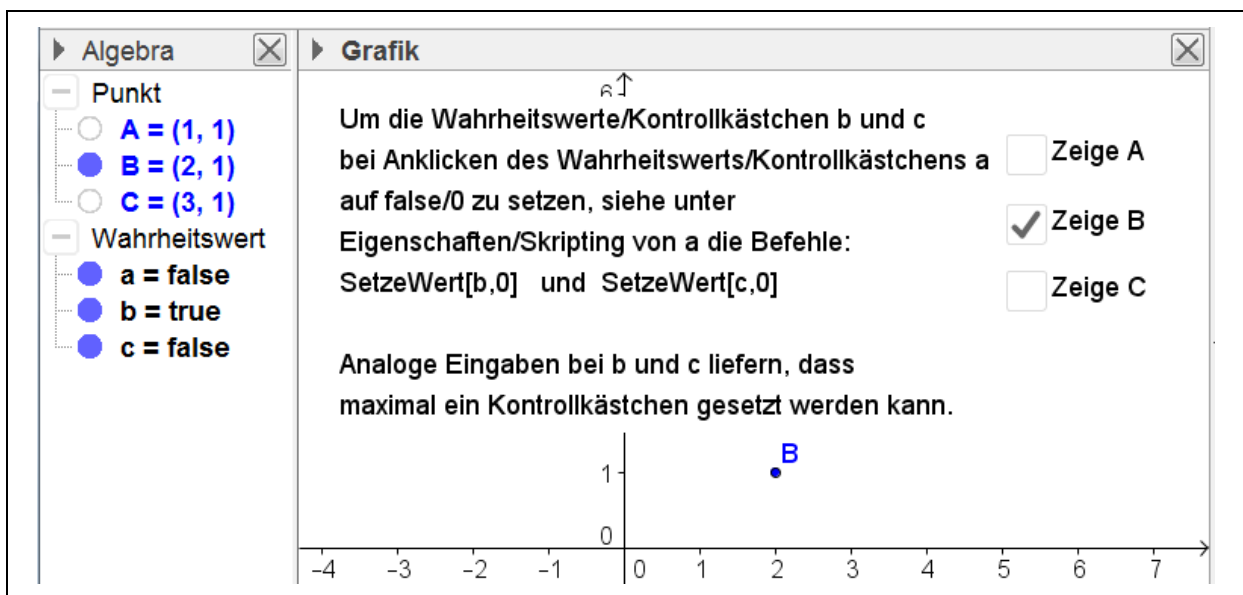
In dieser Figur wurden zu den oben beschriebenen folgende Skript-Befehle neu verwendet:
SetzeFarbe[<Objekt>, <Rot>, <Grün>, <Blau>] z.B. bei A: **SetzeFarbe[B,1,0,1]**

SetzePunktgröße[<Objekt>, <Zahl>] z.B. bei A: **SetzePunktgröße[B,10]**
SetzeSpur[<Objekt>, <Wahrheitswert true|false>] z.B. bei Kontrollkästchen a
SetzeSpur[C,Wenn[a==true,true,false]] oder kurz **SetzeSpur[C,a]**
ZoomIn[<Skalierungsfaktor>] z.B. bei Kontrollkästchen a **Wenn[a==false,ZoomIn[1]]**
 zum Löschen der Spur (Ansichten Auffrischen).
SetzeAchsenverhältnis[<Zahl>, <Zahl>] z.B. bei Kontrollkästchen b
Wenn[b==true,SetzeAchsenverhältnis[1,2],SetzeAchsenverhältnis[1,1]]

Teste weitere Skript-Befehle über die Eingabehilfe (Button rechts unten) unter alle Befehle, die mit Setze... bzw. Zeige... beginnen.

4. GeoGebra-File Skript-Kontrollkästchen

Öffnet man das File Skript-Kontrollkästchen, so kann man immer nur genau ein Kontrollkästchen anklicken. Das vorher angeklickte wird automatisch auf false gesetzt.



Betrachte die Einträge der Kontrollkästchen a, b, c unter Eigenschaften -> Skripting.

Wie müssen die Einträge bei den Objekt-Eigenschaften unter Skripting bzw. Erweitert geändert werden, damit A, B, C nur der Reihe nach, also zuerst A, dann erst B und dann erst C mit den Kontrollkästchen sichtbar gemacht werden können. Auch die Kontrollkästchen b und c sollen der Reihe nach eingblendet werden.

Vorzeichen

<p>Um bei Texten mit Ausgabe von Werten Ausdrücke wie „$x+2$“ zu vermeiden, ersetzt man im Texteditor bei Ausgabe des Objektes a den Eintrag „+ Objekt a“ durch</p> <p>Wenn[$a < 0$, „-“ + $(\text{abs}(a))$, „+“ + a]</p> <p>Zum Textstring in Anführungszeichen werden Werte mit + hinzugefügt, siehe Text2 im File Vorzeichen.ggb.</p>	<p>The screenshot shows the GeoGebra interface with two panels: 'Algebra' and 'Grafik'. In the 'Algebra' panel, there is a variable 'a' with a value of -2. In the 'Grafik' panel, there is a coordinate system with x and y axes ranging from 0 to 6. There are two text objects: 'Text1 = x+a = x + -2' and 'Text2 = x+a = x - 2'. Below them is a slider for 'a' with a value of -2. Below the slider, there is a text box with the following content: 'Wenn[a < 0 , „-“ + (abs(a)) , „+“ + a]'.</p>
---	--

Erste eigene Schritte mit CindyScript:

1. Übung zur Geometrie

- Öffne Cinderella.
- Setze nach Anklicken des Buttons „Punkt setzen“ vier Punkte A, B, C und D.
- Markiere unter „Elemente Bewegen“ den Punkt D mit der rechten Maustaste, wähle „Informationen einblenden“ und klicke dich durch die verschiedenen Menüs der Elementeigenschaften (Inspector) und ändere die Bezeichnung des Punktes D in S.
- Öffne in der Menüleiste unter „**Scripting** -> Skripte erstellen“ den CindyScript-Editor und klicke „Zeichnen“ (Draw) an.
- Klicke in das Fenster und gib **S.xy=(A+B+C)/3;** ein.
- Klicke die **Räder** rechts oben an und betrachte das Ergebnis im Cinderella-Zeichenfenster. Beim Verschieben der Punkte A, B oder C bleibt S stets der Schwerpunkt des Dreiecks ABC, vgl. **Beispiel-Teil-1-Schwerpunkt.cdy**.
- Lösche nun die Punkt B, C und S durch Markieren (Mehrfachselektion mit gedrückter Shift-Taste) und Drücken des Buttons mit dem Radiergummi.
- Gib im CindyScript-Editor **A.color=(0,0,0);** ein. Dann hat A die Farbe schwarz. (Rot-Grün-Blau-Farbcode). Spiele mit anderen Farbgebungen.
- Gib **if(A.x <= 0 , A.color=(0,0,0) , A.color=(1,1,1));** ein. Dann wechselt A die Farbe, wenn man ihn über die y-Achse schiebt, d.h. A.x das Vorzeichen wechselt. Wie erhält man A im 1. und 3. Quadranten weiß und im 2. und 4. Quadranten schwarz ?
- Gib **Draw(A+(1,1) , color->(0,1,0));** ein, so erscheint rechts oberhalb von A ein grüner Punkt, der sich mit A bewegt, aber nur ein reines „Layout-Objekt“ ist, mit dem man nicht weiter konstruieren kann.
- Um gleich mehrere Punkt (z.B. n Punkte) hinzuzufügen, erzeuge im Zeichenfenster einen Punkt mit Namen N und gib im CindyScript-Editor **N.x=round(N.x) ; N.y=0 ;** ein. Dann lässt sich N nur noch ganzzahlig auf der x-Achse bewegen (**Schieberegler N**)
- Lösche obigen Draw-Befehl und gib nach $N.x=...$ folgendes ein:

```
n=N.x ;  
repeat( n , i , draw( A+ i*(1,1) , color->hue(i/10) ); );
```

Abhängig von der Lage N erscheint eine Reihe von Punkten in Regenbogenfarben, die sich mit A bewegt. Wer diese Punkte auf einem Kreis oder einer Spirale anordnen will, kann **i*(1,1)** ersetzen z.B. durch **(cos(i/10),sin(i/10))** oder **sqrt(i)*(cos(i/10),sin(i/10))**. Vergleiche dein bisheriges Script **Beispiel-Teil-2-Punktfolge.cdy** mit dem Script der Figur **Sunflower.cdy**.
- Punkte können als Tupel mit runden **oder** mit eckigen Klammern gesetzt werden.
- Eine Online Dokumentation der Befehle von CindyScript findet man im Menü unter „Hilfe“ im Reference Guide unter CindyScript. Der „Operator Index“ steht dort unten.
- Abfrage von Werten und Attributen durch **err(<Objekt>);** oder **err(inspect(<Objekt>));** Einigen Attributen sind Wahrheitswerte true/false zugeordnet, anderen Zahlen, Tupel, Tripel, vgl. Online Dokumentation. Nicht alle Attribute sind direkt verfügbar.

2. Übung zur Analysis

- Öffne Cinderella und unter „Scripting -> Skripte erstellen“ den CindyScript-Editor und klicke „Zeichnen“ (Draw) an.
- Definiere eine Funktion **f(x) := 0.25*x^2;** oder eine andere Funktion in x.
- Plote die Funktion mit **plot(f(x), start->-5, stop->5, steps->100, size->3);**
- Gib im Zeichenfenster einen Punkt A ein und setze diesen mit CindyScript auf den Graphen von f durch **A.y=f(A.x);**
- Erzeuge eine Wertetabelle mit **werte=[];** (setzt Liste auf leere Menge)

```
repeat( 10 , i , x= -6+i ; werte=werte++[(x,f(x))] );
```


- Ausgabe mit `drawtable((10,0), werte, width->35, align->"mid");` an Position (10,0). Man kann statt der festen Koordinaten einen frei beweglichen Punkt angeben.
- Ausgabe der Punkte auf dem Graphen mit `apply(werte, draw(#));`
- Ausgabe der Tangente im Punkt A mit `draw(tangent(f(#), A.x), color->(0,0.8,0));`
- Berechnung der Ableitungsfunktion mit `g(x):=d(f(#),x); plot(g(x));`
 Wähle nun eine andere Funktion, vgl. **Beispiel-Teil-3-Analysis.cdy**.

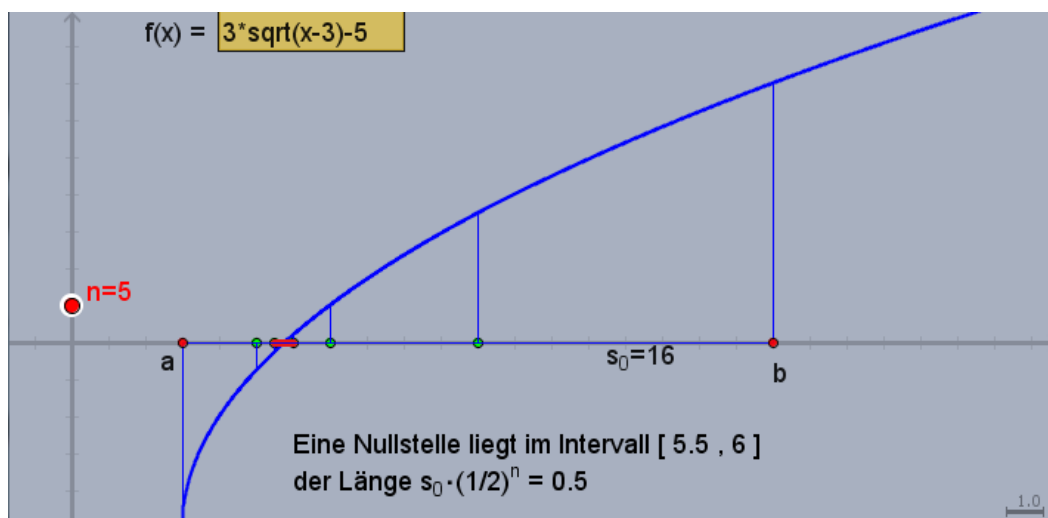
3. Beispiel zur Musik

Töne kann man mit dem Befehl `playtone(_)`; erzeugen, sofern eine Sounddatenbank von Java verfügbar ist. Die Tonleiter erhält man z.B. mit:

```
repeat( 30 , i , playtone(51+i); wait(100); ); //Millisekunden
```

Weitere Beispiele siehe, **Beispiel-Teil-4-Musik.cdy**.

4. Nullstellensuche mit Bisektions-Verfahren, siehe File.: Bisektion.cdy



CindyScript unter „Zeichnen“:

<code>f(x):=parse(Text0.val);</code>	Übernimmt den Text aus Eingabefenster im Applet
<code>plot(f(x),size->2);</code>	einfacher Funktionsplot mit Strichstärke 2
<code>N.x=0; N.y=round(N.y); if(N.y<-4,N.y=-4);</code>	Definiert einen einfachen Schieberegler für die Zahl n
<code>n=N.y+4;</code>	
<code>Drawtext(N+(0.4,0.1),"n="+n, size->18, color->(1,0,0));</code>	Gibt Wert n aus
<code>a.y=0;b.y=0;</code>	Gibt Strecke von $a=(a.x,a.y)$
<code>Draw([a,[a.x,f(a.x)]],size->1);Draw([b,[b.x,f(b.x)]],size->1);</code>	nach $(a.x,f(ax))$ aus
<code>if(f(a.x)*f(b.x)>0,</code>	Prüft, ob Voraussetzung für Bisektion erfüllt ist
<code>Drawtext((2,4),"Intervall [a,b] nicht geeignet !",size->18)</code>	Ausgabe der Fehlermeldung
<code>,</code>	Komma trennt „Then“ und „Else“
<code>ai=a.xy; bi=b.xy;</code>	Setzt Endpunkte a_i, b_i für Strecke $[a,b]$ zu Beginn der Bisektion (Startwerte)
<code>repeat(n,i,</code>	Repeat-Schleife mit n Schritten, Laufvariable i
<code>ci=0.5*(ai+bi);</code>	Berechnet Mittelpunkt von a_i und b_i
<code>Draw(ci);Draw([ci,[ci.x,f(ci.x)]],size->1);</code>	Gibt Punkt c_i und zugehörige Ordinate aus
<code>if(f(ai.x)*f(ci.x)<=0, bi=ci, ai=ci);</code>	Wählt Teilstrecke $[a_i,b_i]$ oder $[b_i,c_i]$ als „neue“
<code>);</code>	Strecke $[a_i,b_i]$
<code>draw([ai,bi],color->[0.9,0.1,0.1],size->5);</code>	Ausgabe der Strecke $[a_i,b_i]$
<code>drawtext((6,-3),"Eine Nullstelle liegt im Intervall ["+ai.x+ " , "+bi.x+"],size->18);</code>	
<code>drawtext((6,-4),"der Länge $s_0 \cdot (1/2)^n =$ "+(abs(bi-ai)),size->18);</code>	
<code>);</code>	Ergebnis-Ausgabe mit TeX-Text „“ und Wert +Variable

Betrachte auch die Figuren **Konvexe Hülle**, **Nachziehen** oder **Schieberegler**.