

## SYNTAX ERROR: ARITHMETIK UND DEZIMALPUNKT

Besonders zu Beginn der Arbeit mit einem grafikfähigen Taschenrechner kann es zu einer irrtümlichen Verwendung des Kommas statt des Dezimalpunkts kommen.

### Komma statt Dezimalpunkt

Nahezu alle grafikfähigen Taschenrechner besitzen eine eigene Taste sowohl für den Dezimalpunkt, als auch für das Komma. Dezimalbrüche sind aber immer mit dem Dezimalpunkt zu schreiben!

Wird irrtümlich statt des Dezimalpunktes in einem numerischen Ausdruck das Komma verwendet, so zeigt der TI-84 nach der Auswertung mittels der **[ENTER]**-Taste einen Bildschirm, auf dem in der ersten Zeile der Fehlertyp angegeben ist. Hier **ERR: SYNTAX**, also ein Fehler im Aufbau der Zeichenreihe.

Nach der Beschreibung des Fehlertyps folgen meistens zwei Optionen, die dann entweder durch Markieren und **[ENTER]**, oder über die Schnelleingabe der Ziffer auszuführen sind. **Quit** bedeutet dabei, dass der TI-84 die Berechnung abbricht und zum **HOME**-Screen zurückkehrt. Jetzt könnte der Ausdruck neu eingegeben werden.

Die zweite Option **Goto**, die hier zur Verdeutlichung markiert wurde, erlaubt es nach ihrer Ausführung...

... den fehlerbehafteten Ausdruck zu kontrollieren und zu verändern. Dabei setzt der TI-84 den Cursor an eine Position, die entweder der Fehlerstelle entspricht, oder zumindest in der Nähe dieser Stelle lokalisiert ist.

Somit kann man durch einfaches Editieren die falsche Eingabe korrigieren und den gewünschten Ausdruck mit seinem Wert erzeugen.

(3,5+2.9)\*-3

ERR: SYNTAX  
1:Quit  
2:Goto

(3,5+2.9)\*-3

ERR: SYNTAX  
1:Quit  
2:Goto

(3,5+2.9)\*-3

(3.5+2.9)\*-3  
-19.2

## SYNTAX ERROR: ARITHMETIK UND VORZEICHEN

Ungewohnt, weil in der Schreibweise per Hand nicht unterscheidbar, ist der Gebrauch eines eigenen Vorzeichens für negative Zahlen.

### Minuszeichen statt Vorzeichen

Die meisten numerischen Taschenrechner kennzeichnen negative Zahlen über eine eigene Vorzeichen-Taste. Wird dieses Vorzeichen  $\ominus$  mit dem Zeichen für die Subtraktion  $\omin�$  verwechselt...

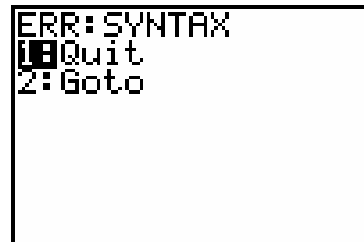
...so liefert der TI-84 wieder eine **SYNTAX**-Fehlermeldung. Damit erhalten wir hier also lediglich den Hinweis, dass mit der Reihenfolge und Anordnung der eingegebenen Zeichen etwas nicht in Ordnung ist.

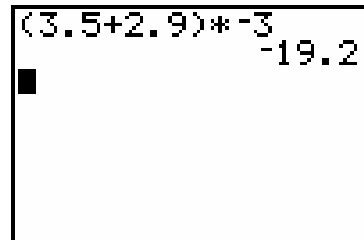
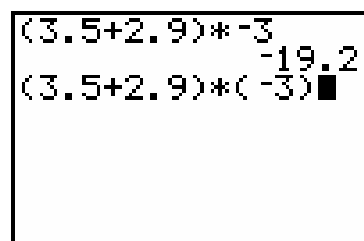
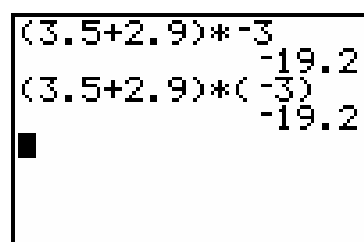
Benutzt man in diesem Fall den Befehl **Goto**, so setzt der TI-84 den Cursor direkt auf das  $\omin�$ -Zeichen. Man muss sich jedoch darüber im Klaren sein, dass es sich hier nicht um eine eindeutige Fehlerdiagnose handelt, sondern nur um einen Hinweis auf die mögliche Fehlerquelle...

...denn durch die Einführung des Vorzeichens an dieser Stelle ergibt sich zwar ein syntaktisch korrekter Term, wie die Auswertung zeigt. Ob es sich aber um den Term der Aufgabenstellung handelt, das vermag natürlich einzig und allein der Benutzer zu beurteilen. Man hätte auch die Multiplikation oder das  $\omin�$ -Zeichen löschen können, um einen korrekten Ausdruck zu erhalten, dann jedoch mit völlig unterschiedlichen Bedeutungen.

Im Übrigen zeigt sich an der Multiplikation mit einer negativen Zahl noch eine weitere Besonderheit: Was wir in unserer Handschreibweise in jedem Fall mit Klammern notieren müssen...

...versteht der TI-84 dank des unterschiedlichen Zeichens (man vergleiche die Länge der beiden Minus-Zeichen in Bildschirm 1 und 4) auch so. Dies beweist die „traditionelle“ Eingabe.

## SYNTAX ERROR: ARITHMETIK UND KLAMMERSETZUNG

Eine häufige Fehlerquelle ist die nichtkorrekte Klammersetzung. Beim TI-84 muss für jede schließende Klammer auch eine öffnende Klammer vorhanden sein.

### Mehr schließende Klammern als öffnende Klammern

Wird in einem numerischen Ausdruck eine schließende Klammer (oder auch mehrere) zu viel eingetippt und ist damit die Anzahl der schließenden Klammern größer als die Anzahl der öffnenden Klammern...



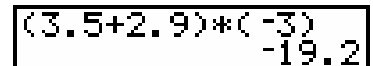
...so zeigt der TI-84 nach der Auswertung mittels **ENTER**-Taste wieder den Bildschirm mit der Fehlermeldung **ERR: SYNTAX**. Also auch in diesem Fall geht es um eine nicht korrekte Zeichensetzung...



...die der TI-84 nach dem Befehl **Goto** auch dokumentiert: Er setzt den Cursor auf die überzählige schließende Klammer. Es gibt nun zwei Möglichkeiten, den numerischen Ausdruck zu korrigieren.



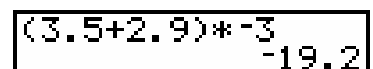
Entweder, man fügt eine zusätzliche öffnende Klammer ein und erhält somit das gewünschte Ergebnis. Dies entspricht der traditionellen Schreibweise.



Oder es wird die zusätzliche Klammer schlicht und einfach gelöscht. Denn mit der eigenen Vorzeichen-Taste **(+/-)** ist auch hiermit sichergestellt, dass eine korrekte Reihenfolge von Zeichen eingegeben ist...



...und somit der TI-84 auch in diesem Fall zum gewünschten Ergebnis gelangt.



## FEHLER IN DER KLAMMERSETZUNG

Häufig treten falsche Ergebnisse auf, die nicht als Fehler kenntlich gemacht werden, da die entsprechenden Eingaben durchaus einer gültigen Syntax entsprechen.

### Fehlende schließende Klammer

Speziell bei Berechnungen mithilfe der in den TI-84 eingebauten Funktionen können Ergebnisse resultieren, die so nicht erwünscht waren, ohne dass deutlich wird, dass hier ein Problem vorgelegen haben könnte.

Denn jeder Aufruf einer Funktion erfolgt generell mit einer öffnenden Klammer. Danach ist dann also das Argument (oder die Argumente) einzugeben und dann mittels einer schließenden Klammer kenntlich zu machen, dass an dieser Stelle die Auswertung der Funktion zu beenden ist.

Wird die schließende Klammer nicht eingetippt, was aus mangelnder Disziplin im Umgang mit Syntax oder schlicht durch Vergessen geschehen kann...

...so verwendet der TI-84 die gesamte noch folgende Zeichenkette als Argument der geöffneten Funktion. Dies dokumentiert die rechts dargestellte Vergleichsrechnung, in der die schließende Klammer nach der letzten Zahl eingegeben wurde. Als Ergebnis kommt hier der gleiche Zahlenwert heraus, wie oben ohne die schließende Klammer.

War es jedoch die Absicht, zunächst die Wurzel zu berechnen und dann zum Resultat den zweiten Wert zu addieren, so ist das nur mithilfe der schließenden Klammer direkt nach dem ersten Wert zu erreichen.

Der Unterschied zu den beiden vorherigen Eingaben wird an diesem Ergebnis sehr deutlich. Damit diese Probleme nicht auftreten, deren Ursache oft nicht einfach zu finden ist, hilft nur eins: **Syntaxdisziplin von Beginn an!**

## DIVIDE BY 0 ERROR: BRUCHTERME UND KLAMMERN

Eine nicht sauber ausgeführte Syntax in der Klammersetzung kann zu einer offenkundigen Fehlermeldung führen. Sie kann aber auch ohne „Absturz“ für unerwünschte Resultate sorgen.

### Teilen durch Null

Gerade durch die auf der vorherigen Seite angesprochene Eigenheit des TI-84, alles nach einer öffnenden Klammer automatisch zum Inhalt dieser Klammer zu machen, kann es zu einem Fehler kommen, der natürlich nicht beabsichtigt war...

...nämlich zum Teilen durch null (**ERR: DIVIDE BY 0**). Entgegen der Absicht wurde in der ersten Eingabe zunächst addiert und gleich danach subtrahiert, sodass eine Null resultiert. Erst als letzte Operation erfolgte dann die Division und damit zwangsläufig die entsprechende Fehlermeldung.

Zwingend muss die schließende Klammer eingegeben werden, um diese Art von „Fehlinterpretation“ zu vermeiden. Im Beispiel sollte erst addiert, dann geteilt und zum Schluss subtrahiert werden.

```
1/(1.5+0.5)-2
```

```
ERR:DIVIDE BY 0
1:Quit
2:Goto
```

```
1/(1.5+0.5)-2
-1.5
```

### Bruchterme, Klammern und unerwünschte Resultate

In diesem Zusammenhang ist auf ein weiteres Problem hinzuweisen, das bei der Berechnung von Bruchtermen auftreten kann. Wird eine schließende Klammer vergessen, so gibt der TI-84 dafür keine Fehlermeldung aus, sondern macht den gesamten Rest zum Inhalt der Klammer.

Im Beispiel liefert er also das gleiche Resultat, als wäre die erste öffnende Klammer gar nicht mit eingegeben worden. War es allerdings die Absicht, die Summe durch die Differenz zu teilen, so muss dies natürlich durch die Klammersetzung angezeigt werden...

...denn selbstredend erhält man somit ein anderes Ergebnis. Aber da der TI-84 keine Fehlermeldung bei fehlender schließender Klammer gibt, können durch Unachtsamkeiten im Umgang mit Klammern leicht falsche Gesamtergebnisse im Verlauf einer umfangreicheren Rechnung entstehen.

```
(1.5+4/(2.5-0.5)
```

```
3.5
```

```
(1.5+4/(2.5-0.5)
1.5+4/(2.5-0.5)
3.5
```

```
1.5+4/(2.5-0.5)
(1.5+4)/(2.5-0.5)
2.75
```

## DOMAIN ERROR: RUNDEN VON ZAHLEN

Unabhängig von der gewählten **MODE**-Einstellung kann der TI-84 nicht mehr als neun Ziffern nach dem Dezimalpunkt anzeigen.

### Zu viele Dezimalstellen

Beim **round**-Befehl muss neben dem zu rundenden Wert auch die Anzahl der Ziffern nach dem Komma als zweites Argument angegeben werden.

Wird hierfür ein zu großer Zahlenwert eingetragen (mehr als 9 Stellen nach dem Dezimalpunkt kann der TI-84 nicht anzeigen), so liefert er die Fehlermeldung **ERR: DOMAIN**. Die gewünschte Anzahl liegt nicht im Definitionsbereich für das zweite Argument des **round**-Befehls.

Mit dem Befehl **Goto** kann die Anzahl der Stellen nach dem Komma innerhalb der Rechengrenzen angepasst werden. Maximal sind also 9 Stellen anzugeben.

Ist im **MODE**-Menü die Anzahl der Stellen nach dem Dezimalpunkt auf einen geringeren als den maximalen Wert von 9 begrenzt...

...so führt die Verwendung einer größeren Zahl im **round**-Befehl allerdings nicht zu einer Fehlermeldung...

...sondern wird einfach nur ignoriert, solange sie nicht oberhalb der 9 liegt.

```
round( $\pi$ , 10) █
```

```
ERR: DOMAIN
Quit
2: Goto
```

```
round( $\pi$ , 9)
3.141592654
█
```

```
Normal Sci Eng
Float 01234 6789
Radian Degree
Func Par Pol Seq
Connected Dot
Sequential Simul
Real a+bi re^ $\theta$ i
Full Horiz G-T
```

```
round( $\pi$ , 7) █
```

```
round( $\pi$ , 7)
3.14159
█
```

## ARGUMENT ERROR: FUNKTIONEN UND ARGUMENTE

Im Umgang mit Funktionen ist nicht nur die Klammersetzung wichtig, sondern auch die Kenntnis der korrekten Anzahl von Argumenten.

### Funktionen und Dezimalpunkt

Obwohl hier die Syntax im Bereich der Klammersetzung sauber befolgt wurde, passiert eher durch Unachtsamkeit die falsche Eingabe eines Kommas an der Stelle, an der eigentlich der Dezimalpunkt erforderlich ist.

Der TI-84 antwortet darauf mit der Fehleranzeige **ERR: ARGUMENT**. Das bedeutet, entweder passt der Typ des Arguments nicht zur Funktion, oder die Anzahl der Argumente ist nicht korrekt. Letzteres trifft in diesem Beispiel zu, da das Komma vom TI-84 als Trennzeichen angesehen wird und er somit von zwei Argumenten für die Wurzel ausgeht...

...obwohl eigentlich nur eine Dezimalzahl eingegeben werden sollte.

### Funktionen und Argumentanzahl

Oben eher durch einen Tippfehler bedingt, kann der gleiche Fehlertyp aber auch durch eine nicht passende Anzahl von Argumenten verursacht werden. Obwohl wir den größten gemeinsamen Teiler auch von mehreren Zahlen bilden können...

...ist im TI-84 eine Version eingebaut, die sich lediglich auf zwei Zahlen bezieht. Konsequenterweise erfolgt an dieser Stelle somit die Fehlermeldung in Bezug auf das Argument...

...die sich aber durch eine entsprechende Korrektur in der Anzahl der Argumente beheben lässt.

```
√(2,25)■
```

```
ERR: ARGUMENT
1: Quit
2: Goto
```

```
√(2.25) 1.5
■
```

```
gcd(4, 14, 48)■
```

```
ERR: ARGUMENT
1: Quit
2: Goto
```

```
gcd(4, 14) 2
■
```

## NONREAL ANS ERROR: WURZELN UND NEGATIVE ZAHLEN

Beim Lösen einer quadratischen Gleichung mittels Variablen und Lösungsterm ist es notwendig, sehr sorgfältig auf das Vorzeichen der Variablen  $q$  zu achten.

### Nonreal Ans Error: Quadratwurzeln

Auf dem Bildschirm rechts ist ein positiver Lösungsterm dargestellt. Allerdings wurde hier irrtümlich der Wert der Variablen  $q$  als positiver Zahlenwert  $4$ , anstatt  $-4$  eingegeben.

3→P: 4→Q  

$$-P/2 + \sqrt{(P/2)^2 - Q}$$
 4

Die Auswertung des Terms unter der Wurzel liefert damit folglich eine negative Zahl, sodass der TI-84 mit der Fehlermeldung **ERR: NONREAL ANS** reagiert. Das bedeutet, entsprechend der Standardeinstellung **Real** im **MODE**-Menü, die wir hier voraussetzen, sagt der Rechner, dass es kein reelles Resultat gibt.

ERR:NONREAL ANS  
 1:Quit  
 2:Goto

Benutzt man hier die **Goto**-Option, so springt der Cursor in den Wurzelterm, denn hier tritt die negative Zahl auf. Allerdings liegt die Ursache des Fehlers nicht im Term selbst, der ist korrekt eingegeben...

$$-P/2 + \sqrt{(P/2)^2 - Q}$$

...sondern in der Zuweisung des Wertes  $4$  zur Variablen  $q$ . Man muss also zur Behebung des Fehlers die Wertzuweisung erneut vornehmen, diesmal mit dem korrekten Vorzeichen und dann den Lösungsterm noch einmal auswerten.

3→P: -4→Q  

$$-P/2 + \sqrt{(P/2)^2 - Q}$$
 -4  
 1

### Wurzeln aus negativen Zahlen

Bei uns ist wegen der Definition, dass ein Radikand größer oder gleich Null sein muss, die Berechnung einer Wurzel mit geradem oder ungeradem Wurzelexponenten aus einer negativen Zahl nicht zulässig. In Amerika und damit in den amerikanischen Rechnern ist dies aber...

$$\sqrt{-27}$$

...für ungerade Wurzelexponenten durchaus möglich und liefert ein Resultat, das wir hier als die korrekte Lösung der entsprechenden Gleichung dritten Grades interpretieren würden.

$$\sqrt[3]{-27}$$
 -3



## SYNTAX UND ARGUMENT ERROR: FUNKTIONEN UND SYNTAX

Im Umgang mit Funktionen, die mehrere Argumente erfordern, ist es notwendig, nicht nur die Anzahl, sondern auch den Typ der einzelnen Eingaben zu kennen.

### Syntax Error: Funktionen

Für die numerische Berechnung des minimalen Wertes einer Funktion in einem gegebenen Intervall erfordert der entsprechende Befehl **fMin** in der korrekten Reihenfolge: Term, Variable, untere Grenze, obere Grenze.

```
fMin(X^2+4X+3, -1
0, 10)■
```

Den Term wird man sicher nicht vergessen, aber die Festlegung der Variablen könnte „selbstverständlich“ sein und daher nicht vorgenommen werden. Für den TI-84 ist das natürlich nicht selbstverständlich. Folglich antwortet er mit der **SYNTAX**-Fehlermeldung...

```
ERR: SYNTAX
1: Quit
2: Goto
```

...die allerdings durch nachträgliches Einfügen der benötigten Variablen sofort wieder behoben werden kann.

```
fMin(X^2+4X+3, X,
-10, 10)
-1.999999806
■
```

### Argument Error: Funktionen

Wird statt der unabhängigen Variablen allerdings eine der beiden Grenzen des Intervalls vergessen, dann weiß der TI-84 natürlich nicht, wo er das Minimum suchen soll...

```
fMin(X^2+4X+3, X,
-10)■
```

```
ERR: ARGUMENT
1: Quit
2: Goto
```

...und gibt somit eine Fehlermeldung aus - hier **ERR: ARGUMENT**. Aber auch dieser Fehlertyp erlaubt mittels **Goto**...

```
fMin(X^2+4X+3, X,
-10, 10)
-1.999999806
■
```

...die sofortige Korrektur und liefert am Ende doch noch das gesuchte Ergebnis.

## NO SIGN CHANGE ERROR: NICHT VORHANDENE NULLSTELLEN

Bei der numerischen Berechnung von Nullstellen ist sehr sorgfältig auf die korrekte Eingabe der entsprechend umgeformten Gleichung zu achten.

### Keine Nullstelle vorhanden

Um mit dem **EQUATION SOLVER** Gleichungen numerisch zu lösen, ist es erst notwendig, diese auf die Berechnung einer Nullstelle umzuschreiben. Natürlich können dabei verschiedene Fehler unterlaufen, insbesondere Vorzeichenfehler.

```
EQUATION SOLVER
eqn: 0=X^2+1
```

Trägt man eine solche Gleichung in den **EQUATION SOLVER** ein...

```
X^2+1=0
X=0
bound=(-1E99, 1...
```

...und lässt den TI-84 nach einer Lösung suchen, so kann es passieren, dass gar keine Nullstelle in dem angegebenen Intervall existiert. Der TI-84 antwortet dann mit der Fehlermeldung **ERR: NO SIGN CHNG**, also kein Vorzeichenwechsel.

```
ERR:NO SIGN CHNG
1:Quit
2:Goto
```

Nach dem Ausführen von **Goto** kehrt der Rechner in den **SOLVER** zurück. Man kann sich dann also überlegen, ob es irrtümlich eine falsche Eingabe war oder ob die zu untersuchende Gleichung wirklich keine Lösung hat. Im Zweifelsfall hilft bei der Berechnung von Nullstellen eine grafische Kontrolle.

```
X^2+1=0
X=
bound=(-1E99, 1...
left-rt=1
```

In unserem Beispiel wurde fälschlicherweise ein  $\oplus$ -Zeichen eingegeben, sodass nach der entsprechenden Korrektur...

```
EQUATION SOLVER
eqn: 0=X^2-
```

...dann auch eine Lösung angezeigt wird.

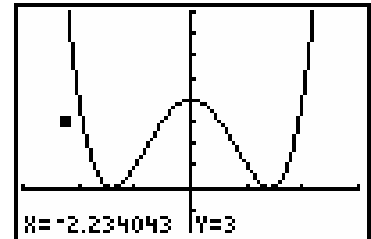
```
X^2-1=0
X=1
bound=(-1E99, 1...
left-rt=0
```

## ZOOM ERROR: BESTÄTIGUNG DER ZOOM BOX ECKEN

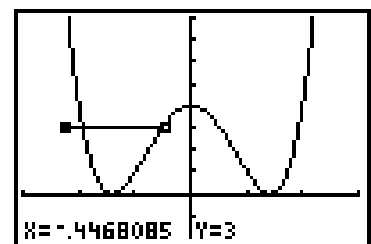
Soll ein Graf mit Hilfe des Boxzoomens näher untersucht werden, so ist darauf zu achten, dass mit dem Cursor wirklich eine Box aufgezogen wird und nicht nur ein Punkt oder eine Linie.

### Versuch die zweite Zoom Box Ecke zu bestätigen

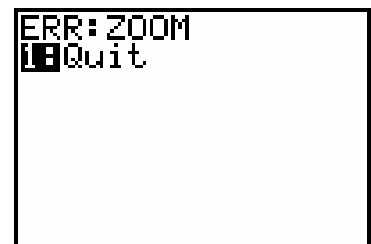
Im Bild ist bereits die erste Ecke der Zoom Box mittels **[ENTER]** bestätigt worden.



Als Nächstes hat man mit dem Cursor entweder die horizontale, wie hier im Bild gezeigt, oder die vertikale Kante der Zoom Box aufzuziehen. Die entstehende zweite Ecke der Zoom Box darf aber nicht mit **[ENTER]** bestätigt werden...



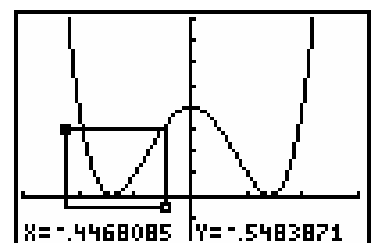
...denn sonst resultiert eine Fehlermeldung **ERR: ZOOM**, die darauf hinweist, dass kein Rechteck ausgewählt wurde. Leider gibt es hier nur die Option **Quit**, die nach ihrer Ausführung...



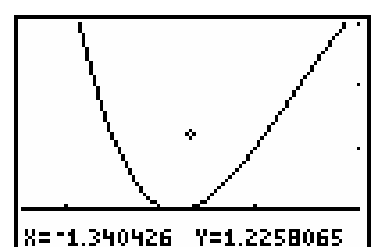
...den TI-84 in den **HOME**-Screen zurückschaltet.



Mit einem neuen Anlauf, bei dem man nun darauf achtet, nur die beiden sich diagonal gegenüberliegenden Ecken des Rechtecks mit **[ENTER]** zu bestätigen...



...erhält man schließlich doch noch die gewünschte Darstellung.



## WINDOW RANGE ERROR: EINSTELLUNG DES GRAFIKFENSTERS

Für die oft vorzunehmenden Neueinstellungen des Grafikfensters ist es natürlich Bedingung, dass der jeweilige maximale Wert größer als der minimale Wert ist.

### Vertauschte Parameter

In diesem Bildschirm wird zunächst der Funktionsterm dokumentiert, der dann grafisch dargestellt werden soll.

```
Plot1 Plot2 Plot3
Y1=X^4-4*X^2+4
Y2=
Y3=
Y4=
Y5=
Y6=
Y7=
```

Für die Parameter der x-Achse und der y-Achse muss natürlich die Bedingung erfüllt sein, dass der maximale Wert auch größer als der minimale Wert ist. Gerade wenn bestehende Werte nur editiert werden, kann es passieren, dass am Ende der Eingaben diese Bedingung nicht mehr erfüllt ist.

```
WINDOW
Xmin=-3
Xmax=3
Xscl=1
Ymin=10
Ymax=8
Yscl=1
Xres=1
```

Der TI-84 antwortet darauf mit der Fehlermeldung **ERR: WINDOW RANGE**, das bedeutet, einer oder beide Bereiche des Grafikfensters sind nicht korrekt eingestellt worden. Leider gibt es hier nur die **Quit**-Option...

```
ERR:WINDOW RANGE
Quit
```

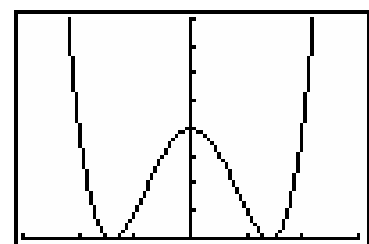
...sodass man direkt in den **HOME**-Screen zurückkehrt.



Man muss dann halt erneut in das Einstellungs Menü für das Grafikfenster gehen, um die erforderlichen Korrekturen vorzunehmen...

```
WINDOW
Xmin=-3
Xmax=3
Xscl=1
Ymin=0
Ymax=8
Yscl=1
Xres=1
```

...um schließlich doch noch die gewünschte Darstellung zu erhalten.

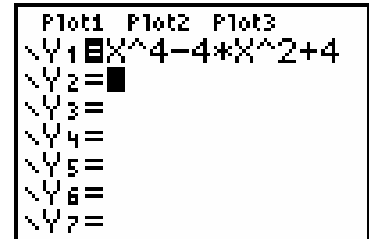


## BOUND ERROR: GRAFISCH-NUMERISCHE BERECHNUNGEN UND GRENZEN

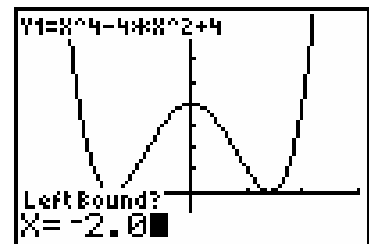
Bei der Angabe der Grenzen innerhalb grafisch numerischer Berechnungen ist natürlich darauf zu achten, dass die linke Grenze kleiner als die rechte Grenze ist.

### Nullstellen und ihre Grenzen

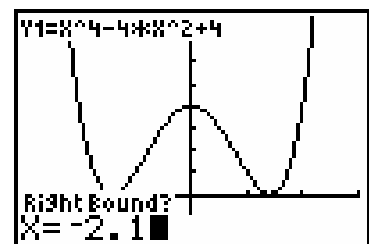
Für den rechts dokumentierten Term könnte eine mögliche Aufgabenstellung die Berechnung der Nullstellen erforderlich machen.



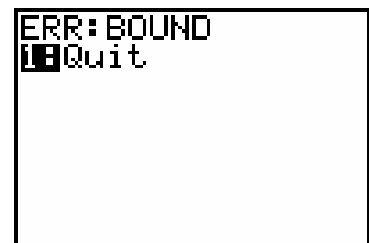
Nach der grafischen Darstellung und der Auswahl des **zero**-Befehls aus dem **CALC**-Menü fragt der TI-84 zuerst nach der linken Grenze. Aber gerade bei der Eingabe der entsprechenden Zahlenwerte über die Tastatur...



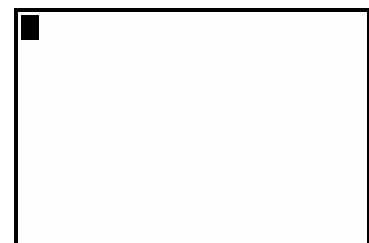
...und wohl auch bei negativen Grenzen kann es passieren, dass die Bedingung: linke Grenze kleiner rechte Grenze nicht erfüllt wird.



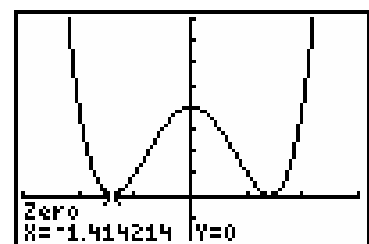
Obwohl danach vielleicht ein korrekter Schätzwert eingegeben worden sein mag, gibt der TI-84 hier die Fehlermeldung **ERR: BOUND** aus. Das bedeutet also, dass etwas mit den vorgegebenen Grenzen nicht in Ordnung ist.



Leider gibt es auch hier nur die Option, in den **HOME**-Screen zurückzukehren. Will man also die Aufgabe lösen...



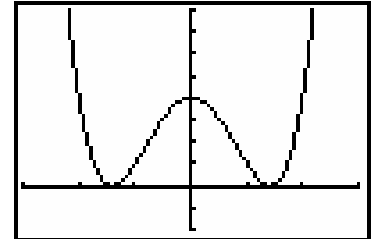
...so muss die gesamte Prozedur von vorne durchlaufen werden. Mit etwas mehr Konzentration ist dann am Ende wohl auch das erwünschte Ergebnis zu sehen.



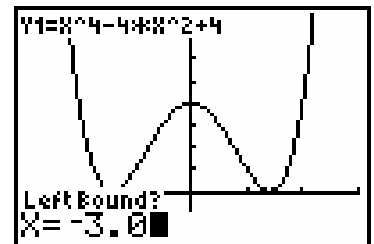
# NO SIGN CHANGE ERROR: GRAFISCH-NUMERISCHE NULLSTELLEN UND GRENZEN

Natürlich muss auch im Grafikmenü bei der Berechnung von Nullstellen beachtet werden, dass im definierten Intervall auch eine Nullstelle liegt.

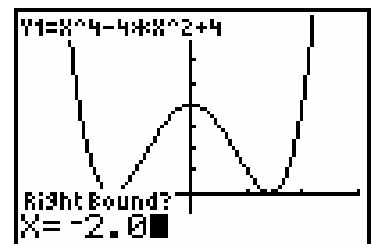
## Grenzen und keine Nullstellen



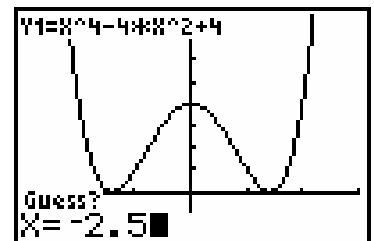
Wir gehen wieder von der Funktion der beiden vorherigen Seiten aus.



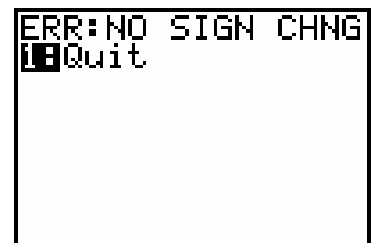
Selbst wenn die Eingaben für den **zero**-Befehl und seine Grenzen...



...in der korrekten Anordnung vorgenommen werden...



...und auch der Schätzwert noch zwischen den beiden Grenzen liegt, sollte wenn möglich auch eine Nullstelle innerhalb der Grenzen vorhanden sein. Ist dies nicht der Fall, so bleibt dem TI-84 nichts anderes über...



...als eine Fehlermeldung auszugeben. Sie lautet hier **ERR: NO SIGN CHGN** als Standardmeldung für nicht vorhandene Nullstellen mit Vorzeichenwechsel. Auch sie erlaubt leider ebenfalls nur...



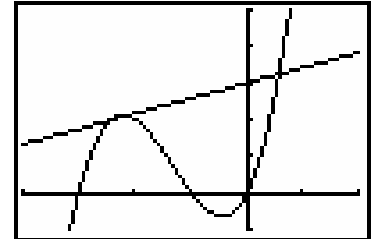
...die Rückkehr in den **HOME**-Screen. Mit entsprechender Ausdauer könnte man die Prozedur (mit sinnvollen Grenzen) erneut ablaufen lassen und würde dann das Ergebnis des letzten Bildschirms der vorherigen Seite bekommen!

## INVALID ERROR: GRAFIKPARAMETER UND NUMERISCHE EINGABEN

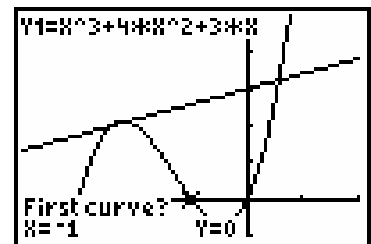
Eingaben per Tastatur, die im Grafikmenü für numerische Rechnungen benötigt werden, müssen innerhalb der aktuell eingestellten Fensterparameter liegen.

### Wert außerhalb des Grafikfensters

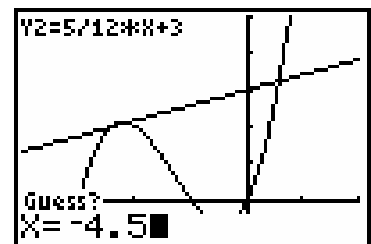
Der linke Schnittpunkt der dargestellten Funktionsgraphen soll berechnet werden.



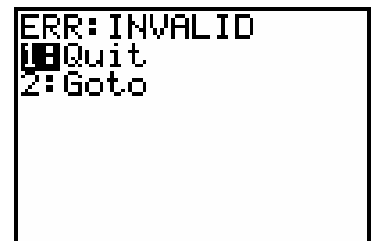
Nach dem Auswählen des **intersect**-Befehls aus dem **CALC**-Menü kommen die üblichen Abfragen nach den Kurven...



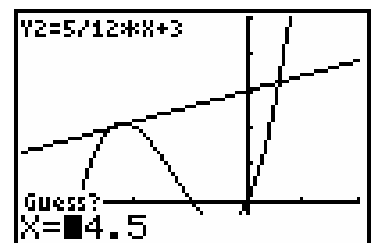
...und dann wieder die Frage nach einem Schätzwert. Manchmal passiert es dann schon, dass der gewählte Wert nicht innerhalb des durch die Parameter des Grafikfensters definierten Ausschnitts liegt.



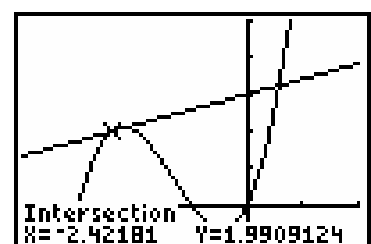
Der TI-84 gibt daraufhin die Fehlermeldung **ERR: INVALID** aus, und meint damit einen ungültigen Wert. Er erlaubt dann aber auch über den **Goto** Befehl...



...der den Cursor wieder in die Eingabezeile für den Schätzwert positioniert...



...den Fehler zu korrigieren, um dann doch noch zum gewünschten Schnittpunkt zu gelangen.

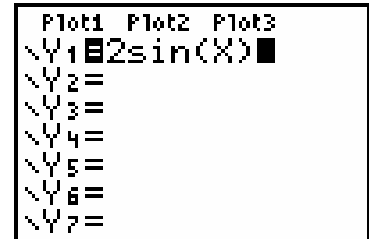


## FALSCHER MODE-EINSTELLUNG FÜR TRIGONOMETRISCHE FUNKTIONEN

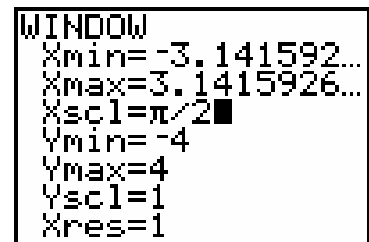
Für die grafische Darstellung trigonometrischer Funktionen ist darauf zu achten, dass das Winkelmaß der Einstellung im MODE-Menü und die Skalierung der x-Achse übereinstimmen.

### Falsche Winkelmaße

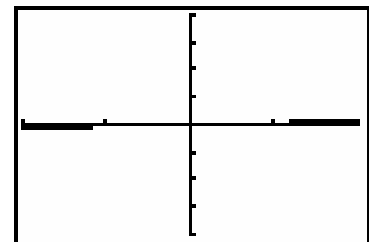
In diesem Bildschirm wird zunächst der Funktionsterm dokumentiert, der gleich grafisch dargestellt werden soll.



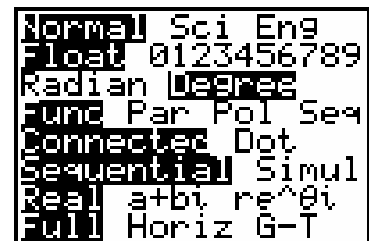
Da es sich um eine trigonometrische Funktion handelt, bietet es sich an, die Fensterparameter als Vielfache von  $\pi$  einzustellen. Der Bildschirm rechts zeigt eine mögliche Wahl.



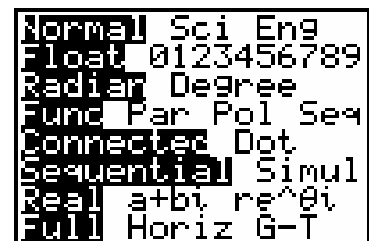
Lässt man nun mit der **GRAPH**-Taste den Grafen zu obigem Term darstellen, so entspricht das Resultat überhaupt nicht den Erwartungen an eine übersichtliche Grafik.



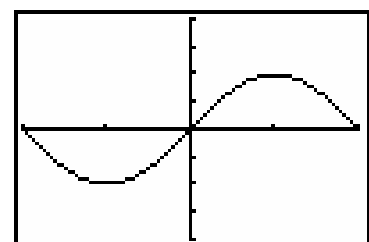
Der Grund hierfür ist in den Einstellungen im **MODE**-Menü zu suchen. Wie der Bildschirm rechts zeigt, ist die obige Grafik mit dem **Degree** Maß für Winkel gezeichnet worden. Da können ja nur sehr kleine Funktionswerte herauskommen, denn die Parameter für die x-Achse waren unter der Annahme des Bogenmaßes eingegeben worden.



Schaltet man jetzt einfach auf das Bogenmaß im **MODE**-Menü um...



...so erhält man schließlich doch noch was man ursprünglich erwartet hatte.





## UNDEFINED ERROR: NICHT DEFINIERTE MATRIX

Nur bereits definierte, also mit Werten belegte Matrizen können in weiteren Rechnungen benutzt werden.

### Verwenden einer nicht definierten Matrix

Die Namen für Matrizen sind im Menü **MATRIX/NAMES** aufgelistet und auch von hier aus aufzurufen. Achtet man nicht auf die Belegung einer Matrix, die in diesem Menü hinter dem Namen durch Zeilen- und Spaltenanzahl angezeigt würde...

```

NAMES MATH EDIT
1: [A]
2: [B]
3: [C]
4: [D]
5: [E]
6: [F]
7↓ [G]
  
```

...und baut den Namen einer nicht definierten Matrix in eine Rechnung ein, wie hier einfach nur der Name der Matrix selbst zum Ansehen ihres Inhaltes...

```

[A]
  
```

...so bekommt man die Fehlermeldung **ERR: UNDEFINED**. Man hat nun die Möglichkeit den Fehler mit **Quit** zu ignorieren oder mittels **Goto** an die Stelle zu gehen, an der der Fehler aufgetreten ist.

```

ERR: UNDEFINED
1: Quit
2: Goto
  
```

Da wir uns einfach nur den Inhalt der Matrix **A** ansehen wollten, ist es sinnvoll, gleich wieder in das **MATRIX/NAMES** Menü zu gehen. Für keinen Namen sind hier Zeilen- und Spaltenzahlen dargestellt, also ist auch keine Matrix definiert.

```

NAMES MATH EDIT
1: [A]
2: [B]
3: [C]
4: [D]
5: [E]
6: [F]
7↓ [G]
  
```

Natürlich muss man zunächst im **MATRIX/EDIT** Menü Zeilen- und Spaltenanzahl einer Matrix festlegen und danach die entsprechenden Werte eingeben...

```

MATRIX[A] 2 x2
[ 2   ]
[ 1   ]
2, 2=1
  
```

...bevor man wie hier dargestellt eine Matrix für Rechnungen verwenden kann.

```

[A]
[[2 1]
 [1 1]]
  
```

## SYNTAX ERROR: FALSCH EINGABE DES MATRIXNAMENS

Matrixnamen sind reservierte Zeichenkombinationen.  
Sie lassen sich nicht über die Tastatur direkt eingeben.

### Versuch der direkten Namenseingabe

Beim TI-84 ist die Zeichenkombination für den Datentyp Matrix ein großer Buchstabe in eckigen Klammern. Sowohl die Klammern, als auch der Buchstabe lassen sich über die Tastatur eingeben, wie es rechts dargestellt ist.

```
[A]
```

Möchte man sich dann allerdings den Inhalt der vermeintlichen Matrix anzeigen lassen, so bekommt man die Fehlermeldung **ERR: SYNTAX**, also einen Fehler in der eingegebenen Zeichenfolge.

```
ERR:SYNTAX
1:Quit
2:Goto
```

Lässt man sich nun den Fehler mit der **Goto** Option anzeigen, so wird der Cursor auf der rechten Klammer positioniert. Bei einer nicht definierten Matrix (siehe vorherige Seite) würde der Cursor dahinter stehen.

```
[A]
```

Es bleibt also nichts anderes übrig, als den Namen im **MATRIX/NAMES** Menü zu markieren...

```
NAMES MATH EDIT
1:[A] 2x2
2:[B]
3:[C]
4:[D]
5:[E]
6:[F]
7↓[G]
```

...ihn auf den HOME-Screen in eine Rechnung einzubauen oder wie hier einfach nur darstellen lassen...

```
[A]
```

...um dann schließlich den Inhalt (oder das Ergebnis einer Rechnung) zu sehen..

```
[A]      [[2 1]
          [1 1]]
```

## DIMENSION MISMATCH ERROR: FALSCHER DIMENSIONEN VON MATRIZEN

Bei Rechnungen mit Matrizen, insbesondere auch bei der Matrixmultiplikation ist darauf zu achten, dass die Dimensionen „passend“ eingestellt sind.

### Matrixmultiplikation mit falschen Dimensionen

Für das Produkt zweier Matrizen gilt die Festlegung, dass die Spaltenzahl des ersten Faktors gleich der Zeilenzahl des zweiten Faktors sein muss.

Ist dies nicht gegeben, so antwortet der TI-84 mit der Fehlermeldung **ERR: DIM MISMATCH**. Es wird also angezeigt, dass die Dimensionen nicht zueinander „passen“. Natürlich kann man mit **Quit** den Fehler ignorieren, aber auch **Goto** führt nur wieder in die Eingabezeile, ist also keine echte Hilfe.

Daher ist es sinnvoll, direkt in das **MATRIX/NAMES** Menü zu gehen, denn hier werden die Zeilen- und Spaltenanzahlen für definierte Matrizen angezeigt. Es ist zu sehen, dass die Zeilenzahl der Matrix **B** nicht mit der der Matrix **A** übereinstimmt. Die Multiplikation konnte also gar nicht ausgeführt werden.

Über das **MATRIX/EDIT** Menü kann durch eine Neueingabe der Zeilenanzahl für die Matrix **B**...

...die hier nach der Bestätigung mit **ENTER** zu sehen ist, der Fehler behoben werden. Man erkennt, dass durch die Veränderung der Zeilenanzahl die dritte Zeile der vorherigen Matrix gelöscht wurde. Nun stimmen die Dimensionen...

...sodass schließlich auch die Multiplikation durchgeführt werden kann.

```
[A]*[B]
```

```
ERR: DIM MISMATCH
1: Quit
2: Goto
```

```
NAMES MATH EDIT
1: [A] 2x2
2: [B] 3x2
3: [C]
4: [D]
5: [E]
6: [F]
7↓ [G]
```

```
MATRIX[B] 2x2
[ 2 1 ]
[ 1 1 ]
[ -1 2 ]
```

```
MATRIX[B] 2x2
[ 2 1 ]
[ 1 1 ]
```

```
[A]*[B]
[ [5 3]
[ 3 2] ]
```