DISKUS

—

Ein Programm zur symbolischen Diskussion reeller elementarer Funktionen

M. Cohrs, E. H. A. Gerbracht, W. Struckmann

Informatik-Bericht Nr. 98-03
Januar 1998

Inhaltsverzeichnis

1 Einleitung **2**

2 Installation und Laden des Packages funcdisc **3**
2.1 Benutzung der kompilierten Version 3
2.2 Installation der Quellen 4

3 Bedienung und Funktion des Rahmenprogramms `diskus` **6**
3.1 Aufruf 6
3.2 Syntax der Funktionsdefinition 7
3.3 Optionale Parameter 8
3.4 Ausgaben von `diskus` 14
3.4.1 Definitionslücken 15
3.4.2 Nullstellen, Extremstellen, Wendestellen 19
3.4.3 Symmetrieverhalten 20
3.4.4 Asymptoten 21
3.4.5 Verhalten im Unendlichen 21
3.4.6 Die Wertetabelle 21
3.4.7 Der Funktionsplot 22
3.4.8 Anmerkungen zur LaTeX-Ausgabe 23
3.5 Warnungen und Fehlermeldungen von `diskus` 24
3.6 Weiterverwendung der Ergebnisse 28

4 Details der Implementierung **31**
4.1 Prozeduren des Rahmenprogramms 33
4.2 Die automatische Plotbereichsbestimmung 34
4.2.1 Die x-Plotbereichsbestimmung 34
4.2.2 Die y-Plotbereichsbestimmung 38

5 Anwendungsbeispiele **43**

6 Erfahrungen und Ausblick **56**
6.1 Auswahl der zu diskutierenden Funktionen 56
6.2 Die Notwendigkeit hybrid numerisch-symbolischer Methoden . 57
6.3 Probleme mit Maple 58
6.3.1 Die Maple-interne Programmiersprache 58
6.3.2 Unerwünschte „Vereinfachungen“ 58
6.4 Schlußfolgerungen 60

A Beispielausgaben **61**
A.1 Die LaTeX-Ausgabe 61
A.2 Die Bildschirmausgabe im Prettyprint-Modus 64
A.3 Die Bildschirmausgabe im Textmodus 67

Literatur **69**

1 Einleitung

Die Diskussion elementarer Funktionen spielt in der Schulmathematik und jeder einführenden Analysisvorlesung eine besondere Rolle. Inwieweit können Computer dieses Problem lösen? E. H. A. Gerbracht und W. Struckmann [GS96] zeigen, daß kein Algorithmus existieren kann, der in der Lage wäre, alle elementaren Funktionen vollständig zu diskutieren. Jedes Programm muß sich daher auf Teilklassen oder Teilprobleme beschränken.

Am Institut für Programmiersprachen und Informationssysteme der Technischen Universität Braunschweig wurden in den vergangenen Jahren in einer Reihe von Arbeiten Klassen elementarer reeller Funktionen, die algorithmisch diskutiert werden können, untersucht. E. H. A. Gerbracht [Ger92] lenkt in seiner Diplomarbeit den Blick auf exakt-invertierbare Funktionen und ihre Eigenschaften. Seine theoretischen Überlegungen werden von E. H. A. Gerbracht und W. Struckmann [GS94] fortgeführt und von J. Banning [Ban95] und S. Wärther [Wär97] in Maple-Prozeduren umgesetzt. A. Klos [Klo95] sowie E. H. A. Gerbracht und W. Struckmann [GS97] gehen auf Fragen der Symmetrie von Polynomen und gebrochen rationalen Funktionen ein. M. Cohrs [Coh97] beschäftigt sich in seiner Studienarbeit mit den durch FOE-Termen (first order exponential) beschriebenen Funktionen.

Als Ergebnis dieser theoretischen und praktischen Arbeiten stehen eine Anzahl von Maple-Prozeduren zur Verfügung, die Teilprobleme einer Kurvendiskussion lösen können. Jede einzelne dieser Prozeduren dient dabei einem ganz bestimmten Zweck. So können z. B. durch die Prozedur `Defbreaks` die Definitionslücken einer Funktion bestimmt werden, mit der Prozedur `Nullstellen` werden die Nullstellen ermittelt und die Prozedur `Extremstellen` berechnet die Extremwerte. Der nächste Schritt ist die Zusammenfassung der einzelnen Prozeduren zu einem Gesamtsystem. Das Ergebnis ist das Maple-Programm `diskus`, das insbesondere auch die meisten der im Schulunterricht behandelten Funktionen vollständig diskutieren kann. Hierunter verstehen wir die symbolische Ermittlung der Null-, Extrem- und Wendestellen einer gegebenen reellen Funktion sowie die Bestimmung ihrer Symmetrieeigenschaften, ihrer Asymptoten und ihres Verhaltens im Unendlichen und zusätzlich, basierend auf diesen Daten, eine Zeichnung des Funktionsverlaufs und die Ausgabe einer Wertetabelle (vgl. [MW85]).

In diesem Report beschreiben wir `diskus`. In Abschnitt 2 gehen wir zunächst auf die verschiedenen Versionen des Programms und ihre Installation ein. Abschnitt 3 enthält ein vollständiges Bedienhandbuch. Einzelheiten der Implementierung – und hier insbesondere Fragen der automatischen Plotbereichsbestimmung – werden in Abschnitt 4 diskutiert. Abschnitt 5 stellt einige ausführliche Beispiele vor. Das abschließende Abschnitt 6 befaßt sich mit Fragen, die bei der Implementierung von `diskus` aufgetreten sind. Hier sind sowohl grundsätzliche theoretische Probleme als auch praktische Fehler in der benutzten Maple-Version gemeint.

2 Installation und Laden des Packages funcdisc

Um das Programm `diskus` benutzen zu können, muß zunächst das Package funcdisc installiert werden. Hierzu stehen grundsätzlich zwei verschiedene Möglichkeiten zur Verfügung: zum einen die Installation einer „kompilierten" Version und zum anderen die der vollständigen Quellen. Die entsprechenden Vorgehensweisen und die Vor- und Nachteile beider Möglichkeiten werden in den folgenden Abschnitten näher erläutert.

2.1 Benutzung der kompilierten Version

Bei der Verwendung der kompilierten Form des Packages funcdisc ist auf die richtige Version zu achten, d. h., die zugrunde liegende Maschine (d. i. das Betriebssystem, wahlweise UNIX oder MS-Windows), sowie die verwendete Version von Maple V (Release 3 bzw. Release 4) sind zu berücksichtigen.

Die kompilierte Form von funcdisc liegt als Maple-spezifische `.m`-Datei vor. Aus dem Namen dieser `.m`-Datei läßt sich die Version ablesen. So steht z. B. der Name `fd_w_x_4.m` für eine Version, die unter UNIX mit X-Window in Maple V Rel. 4 läuft. Entsprechend ist die Datei `fd_w_w_3.m` nur unter MS-Windows mit Maple V Rel. 3 lauffähig.

Ist die benötigte Version ermittelt, so sollte die entsprechende `.m`-Datei in das `share`-Verzeichnis des Maple-Verzeichnisbaumes kopiert werden. Ist dies geschehen, so kann anschließend in Maple mit der Kommandosequenz[1]

```
> with(share): readlib(fd_w_x_3):
```

das Package eingelesen werden. Danach sind alle Prozeduren des Packages funcdisc verfügbar.

Der vergleichsweise einfachen Installation der vorkompilierten Version von funcdisc steht jedoch der Nachteil gegenüber, daß in diesem Fall nicht die gesamte Funktionalität des Packages genutzt werden kann. Dies erklärt sich dadurch, daß in einige der Prozeduren gewisse Pfadangaben fest hineinkompiliert sind. Hiervon ist allerdings nur die LaTeX-Ausgabe, welche durch das Rahmenprogramm `diskus`, die eigentliche Benutzerschnittstelle, generiert wird, betroffen. Wird also eine kompilierte Version von funcdisc benutzt, ist darauf zu achten, daß die LaTeX-Ausgabe unterdrückt wird (siehe dazu auch Abschnitt 3.3)[2].

1 Hier dargestellt an der funcdisc-Version für Maple V Rel. 3 unter UNIX mit X-Window.

2 Es wäre auch möglich gewesen, eine kompilierte Version von funcdisc zu erstellen, welche die LaTeX-Ausgaben immer in das aktuelle Verzeichnis schreibt. Dies hat allerdings den Nachteil, daß diese Ausgaben in das für Maple aktuelle Verzeichnis geschrieben werden. Unter UNIX kann das aber bedeuten, daß man für dieses Verzeichnis nicht immer Schreibberechtigung hat, weshalb von dieser Methode Abstand genommen wurde.

2.2 Installation der Quellen

Um die volle Funktionalität des Packages funcdisc ausnutzen zu können, ist es in der aktuellen Version nötig, die Quelldateien zu installieren und einige Konfigurationen vorzunehmen. Hierbei ist es sinnvoll, die Quellen des Packages in einem eigenen Unterverzeichnis (z. B. ~/maple/funcdisc/) zu installieren.

Die Quelldateien selbst sind auf mehrere Unterverzeichnisse verteilt. So finden sich in den Verzeichnissen banning/, cohrs/, klos/ und waerther/ die einzelnen Programm-Module des jeweiligen Autors, das Verzeichnis help/ enthält die Dateien der Online-Hilfe[3] und die Verzeichnisse texout/ sowie session/ sind zur Ausgabe der LaTeX-Dateien, respektive zum Speichern von Maple-Sessions vorgesehen. Um die nötige Konfiguration so gering wie möglich ausfallen zu lassen, sollte diese Verzeichnisstruktur beibehalten werden.

Im Hauptverzeichnis der funcdisc-Quellen finden sich noch zwei weitere wichtige Dateien: main.fd und update.

Die Datei update ist ein Shell-Skript, mit welchem aus den Maple V Rel. 3-Quelldateien die Maple V Rel. 4-Quelldateien erzeugt werden können. Dabei ruft update für jede der Maple V Rel. 3-Quelldateien das Programm updtsrc[4] auf und erzeugt eine neue Datei. Um die korrekte Funktionsweise von update zu gewährleisten, sind einige Punkte zu beachten. Wird bei der Installation der funcdisc-Quellen eine andere Verzeichnisstruktur als die beschriebene gewählt, so muß das Shell-Skript update entsprechend angepaßt werden. Weiter sollte das den einzelnen Programm-Modulen zugrundeliegende Namensschema bei neu implementierten Modulen beibehalten werden. So deutet die Endung .m53 darauf hin, daß es sich bei einer solchen Datei um ein Modul für Maple V Rel. 3 handelt. Entsprechend kennzeichnet die Endung .m54 ein Modul, welches den syntaktischen Anforderungen von Maple V Rel. 4 genügt. Änderungen oder Erweiterungen der Quellen sollten an der .m53-Version der jeweiligen Datei durchgeführt werden, da ja die auf Maple V Rel. 4 lauffähige .m54-Version mittels des update-Skriptes automatisch generiert werden kann.

Die Datei main.fd stellt das Hauptprogramm von funcdisc dar. Es regelt in erster Linie das Einlesen der korrekten Versionen der einzelnen Programm-Module. Darüber hinaus werden hier einige globale Variablen belegt, die

3 Diese ist derzeit leider unvollständig. Zu den Prozeduren von S. Wärther existiert keine Online-Hilfe. Aus diesem Grund ist auch die Online-Hilfe per Voreinstellung nicht verfügbar.

4 updtsrc ist ein Bestandteil der Maple V Rel. 4-Distribution und dient dazu, ein für Maple V Rel. 3 geschriebenes Programm an die neue Syntax von Maple V Rel. 4 anzupassen.

für den reibungslosen Ablauf des Rahmenprogramms `diskus` wichtig sind. Desweiteren erzeugt `main.fd` eine kompilierte Version des Packages funcdisc (die bereits erwähnte `.m`-Datei), welche im Hauptverzeichnis der Installation abgelegt wird. Die Datei `main.fd` ist es auch, in der einige benutzerspezifische Konfigurationen vorgenommen werden müssen, welche die Angabe der Suchpfade betreffen. Anzupassen sind hier zum einen die Variablen `_home_path` und `_pre_path` sowie `_path_banning`, `_path_cohrs`, `_path_klos`, `_path_waerther`, `_path_help` und `_path_texout`.

In `_home_path` sollte das Home-Verzeichnis des Benutzers eingetragen werden. Das kann unter Umständen einfach durch die Angabe von „~“ geschehen, sofern dies auf der zugrundeliegenden Maschine unterstützt wird. In `_pre_path` sollte das funcdisc-Unterverzeichnis (z. B. `~/maple/funcdisc/`) eingetragen werden. Die restlichen Variablen beinhalten den Pfad zu den einzelnen Quelldateien und sind in der vorgegebenen Verzeichnisstruktur belegt mit `banning/`, `cohrs/`, `klos/` etc., d. h., diese Variablen brauchen bei Beibehaltung der vorgeschlagenen Verzeichnisstruktur nicht verändert zu werden. Sind die Variablen korrekt belegt, so erhält man dann durch Konkatenation der Variablen `_home_path`, `_pre_path` und z. B. `_path_banning` eine gültige Pfadangabe zu den Quelldateien der von J. Banning implementierten Prozeduren.

Durch die Variable `_path_texout` wird das Verzeichnis festgelegt, in welche die LaTeX-Ausgaben geschrieben werden. Hier ist besonders darauf zu achten, daß man für das Verzeichnis, welches sich durch die Konkatenation von `_home_path`, `_pre_path` und `_path_texout` ergibt, auch Schreibberechtigung hat.

Mittels Belegung der Variablen `_load_info` mit `1` kann man sich beim Laden der Module anzeigen lassen, von welchen Pfaden welche Programm-Module geladen werden (sollen). Dies ist insbesondere dann nützlich, wenn hierbei Fehler auftreten.

Sind alle notwendigen Änderungen in der Datei `main.fd` vorgenommen, kann das Package funcdisc in Maple geladen werden. Dazu ist es sinnvoll, Maple aus dem Verzeichnis zu starten, in dem auch `main.fd` liegt, und danach die Datei `main.fd` über das Kommando

```
> restart: read `main.fd`:
```

einzubinden. Nach erfolgreicher Ausführung von `main.fd`, d. h. nach erfolgreichem Laden der Programm-Module, steht das Package funcdisc zur Verfügung. Da der Modul-Lader `main.fd` aus den geladenen Programm-Modulen eine kompilierte Version (die `.m`-Datei) erstellt, ist das zuvor beschriebene Vorgehen zur erneuten Benutzung von funcdisc nicht notwendig, sofern die Quellen unverändert bleiben. Mit der soeben erstellten personalisierten `.m`-Datei kann dann wie in Abschnitt 2.1 beschrieben verfahren werden.

3 Bedienung und Funktion des Rahmenprogramms `diskus`

Das Package `funcdisc` stellt im wesentlichen eine Sammlung von einzelnen Prozeduren und Funktionen dar, mit deren Hilfe man gewisse Klassen elementarer Funktionen symbolisch diskutieren kann. Dabei übernimmt jede dieser Prozeduren eine ganz bestimmte Aufgabe. So kann man z. B. mit der Prozedur `Defbreaks` die Definitionslücken oder mit der Prozedur `Nullstellen` die Nullstellen einer Funktion bestimmen. Eine vollständige Kurvendiskussion durch sukzessive Anwendung dieser einzelnen Spezialprozeduren erweist sich allerdings als ziemlich mühselig, da dies eine ganze Reihe von Funktionsaufrufen erfordert. Aus diesem Grund wurde mit dem Rahmenprogramm `diskus` eine einfach zu bedienende Benutzerschnittstelle geschaffen, die es ermöglicht, eine komplette Kurvendiskussion mit nur einem Funktionsaufruf durchzuführen.

Die automatisierte Abarbeitung der einzelnen Diskussionsalgorithmen war jedoch nicht die einzige Motivation zur Erstellung von `diskus`. Es gab darüber hinaus noch weitere Maßgaben, die von `diskus` erfüllt werden sollten. So war es wünschenswert, die ermittelten Ergebnisse sowohl symbolisch als auch numerisch in einer optisch ansprechenden und vor allem übersichtlichen Form zur Ausgabe zu bringen. Eine weitere Zielsetzung war die automatische Erstellung eines möglichst aussagekräftigen Funktionsplots. Dabei galt es, den Plotbereich der Funktion so zu wählen, daß alle berechneten und die Funktion charakterisierenden Resultate auch im Plot wiederzufinden sind. Den Funktionsplot unterstützend sollte zudem eine Wertetabelle generiert werden. Um schließlich die berechneten Ergebnisse, den Plot und die Wertetabelle zu Papier bringen zu können, sollte noch die Möglichkeit implementiert werden, diese Daten in eine Datei auszugeben. Dies geschieht in der Form einer LaTeX-Datei, in welche der Funktionsplot als PostScript-Datei eingebunden wird, da Maple bereits eine (rudimentäre) LaTeX-Schnittstelle besitzt.

Die dem Rahmenprogramm `diskus` zugrundeliegenden Spezialprozeduren wurden bereits in den entsprechenden Arbeiten [Ban95], [Klo95], [Coh97] und [Wär97] beschrieben, so daß wir hier nun ausführlich auf die Bedienung des Rahmenprogramms `diskus` eingehen werden.

3.1 Aufruf

Nachdem das Package `funcdisc` geladen wurde (siehe dazu auch Abschnitt 2) kann eine vollständige Kurvendiskussion mittels des Rahmenprogramms `diskus` durchgeführt werden. Im Normalfall genügt hierzu die Eingabe

```
> diskus(f,x);
```

Dabei ist f die zu untersuchende Funktion und x die Variable, von der f abhängt. Einige Anmerkungen zur korrekten Syntax der Funktionen, die mit den Prozeduren des Packages `funcdisc` und speziell mit dem Rahmenprogramm `diskus` diskutiert werden können, finden sich im folgenden Abschnitt 3.2.

Nach dem Start von `diskus` werden nacheinander die einzelnen Prozeduren zur Bestimmung der Definitionslücken, Nullstellen, Extremwerte, Wendestellen, Symmetrieeigenschaften, Asymptoten und des Verhaltens im Unendlichen abgearbeitet. Alle Ergebnisse werden unmittelbar auf dem Bildschirm ausgegeben. Sollte es sich bei der betrachteten Funktion um eine solche handeln, die nicht vollständig symbolisch diskutiert werden kann, d. h., für die eine oder gar mehrere der spezialisierten Prozeduren keine Ergebnisse ermitteln können, dann wird in diesen Fällen eine entsprechende Meldung geliefert. Der weitere Ablauf von `diskus` wird aber nicht unterbrochen, d. h., die Erstellung des Funktionsplots, der Wertetabelle und der LaTeX-Datei wird auch in diesem Fall ausgeführt.

3.2 Syntax der Funktionsdefinition

Das Rahmenprogramm `diskus`, wie auch die meisten der von `diskus` benutzten Prozeduren des Packages `funcdisc`, erwartet die zu diskutierende Funktion in einer speziellen Notation. Da diese bereits in den Arbeiten [Ban95], [Klo95] und [Wär97] erläutert wurde, sollen an dieser Stelle nur die wichtigsten Details wiederholt werden.

- Polynome (und andere algebraische Funktionen) müssen in der sogenannten Pfeilschreibweise eingegeben werden, d. h. in der Form `x-><Polynom>`. So wird zum Beispiel die durch

  $$f(x) = x^2$$

  gegebene Normalparabel als

  ```
  f := x->x^2;
  ```

  eingegeben.

- Die Eingabe der elementaren transzendenten Grundfunktionen, wie z. B. der Sinus-, Cosinus- oder der Logarithmusfunktion, geschieht ohne Angabe eines Arguments, d. h., daß z. B. die durch

  $$f(x) = \sin(x)$$

  gegebene Funktion lediglich durch den Maple-Ausdruck

  ```
  f := sin;
  ```

  definiert wird.

- Funktionsschachtelungen der Form $f(g(x))$ sind über den @-Operator zu realisieren. So ist zur Diskussion der durch

  $$f(x) = \sqrt{\sin(x^2 + 1)}$$

  definierten Funktion die Eingabe von f als

  ```
  f := (x->x^(1/2))@sin@(x->x^2+1);
  ```

  notwendig.

- Im Zusammenhang mit dem @-Operator ist auf eine korrekte Klammerung bei der Definition der Funktion zu achten, da aufgrund der Operator-Präzedenzen der @-Operator die gleiche Bindungsstärke hat wie der Multiplikations- und der Divisionsoperator. So ist z. B. die Funktionsvorschrift

  $$f(x) = (x^2 - 3) \cdot e^{(x^2+3)}$$

  durch den Maple-Ausdruck

  ```
  f := (x->x^2-3)*(exp@(x->x^2+3));
  ```

  zu definieren. Man beachte hierbei die *notwendige* Klammerung des Teil-Ausdrucks `(exp@(x->x^2+3))`.

3.3 Optionale Parameter

Über eine Reihe von optionalen Parametern, die der Prozedur `diskus` beim Aufruf mit übergeben werden, wird dem Benutzer ermöglicht, das Verhalten von `diskus` gezielt zu beeinflussen. Die Aufrufsyntax von `diskus` lautet dann

```
> diskus(f, x[, key1=param1, key2=param2, ...]);
```

wobei `key`$_n$ das den Parameter identifizierende Schlüsselwort ist und `param`$_n$ der zugehörige Wert. Im einzelnen versteht `diskus` die folgenden optionalen Parameter:

- `verbose=<num>` mit `num` $\in \{0, 1, 2\}$ Default: `verbose=1`

  Hiermit wird der Grad der „Gesprächigkeit“ von `diskus` eingestellt. Für `verbose=0` werden nur die wichtigsten Warnungen ausgegeben, während für `verbose=2` entsprechend alle Warnungen ausgegeben werden.

- `noout=<bool>` mit `bool` $\in$ {`true, false`} Default: `noout=false`

  Für `noout=true` wird die Bildschirmausgabe der berechneten Ergebnisse vollständig unterdrückt[5]. Die Erstellung des Funktionsplots sowie die Erzeugung der Wertetabelle und die Ausgabe der LaTeX-Datei bleiben davon jedoch unberührt.

---

5 Es ist zu beachten, daß durch die Angabe von `noout=true` *nicht* die Berechnungen selbst, sondern nur die Ausgabe der berechneten Werte auf dem Bildschirm unterdrückt werden.

- `noplot=<bool>` mit `bool` $\in$ {`true, false`} Default: `noplot=false`

  Über die Angabe von `noplot=true` kann die Ausgabe des Funktionsplot unterbunden werden. Dadurch unterbleibt auch die Bestimmung des y-Plotbereichs. Ist zusätzlich noch die Option `notab=true` gesetzt (siehe unten), so wird außerdem noch die x-Plotbereichsbestimmung unterdrückt.

- `notab=<bool>` mit `bool` $\in$ {`true, false`} Default: `notab=false`

  Bei Angabe der Option `notab=true` wird keine Wertetabelle ausgegeben. Ist darüber hinaus noch der Parameter `noplot=true` (siehe oben) gesetzt, dann fallen auch die Berechnungen zur Plotbereichsbestimmung weg.

- `notex=<bool>` mit `bool` $\in$ {`true, false`} Default: `notex=false`

  Für `notex=true` wird die Erzeugung der LaTeX-Datei und der Postscript-Datei, welche den Funktionsplot enthält, unterdrückt. Die optionalen Parameter `texfile` sowie `psfile` (siehe unten) bleiben dann ohne Wirkung.

- `nonum=<bool>` mit `bool` $\in$ {`true, false`} Default: `nonum=false`

  Mit `nonum=true` wird die Ausgabe der numerischen Werte auf dem Bildschirm verhindert. Dies beeinflußt jedoch nicht die Ausgabe der numerischen Werte in die LaTeX-Datei.

- `noshort=<bool>` mit `bool` $\in$ {`true, false`} Default: `noshort=false`

  Mittels der Option `noshort` kann der Benutzer festlegen, ob versucht werden soll, bestimmte symbolische Ausdrücke zu vereinfachen oder nicht.

  Eine Vereinfachung symbolischer Ausdrücke wird dann interessant, wenn Funktionswerte berechnet werden sollen, d. h., wenn ein symbolisches Ergebnis in die Funktionsgleichung eingesetzt wird. Es hat sich gezeigt, daß Maple beim Einsetzen eines symbolischen Wertes in eine Funktionsgleichung in der Regel nur eine bloße Ersetzung der Variable x durch den einzusetzenden Wert vornimmt und für den Benutzer offensichtliche Vereinfachungen unterläßt. So ergibt sich zum Beispiel für die durch $f(x) = (x^2+1)/(x+\sqrt{2})$ definierte Funktion, welche ein Minimum bei $x_{\min} = -\sqrt{2}+\sqrt{3}$ besitzt, der Funktionswert an der Stelle $x_{\min}$ zu $f(x_{\min}) = (1/3)\cdot((-\sqrt{2}+\sqrt{3})^2+1)\cdot\sqrt{3}$. Dies entspricht auch dem von `diskus` gelieferten Ergebnis für `noshort=true`. Setzt man in diesem Falle jedoch `noshort=false`, so erhält man als Resultat den erheblich kürzeren Ausdruck $f(x_{\min}) = -2\sqrt{2}+2\sqrt{3}$.

  Auf der anderen Seite ergibt sich bei dem Versuch einen symbolischen Ausdruck zu vereinfachen allerdings nicht immer ein einfacherer bzw. kürzerer Ausdruck. Das Problem liegt hier bereits in der (wohl zum Teil subjektiven) Beurteilung dessen, was die Komplexität eines Ausdrucks ausmacht. In der vorliegenden Version von `diskus` wird der zu vereinfachende Ausdruck durch einige Standard-Kommandos wie z. B. `simplify` und `expand` transformiert und die jeweiligen Ergebnisse bzgl. ihrer Komplexität bewertet. Als Maß der Komplexität der so erhaltenen äquivalenten Ausdrücke wurde die absolute Länge gewählt, welche sich über den Speicherbedarf definiert. Dies hat allerdings unter Umständen zur Folge, daß augenscheinlich einfachere Ausdrücke von `diskus` verworfen werden, da sie Maple-intern eine längere Darstellung besitzen.

- `pretty=<bool>` mit `bool` $\in$ {`true, false`} Default: `pretty=true`

  Maple V Rel. 3 kennt zwei verschiedene Bildschirmausgabemöglichkeiten: die Bildschirmausgabe im Prettyprint-Modus und die Ausgabe der Formeln im Textmodus[6]. Beim Prettyprint-Modus (`pretty=true`) handelt es sich um eine grafische 2D-Ausgabe von Formeln. Sie entspricht der aus der Mathematik gewohnten Schreibweise mit Hoch- und Tiefstellungen, Integralzeichen, Sonderzeichen usw. Die Ausgabe im Textmodus (`pretty=false`) entspricht dagegen der Schreibweise, wie sie auch bei der Eingabe von Formeln verwendet wird. Das heißt, daß z. B. ein Ausdruck wie x^2 als `x^2` erscheint.

  Beide Ausgabemodi haben ihre Vor- und Nachteile. So ist natürlich die Ausgabe von symbolischen Werten, Formeln, etc. im Prettyprint-Format deutlich lesbarer und übersichtlicher als im Textmodus, wo die Ausgabe eines symbolischen Ausdruckes schnell mehrere Zeilen füllen kann. Unglücklicherweise bietet Maple V Rel. 3 aber keine Möglichkeit, die Formatierung der Prettyprint-Ausgaben zu beeinflussen, da im Prettyprint-Modus ausgegebene Ausdrücke grundsätzlich zentriert auf dem Bildschirm erscheinen. Hierdurch wird jedoch die übersichtliche Ausgabe von mehreren symbolischen Werten, wie sie ja im Zusammenhang mit `diskus` nötig ist, erheblich erschwert. Diesen Nachteil hat man im Textmodus nicht. Hier ist es relativ leicht möglich tabellarische Ausgaben zu erzeugen. Wie bereits erwähnt, macht es allerdings im Interesse der Lesbarkeit nur wenig Sinn, längere Ausdrücke im Textmodus symbolisch auszugeben. In `diskus` wurde deshalb auch auf diese Möglichkeit verzichtet, d. h., für `pretty=false` erscheinen die Ergebnisse auf dem Bildschirm nur in numerischer Form.

- `discont=<bool>` mit `bool` $\in$ {`true, false`}

  Der Ausdruck `discont` steht für „discontinuity“. Über diesen Wert kann man `diskus` darüber in Kenntnis setzen, daß die betrachtete Funktion Unstetigkeitsstellen aufweist. `discont` ist eigentlich eine boolesche Option des `plot`-Kommandos von Maple und wird von `diskus` in Abhängigkeit

6 Eigentlich gibt es drei Ausgabemodi: die grafische 2D-Ausgabe, die textbasierte 2D-Ausgabe und die reine Text-Ausgabe. Die textbasierte 2D-Ausgabe, quasi eine Mischung aus den beiden anderen Modi, eignet sich jedoch unserer Meinung nach weniger zur vernünftigen Ausgabe von symbolischen Werten, weshalb hier auf diese Ausgabemöglichkeit verzichtet wurde.

vom Ergebnis der Defbreaks-Prozedur entsprechend gesetzt. Leider ließen sich im Zusammenhang mit der discont-Option des plot-Kommandos in Maple V Rel. 3 immer wieder Fehler feststellen, die zu Programmabbrüchen führten. Aus diesem Grund wurde mit der discont-Option in diskus die Möglichkeit geschaffen, den Wert von discont explizit zu setzen. Wurde der discont-Wert von außen gesetzt, so wird dadurch die automatische Bestimmung des Wertes der plot-Option discont unterdrückt.

Da discont entweder von außen angegeben oder aber automatisch bestimmt wird, gibt es für diesen Parameter keine Defaulteinstellung.

- tabnum=<posint> mit posint $\in \mathbb{N}$

  Mit dem Parameter tabnum kann die Anzahl der zu berechnenden Funktionswerte in der Wertetabelle angegeben werden. Dies ist allerdings nur nötig, wenn die automatische Wahl von diskus für die Anzahl der Tabelleneinträge unbefriedigend ist.

  Die Benutzung der tabnum-Option beeinflußt auch die Schrittweite, mit welcher die Funktionswerte in der Wertetabelle berechnet werden. Je geringer der Wert für tabnum gewählt wird, desto größer wird natürlich auch die Schrittweite. Der Parameter tabnum wird von diskus ignoriert, wenn gleichzeitig die Option stepsize benutzt wird. Desweiteren steht tabnum in engem Zusammenhang mit der Option tabrange.

  Auch für tabnum gibt es keinen Defaultwert. Wird tabnum nicht von außen gesetzt, so versucht diskus einen „vernünftigen“ Wert dafür zu berechnen.

- stepsize=<real> mit real $\in \mathbb{R}$

  Mit stepsize kann die Schrittweite bei der Berechnung der Funktionswerte in der Wertetabelle gesetzt werden. Hierdurch wird direkt die Anzahl der zu berechnenden Funktionswerte beeinflußt: je kleiner der Wert für stepsize, desto größer ist die Anzahl der zu berechnenden Funktionswerte.

  Ist neben der Option stepsize auch noch die Option tabnum (siehe oben) gesetzt, so wird der Wert für tabnum ignoriert. Allein stepsize bestimmt dann den Umfang der Wertetabelle.

  Ebenso wie bei der Option tabnum gibt es für stepsize keinen Defaultwert.

- plotpoints=<posint> mit posint $\in \mathbb{N}$ Default: plotpoints=1000

  Ähnlich wie der optionale Parameter discont ist auch plotpoints eigentlich eine Option des plot-Kommandos[7]. Über diesen Parameter kann die minimale Anzahl der Plotpunkte, die für den Plot berechnet werden sollen, angegeben werden. Dabei ist allerdings zu beachten, daß die plot-Prozedur von Maple einen adaptiven Plot-Algorithmus benutzt. Das bedeutet, daß für Bereiche, in denen sich die Steigung der Kurve stärker ändert, auch

7 Diese plot-Option heißt tatsächlich numpoints.

mehr Plotpunkte berechnet werden. Aus diesem Grund ist z. B. die Normalparabel auch bei einer Wahl von nur 3 Plotpunkten als solche zu erkennen. Es werden also tatsächlich in den meisten Fällen mehr Plotpunkte berechnet als der Wert von `plotpoints` angibt.

- `xrange=[<xmin>,<xmax>]` mit `xmin,xmax` $\in \mathbb{R}$ und `xmin` $<$ `xmax`

  Über den Parameter `xrange` kann der x-Plotbereich festgelegt werden. Die automatische Bestimmung (vgl. dazu auch Abschnitt 4.2.1) wird in diesem Fall unterdrückt.

  Zur Benutzung dieser Option kann es mehrere Gründe geben. Der trivialste ist der, daß die Funktion, die diskutiert wird, keine (z. B. e^x) oder nur eine (z. B. x^2) kritische Stelle hat, die zur x-Plotbereichsbestimmung herangezogen werden kann. In beiden Fällen muß von `diskus` anhand gewisser Defaultwerte der x-Plotbereich festgelegt werden[8]. Dies kann natürlich mitunter zu unbefriedigenden Plotergebnissen führen.

  Einen weiteren Grund liefern solche Funktionen, bei denen z. B. ein sehr stark ausgeprägtes Maximum, d. h. eines mit einem betragsmäßig großen Funktionswert, einem nur sehr schwach ausgeprägten Maximum, also mit betragsmäßig kleinem Funktionswert, gegenüber steht. Für solche Fälle ist es durch eine geeignete Wahl des x-Plotbereichs mittels der `xrange`-Option möglich, sich auch ein Bild des ansonsten im Plot nicht zu erkennenden Minimums zu machen („Zoom“).

  Das Setzen des x-Plotbereichs über die `xrange`-Option hat sowohl Auswirkungen auf die automatische Bestimmung des y-Plotbereichs, da dieser auf Basis des berechneten (resp. gewählten) x-Plotbereichs ermittelt wird, als auch auf das Aussehen der Wertetabelle, welche ja für den dargestellten x-Plotbereich erstellt wird.

- `yrange=[<ymin>,<ymax>]` mit `ymin,ymax` $\in \mathbb{R}$ und `ymin` $<$ `ymax`

  Analog zur `xrange`-Option gibt es die `yrange`-Option, mittels welcher der y-Plotbereich festgelegt werden kann. Die automatische y-Plotbereichsbestimmung wird bei (korrektem) Gebrauch dieser Option natürlich unterdrückt.

  Da gerade die automatische y-Plotbereichsbestimmung ein recht schwieriges Gebiet ist (vgl. dazu auch Abschnitt 4.2.2), kann es hin und wieder notwendig werden, die automatische Berechnung des y-Plotbereichs durch Angabe eines selbst gewählten y-Plotbereichs mittels `yrange` zu unterdrücken, um einen zufriedenstellenden Plot zu erhalten.

  Läßt sich mittels der Option `yrange` die automatische y-Plotbereichs-

8 Für den Fall, daß keine kritischen Stellen zu ermitteln waren, wird als x-Plotbereich der Bereich $[-2, 2]$ gewählt. Konnte zumindest eine kritische Stelle x_k ermittelt werden, dann wird der x-Plotbereich zu $[x_k - 2, x_k + 2]$ festgelegt.

bestimmung gänzlich ausschalten, so kann man über die Option `yrmethod` (siehe unten) den Algorithmus zur Berechnung des y-Plotbereichs beeinflussen.

- `yrmethod=<num>` mit `num` $\in \{0, 1, 2\}$ Default: `yrmethod=0`

  Zur y-Plotbereichsbestimmung wurden drei sehr ähnliche Verfahren implementiert, die über die Option `yrmethod` ausgewählt werden können[9].

  Da die y-Plotbereichsbestimmung eigentlich nur experimentellen Status hat, ist es schwer, genauere Aussagen über die Qualität der einzelnen Verfahren zu machen. Die Ergebnisse für die verschiedenen möglichen Werte von `yrmethod` unterscheiden sich auch in der Regel nicht sehr. Die Praxis hat allerdings gezeigt, daß die hier implementierte y-Plotbereichsbestimmung in der Defaulteinstellung, also für `yrmethod=0`, durchaus brauchbare Ergebnisse liefert[10] und daß für Polynome höheren Grades (≥ 4) die Einstellung `yrmethod=1` oder `yrmethod=2` gewählt werden sollte. Beim gegenwärtigen Stand der Implementation macht es im Fall eines von `diskus` ungenügend automatisch gewählten y-Plotbereichs wahrscheinlich in der Regel mehr Sinn, diesen über die Option `yrange` manuell zu verändern, anstatt die verschiedenen Möglichkeiten von `yrmethod` auszuprobieren.

  Die Option `yrmethod` bleibt natürlich ohne Wirkung, wenn gleichzeitig durch die Option `yrange` (siehe oben) ein vom Benutzer selbstgewählter y-Plotbereich angegeben wurde.

- `tabrange=[<tmin>,<tmax>]` mit `tmin,tmax` $\in \mathbb{R}$ und `tmin` $<$ `tmax`

  Mit der Option `tabrange` wird der x-Bereich, innerhalb dessen die Funktionswerte für die Wertetabelle berechnet werden, festgelegt. Ist diese Option nicht angegeben, so ist das Intervall `tabrange` nahezu identisch mit dem x-Plotbereich (siehe dazu auch Abschnitt 3.4.6.

- `texfile=<string>` Default: `texfile='diskus.tex'`

  Mit der Option `texfile` kann der Name der LaTeX-Ausgabedatei festgelegt werden. Gibt man den Namen inklusive der Endung `.tex` an, so ist darauf zu achten, daß dieser in die Maple-typischen Zeichenkettenbegrenzer (einfache Hochkommata) einzuschließen ist. Endet der eingegebene Name nicht auf `.tex`, so wird er intern um diese Endung ergänzt. Diese Option bleibt ohne Wirkung, wenn gleichzeitig die Option `notex=true` gesetzt ist.

  Wird die Option `psfile` (siehe unten) weggelassen, so hat die Benutzung der Option `texfile` auch Auswirkungen auf den Namen der zu erzeugen-

9 Genauer müßte man sagen, daß das implementierte Verfahren über die Option `yrmethod` in drei verschiedenen Weisen beeinflußt werden kann.

10 Dies gilt umso mehr, wenn man sie mit den Plots vergleicht, bei denen Maple den y-Plotbereich selbst wählt.

den PostScript-Datei. Diese Datei wird dann ebenso wie die LaTeX-Datei benannt, erhält jedoch die Endung `.ps`.

Bei der Ausgabe der LaTeX-Datei und auch der PostScript-Datei wird *nicht* geprüft, ob Dateien dieser Namen bereits existieren. Eventuell vorhandene Dateien gleichen Namens werden überschrieben.

- `psfile=<string>` Default: `psfile=diskus.ps`

  Mit dieser Option legt man den Namen der zu erzeugenden PostScript-Datei fest, welche den Funktionsplot enthält. Ein eingegebener String, der nicht auf `.ps` endet, wird automatisch um diesen Suffix ergänzt.

  Das Weglassen dieser Option hat zur Folge, daß der resultierende Dateiname aus dem Namen der LaTeX-Datei gebildet wird. Hat z. B. die LaTeX-Datei den Namen `test.tex`, so ergibt sich der Name der Postscript-Datei zu `test.ps`. Ebenso wie bei der Option `texfile` gilt auch hier, daß bereits vorhandene Dateien gleichen Namens überschrieben werden.

3.4 Ausgaben von diskus

Um für den Benutzer eine bessere Lesbarkeit zu erzielen, wurden die Ausgaben der meisten Prozeduren des Packages funcdisc abgeändert[11]. Zudem werden, sofern sinnvoll, zu den ermittelten Resultaten noch die Funktionswerte berechnet. Handelt es sich bei einem Ergebnis um einen echten symbolischen Wert, d. h. nicht um eine ganze Zahl, so wird dieser Wert zusätzlich in numerischer Form ausgegeben. Dies ist insbesondere bei sehr komplexen symbolischen Ausdrücken von Vorteil. Anhand einiger Beispiele soll das Aussehen und die Interpretation der drei verschiedenen Ausgabeformen, die Bildschirmausgabe im Prettyprint-Modus sowie im Textmodus und die LaTeX-Ausgabe, näher erläutert werden.

Konnte zu einer Aufgabenstellung kein Ergebnis ermittelt werden, so wird die Meldung „`keine gefunden oder keine vorhanden`“ ausgegeben. Erscheint diese Meldung z. B. bei der Bestimmung der Nullstellen, so kann dies bedeuten, daß die betrachtete Funktion tatsächlich keine Nullstellen besitzt oder aber, daß vorhandene Nullstellen von Maple nicht ermittelt werden konnten. Eine genauere Aussage ist dann leider nicht möglich, da die von `diskus` benutzten Prozeduren diese Fälle nicht genügend differenzieren. Eine Ausnahme bildet hier die Bestimmung der Definitionslücken von Polynomen. Da Polynome keine Definitionslücken besitzen, erscheint in diesem Fall in der Ausgabe die Meldung „`keine`“. Desweiteren kann es unter Umständen vorkommen, daß in der Ausgabe „`FAILED`“ erscheint. Dies passiert, wenn die

11 Dem Wunsch nach einer übersichtlichen und lesbaren Ausgabe der Ergebnisse konnte leider nur zum Teil nachgekommen werden, da Maple V Rel. 3 keine Möglichkeiten zur formatierten Ausgabe symbolischer Ergebnisse im Prettyprint-Modus bietet.

entsprechende Prozedur nicht korrekt abgearbeitet werden konnte, da z. B. die Funktion syntaktisch falsch eingegeben wurde.

Für den Fall, daß der Benutzer mit den von `diskus` ausgegebenen Ergebnissen weitere Berechnungen anstellen möchte, wurde eine Möglichkeit geschaffen, auf diese Daten auch nach Beendigung des Programmlaufs zugreifen zu können. Näheres dazu findet sich im Abschnitt 3.6.

3.4.1 Definitionslücken

Bei der Bestimmung der Definitionslücken können insgesamt sechs verschiedene Ergebnisformen auftreten: einfache und periodisch wiederkehrende Punkte, einfache und periodisch auftretende Intervalle sowie Gleichungen und Ungleichungen. Funktionswerte werden bei der Definitionslückenbestimmung natürlich nicht berechnet.

- Einfache Punkte am Beispiel von $f(x) = \dfrac{x^3 - 5x - 1}{x^2 - 3}$
  - Bildschirmausgabe im Prettyprint-Modus

    Die einzelnen Ergebnisse werden zeilenweise untereinander angeordnet. Numerische Werte werden als solche gekennzeichnet und direkt unter dem zugehörigen symbolischen Wert ausgegeben. Mehrere Werte werden durch optische Trenner separiert. Im Beispiel ergibt sich somit die Ausgabe

$$-\sqrt{3}$$

$$\mathit{Numerisch}:$$

$$-1.732050808$$

$$- - - - - - - - - - - - - - - -$$

$$\sqrt{3}$$

$$\mathit{Numerisch}:$$

$$1.732050808$$

    Das heißt also, die Funktion f besitzt zwei Definitionslücken, eine bei $-\sqrt{3}$ und eine bei $\sqrt{3}$.

  - Bildschirmausgabe im Textmodus

    Es werden nur numerische Werte ausgegeben, da längere symbolische Ausdrücke in diesem Modus sehr schnell sehr unübersichtlich werden. Die beiden Definitionslücken sind wiederum untereinander aufgelistet.

```
-1.7320508
 1.7320508
```

- ∘ LaTeX-Ausgabe
  Das Aussehen der Ausgabe ist ähnlich der Bildschirmausgabe im Prettyprint-Modus, die einzelnen Punkte werden hier allerdings zusätzlich noch durchnumeriert.

$$\begin{aligned} x_{def_1} &= -\sqrt{3} \\ &\approx -1.732050808 \end{aligned}$$

$$\begin{aligned} x_{def_2} &= \sqrt{3} \\ &\approx 1.732050808 \end{aligned}$$

- Periodisch auftretende Punkte am Beispiel von $f(x) = \tan(x)$
  - ∘ Bildschirmausgabe im Prettyprint-Modus
    Periodisch auftretende Punkte sind durch die Zeichenkette $\_per$ gekennzeichnet, welche die Laufvariable in dem periodischen Ausdruck darstellt. Für sie gilt grundsätzlich $\_per \in \mathbb{N}$. Für periodisch auftretende Punkte werden weder Funktionswerte berechnet, noch werden sie in numerischer Form dargestellt. Die Ausgabe hat in diesem Fall folgendes Aussehen:

$$\frac{1}{2}\pi + \_per\,\pi$$

  - ∘ Bildschirmausgabe im Textmodus
    Auch in diesem Fall werden keine numerischen Werte berechnet, d. h., auch hier wird lediglich der ermittelte Wert ausgegeben.

```
1/2*Pi + _per * Pi
```

  - ∘ LaTeX-Ausgabe
    Die LaTeX-Ausgabe periodisch auftretender Punkte erfolgt in Mengen-Notation. Es wird eine Menge definiert, welche alle Definitionslücken enthält. Auf die Ausgabe von numerischen Werte wurde hier ebenfalls verzichtet.

$$\mathcal{M}_{def_1} = \left\{ x \in \mathbb{R} \mid x = \frac{\pi}{2} + k\pi \text{ mit } k \in \mathbb{Z} \right\}$$

- Einfache Intervalle am Beispiel[12] von $f(x) = \ln(16 - x^2)$
  - ∘ Bildschirmausgabe im Prettyprint-Modus

---

12 Im Zusammenhang mit dieser Funktion tritt bei der Definitionslückenbestimmung mittels funcdisc unter Maple V Rel. 3 ein Fehler auf der auf einen Bug in Maple V Rel. 3 zurückzuführen ist. Statt des hier angegebenen korrekten Intervalls wird $-\infty \leq x, x < -4$ geliefert. Unter Maple V Rel. 4 tritt dieser Fehler nicht auf.

Die Intervalle werden immer in der Form „<linke Grenze> $\lhd_1$ x, x $\lhd_2$ <rechte Grenze>", mit $\lhd_{1,2} \in \{<, \leq\}$, ausgegeben, was dem mathematischen Ausdruck „<linke Grenze> $\lhd_1$ x $\lhd_2$ <rechte Grenze>", mit $\lhd_{1,2} \in \{<, \leq\}$, entspricht[13]. Auch für Intervalle werden weder Funktionswerte noch numerische Werte ausgegeben.

$$-\infty < x, x \leq -4$$

$$- - - - - - - - - - - - - - - -$$

$$4 \leq x, x < \infty$$

Die Ausgabe besagt also, daß f nur im Intervall $-4 < x < 4$ definiert ist. Es sei an dieser Stelle nochmals explizit darauf hingewiesen, daß von `diskus` die Bereiche angegeben werden, in denen die Funktion *nicht* definiert ist.

- Bildschirmausgabe im Textmodus

  Im Textmodus werden die Werte $-\infty$ bzw. ∞ als `-infinity` bzw. `infinity` ausgegeben. Die Ausgabe numerischer Werte entfällt auch hier.

```
-infinity < x <= -4
4 <= x < infinity
```

- LaTeX-Ausgabe

  Ebenso wie bei den periodisch auftretenden Punkten, wird auch für Intervalle in der LaTeX-Ausgabe eine Mengen-Notation benutzt. Jedes Intervall entspricht dabei einer Menge.

$$\mathcal{M}_{def_1} = \{x \in \mathbb{R} \mid -\infty < x \leq -4\}$$

$$\mathcal{M}_{def_2} = \{x \in \mathbb{R} \mid 4 \leq x < \infty\}$$

- Periodisch auftretende Intervalle am Beispiel von $f(x) = \sqrt{\sin(x+1)}$
  - Bildschirmausgabe im Prettyprint-Modus

    Das Aussehen der Ausgabe ist ähnlich der der einfachen Intervalle, mit dem Unterschied, daß es sich bei den Grenzen der periodisch auftretende Intervalle um periodisch wiederkehrende Werte handelt.

$$-\pi - 1 + 2\ \_per\, \pi < x, x < -1 + 2\ \_per\, \pi$$

13 Leider ließ sich es sich aufgrund der Beschränkungen des Maple-Kommandos `print` nicht vermeiden, solche Ausdrücke zweigeteilt auszugeben.

- ◦ Bildschirmausgabe im Textmodus

```
-Pi-1 + _per * 2*Pi < x < -1 + _per * 2*Pi
```

- ◦ LaTeX-Ausgabe

$$\mathcal{M}_{def_1} \quad = \quad \{x \in \mathbb{R} \mid -\pi - 1 + 2\,k\pi < x < -1 + 2\,k\pi \text{ mit } k \in \mathbb{Z}\}$$

- Gleichungen am Beispiel von $f(x) = \tan(x^2)$

  Kann ein symbolischer Ausdruck nicht vollständig nach x aufgelöst werden, so wird das Ergebnis in Form einer Gleichung oder Ungleichung dargestellt. Dies tritt insbesondere im Zusammenhang mit periodisch auftretenden Werten auf.

  - ◦ Bildschirmausgabe im Prettyprint-Modus

$$\frac{1}{2}\,\pi + \_per\,\pi = x^2$$

  - ◦ Bildschirmausgabe im Textmodus

```
1/2*Pi + _per * Pi = x^2
```

  - ◦ LaTeX-Ausgabe

$$\mathcal{M}_{def_1} \quad = \quad \left\{x \in \mathbb{R} \mid \frac{\pi}{2} + k\pi = x^2 \text{ mit } k \in \mathbb{Z}\right\}$$

- Ungleichungen am Beispiel von $f(x) = \sqrt{\sin(x^2+1)}$
  - ◦ Bildschirmausgabe im Prettyprint-Modus[14]

$$(\,-\pi + 2\,\_per\,\pi < x, x < 2\,\_per\,\pi\,) = x^2 + 1$$

  - ◦ Bildschirmausgabe im Textmodus

```
-Pi + _per * 2*Pi < x^2+1 < 0 + _per * 2*Pi
```

  - ◦ LaTeX-Ausgabe

$$\mathcal{M}_{def_1} \quad = \quad \left\{x \in \mathbb{R} \mid -\pi + 2\,k\pi < x^2 + 1 < 2\,k\pi \text{ mit } k \in \mathbb{Z}\right\}$$

---

14 Wie ein solcher Ausdruck zu interpretieren ist, kann der LaTeX-Ausgabe entnommen werden.

3.4.2 Nullstellen, Extremstellen, Wendestellen

Bei der Bestimmung der Null-, der Extrem- und der Wendestellen sieht die Ausgabe der Ergebnisse ähnlich wie die der Definitionslückenbestimmung aus, jedoch sind hier keine Intervalle zu berücksichtigen. Dagegen werden zu den ermittelten Extrem- und Wendestellen die Funktionswerte berechnet, sofern es sich bei diesen Stellen um einfache Punkte handelt. Werden die Funktionswerte mit ausgegeben, so stehen diese in der Ausgabe, durch ein Komma getrennt, direkt hinter dem zugehörigen x-Wert. Anhand des folgenden Beispiels sollen die verschiedenen möglichen Ausgabeformen mit Funktionswerten veranschaulicht werden.

- Ausgabe der Minima am Beispiel von $f(x) = (x^2 - 3) \cdot e^{(x^2+3)}$
  - Bildschirmausgabe im Prettyprint-Modus

$$-\sqrt{2}, -\mathrm{e}^5$$

$$\mathit{Numerisch}:$$

$$-1.414213562, -148.4131591$$

$$- - - - - - - - - - - - - - - - -$$

$$\sqrt{2}, -\mathrm{e}^5$$

$$\mathit{Numerisch}:$$

$$1.414213562, -148.4131591$$

  - Bildschirmausgabe im Textmodus

```
(-1.4142136, -148.41316)
( 1.4142136, -148.41316)
```

  - LaTeX-Ausgabe

$$\begin{aligned}(x_{min_1}\,,\,y_{min_1}) &= \left(-\sqrt{2}\,,\,-e^5\right)\\ &\approx (-1.414213562\,,\,-148.4131591)\end{aligned}$$

$$\begin{aligned}(x_{min_2}\,,\,y_{min_2}) &= \left(\sqrt{2}\,,\,-e^5\right)\\ &\approx (1.414213562\,,\,-148.4131591)\end{aligned}$$

3.4.3 Symmetrieverhalten

Die Ausgabe des Symmetrieverhaltens wird unterschieden nach Achsensymmetrie und Punktsymmetrie. Bei Achsensymmetrie wird der die Achse beschreibende x-Wert ausgegeben, bei Punktsymmetrie die Koordinaten des Symmetriepunktes.

- Achsensymmetrie am Beispiel von $f(x) = 2 \cdot (x-2) \cdot (x+1)$
  - Bildschirmausgabe im Prettyprint-Modus

$$\frac{1}{2}$$

$$Numerisch:$$

$$.5000000000$$

  - Bildschirmausgabe im Textmodus

```
0.5
```

  - LaTeX-Ausgabe

$$\begin{aligned} x_{achssym_1} &= 1/2 \\ &\approx 0.5000000000 \end{aligned}$$

- Punktsymmetrie am Beispiel von $f(x) = x^3 - 4x^2$
  - Bildschirmausgabe im Prettyprint-Modus

$$\frac{4}{3}, \frac{-128}{27}$$

$$Numerisch:$$

$$1.333333333, -4.740740741$$

  - Bildschirmausgabe im Textmodus

```
( 1.3333333, -4.7407407)
```

  - LaTeX-Ausgabe

$$\begin{aligned} (x_{punktsym_1}\,, y_{punktsym_1}) &= \left(4/3\,, -\frac{128}{27}\right) \\ &\approx (1.333333333\,, -4.740740741) \end{aligned}$$

3.4.4 Asymptoten

Bei der Bestimmung des asymptotischen Verhaltens werden die senkrechten, waagerechten und schiefen Asymptoten berechnet. Ihre Ausgabe erfolgt durch Angabe des x-Wertes in der Form $x = <x\text{-Wert}>$ für senkrechte Asymptoten, des y-Wertes in der Form $y = <y\text{-Wert}>$ für waagerechte Asymptoten bzw. der Geradengleichung in der Form $y = <\text{Ausdruck}>$ im Falle schiefer Asymptoten. In allen drei Fällen werden keine zusätzlichen numerischen Werte ausgegeben.

3.4.5 Verhalten im Unendlichen

Die Ausgabe des Verhaltens im Unendlichen erfolgt durch Angabe der ermittelten Grenzwerte für $x \to -\infty$ und $x \to \infty$. Numerische Werte werden auch hier nicht berechnet. Handelt es sich bei den ermittelten Ergebnissen um die Werte $-\infty$ oder ∞, so wird bei der Bildschirmausgabe im Textmodus die Zeichenkette `-Unendlich` bzw. `Unendlich` ausgegeben. Sowohl in der Prettyprint-Ausgabe als auch in der LaTeX-Ausgabe erscheinen hingegen die symbolischen Werte $-\infty$ bzw. ∞.

3.4.6 Die Wertetabelle

Um die von `diskus` erstellte Wertetabelle möglichst platzsparend auszugeben, wurde diese zweidimensional konzipiert. Abbildung 1 auf Seite 21 zeigt exemplarisch eine von `diskus` automatisch generierte Wertetabelle zu der durch $f(x) = \frac{1}{x+1} \cdot e^{x+5}$ gegebenen Funktion.

Ein Wert der ersten Spalten addiert zu einem Wert der ersten Zeile ergibt einen x-Wert. Im Kreuzungspunkt dieser beiden Linien steht der zu diesem x-Wert gehörende Funktionswert. Die in Abbildung 1 dargestellte Wertetabelle umfaßt damit also den x-Bereich $[-2, 1]$, und der Funktionswert an der Stelle -1.1 findet sich in der sechsten Spalte der dritten Zeile, lautet also -494.02.

	+0	+0.1	+0.2	+0.3	+0.4
-2	-20.086	-24.664	-30.666	-38.732	-49.94
-1.5	-66.231	-91.496	-134.82	-223.51	-494.02
-1	undefiniert	603.4	333.43	245.67	203.63
-0.5	180.03	165.81	157.07	151.89	149.21
0	148.41	149.11	151.06	154.11	158.15
0.5	163.13	169.02	175.8	183.5	192.12
1	201.71	-	-	-	-

Abbildung 1: Von `diskus` erzeugte Wertetabelle zu $f(x) = \dfrac{1}{x+1} \cdot e^{x+5}$

Wie man in dem hier gezeigten Beispiel erkennen kann, können außer den reellen Funktionswerten noch weitere Symbole, nämlich „`undefiniert`“,

„`complex`“ und „`-`“, auftreten. Das Symbol `undefiniert` wird ausgegeben, wenn die Funktion an der entsprechenden Stelle undefiniert ist, das Symbol `complex` wird ausgegeben, wenn der Funktionswert an der zugehörigen Stelle komplexwertig ist, und das Symbol `-` steht für nicht berechnete Funktionswerte.

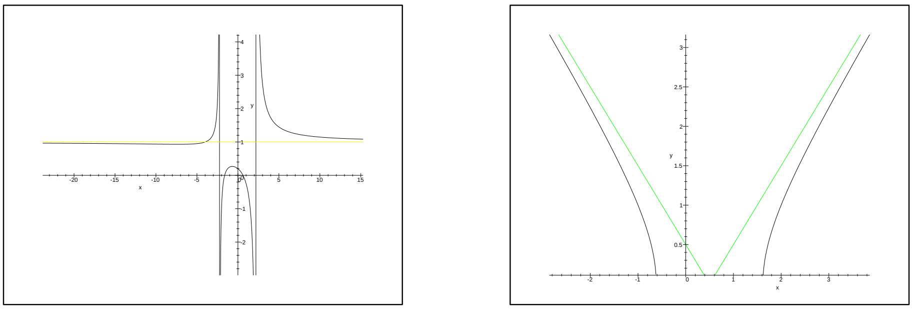
Die Wertetabelle wird begleitend zum Funktionsplot erstellt, d. h., der hier gewählte x-Bereich ist im wesentlichen identisch mit dem des Funktionsplots. Allerdings wird beim Erstellen der Wertetabelle darauf geachtet, daß die zugrundeliegenden x-Werte exakte Dezimalzahlen sind, die gleichverteilt im gewählten Bereich liegen. Wird z. B. durch die automatische x-Plotbereichsbestimmung der x-Plotbereich $\left[-\sqrt{2}, \sqrt{2}\right]$ ermittelt, so macht es sicherlich wenig Sinn, genau diesen Bereich auch als Grundlage für die Wertetabelle zu benutzen, da man hier unweigerlich sehr „krumme“ x-Werte erhält. Daher werden in diesem Fall die Bereichsgrenzen zunächst sinnvoll gerundet, um x-Werte mit einer geringen Anzahl von Nachkommastellen zu erhalten. Dabei wird jedoch darauf geachtet, daß der der Wertetabelle zugrundeliegende x-Bereich nicht wesentlich vom x-Plotbereich abweicht.

3.4.7 Der Funktionsplot

Der von `diskus` erstellte Funktionsplot wird nach den berechneten Ergebnissen der Kurvendiskussion erstellt und zeigt im Normalfall die Funktion mit all ihren kritischen Stellen. Konnte die Funktion vollständig diskutiert werden, so sind in dem Plot alle ermittelten Resultate, d. h. die Definitionslücken, Nullstellen, Extremstellen usw., wiederzufinden.

Zusätzlich zum Graph der diskutierten Funktion werden in dem Plot auch noch die eventuell vorhandenen waagerechten und schiefen Asymptoten dargestellt. Auf die Darstellung der senkrechten Asymptoten muß leider verzichtet werden, da Maple keine senkrechten Linien im Plot unterstützt. Trotzdem kann es, wie es z. B. auch in Abbildung 2a zu sehen ist, vorkommen, daß vermeintliche senkrechte Asymptoten im Plot auftauchen. Auch wenn diese senkrechten Linien mit den Asymptoten deckungsgleich sein sollten, handelt es sich hierbei tatsächlich nicht um die Asymptoten, sondern um einen von Maple's `plot`-Kommando fälschlich eingezeichneten Funktionsverlauf. Zu diesen senkrechten Linien kommt es nämlich immer dann, wenn die Funktion auf der einer Seite einer Definitionslücke gegen ∞ und auf der anderen gegen $-\infty$ geht und die Definitionslücke zwischen zwei Plotpunkten liegt. Das bedeutet, daß z. B. bei Definitionslücken, die bei ganzzahligen x-Werten liegen, dieses Phänomen nicht auftritt und die scheinbaren Asymptoten im Plot fehlen. Zwar läßt sich dem `plot`-Kommando mitteilen, daß die zu plottende Funktion Definitionslücken besitzt, jedoch leider nicht wo diese liegen.

Um die Asymptoten vom Funktionsverlauf unterscheiden zu können, werden sie in einer anderen Farbe als die Funktion geplottet. Dabei gilt, daß der Graph der Funktion in Rot gezeichnet wird und daß die waagerechten Asymptoten in Gelb[15] und die schiefen Asymptoten in Grün erscheinen. Zwei Beispiele für das Aussehen der von `diskus` erstellten Plots sind in Abbildung 2 zu sehen.



(a) $f(x) = \dfrac{x^2 + x - 1}{x^2 - 5}$ mit waagerechter Asymptote

(b) $f(x) = \sqrt{x^2 - x - 1}$ mit zwei schiefen Asymptoten

Abbildung 2: Beispiel für von `diskus` erstellte Plots

Es sei noch darauf hingewiesen, daß sich unter Maple V Rel. 3 zur Darstellung eines Plots immer ein neues Fenster öffnet, während sich bei Maple V Rel. 4 über eine Maple-Option einstellen läßt, ob Plots in einem eigenen Fenster oder aber direkt im Arbeitsblatt der laufenden Maple-Session erscheinen sollen.

3.4.8 Anmerkungen zur LaTeX-Ausgabe

Die von `diskus` erstellte LaTeX-Datei ist eine LaTeX 2_ε-kompatible Datei der Klasse `report` die mit den Optionen `10pt` und `a4paper` aufgerufen wird. Zudem wird von einigen besonderen LaTeX-Paketen Gebrauch macht. So müssen, um die LaTeX-Datei erfolgreich übersetzen zu können, die folgenden LaTeX-Pakete installiert sein:

- `babel`, das mit der Option `german` geladen wird,
- `epsfig`, welches zum Einbinden des als PostScript-Datei vorliegenden Plots dient,
- `longtable`, mit dem die Wertetabelle dargestellt wird und

15 Waagerechte Asymptoten bei $y = 0$ sind in den Plots nicht zu sehen, da sie von der x-Achse überlagert werden.

- `amsfonts`, durch welches die Ausgabe der Symbole $\mathbb{N}$ (Menge der natürlichen Zahlen), $\mathbb{Z}$ (Menge der ganzen Zahlen) und $\mathbb{R}$ (Menge der reellen Zahlen) realisiert wird.

Durch die Verwendung des Paketes `longtable` ist es unter Umständen notwendig, die von `diskus` ausgegebene LaTeX-Datei dreimal zu übersetzen, bis das endgültige Aussehen erreicht ist. Die durch `longtable` dargestellten Tabellen weisen dann aber eine feste Breite auf, auch wenn eine Tabelle über mehrere Seiten des Dokumentes gehen sollte.

Bei der Ausgabe von symbolischen Ausdrücken in die LaTeX-Datei kann es vorkommen, daß längere Ausdrücke nicht auf die Seite passen, d. h., daß diese die Seitenbreite überschreiten. Der an dieser Stelle notwendige Umbruch in dem Ausdruck passiert nicht automatisch, sondern muß vom Benutzer manuell durchgeführt werden.

3.5 Warnungen und Fehlermeldungen von `diskus`

Während des Ablaufs des Rahmenprogramms `diskus` können eine Reihe von Warnungen und Fehlermeldungen auf dem Bildschirm erscheinen, die im folgenden kurz erläutert werden sollen.

Grundsätzlich wurde versucht, daß Programm so auszulegen, daß es bei einem auftretenden Fehler in einem der benutzten Prozeduren nicht sofort abbricht. Es gibt jedoch wenigstens zwei Fälle, in denen ein Programmabbruch unausweichlich ist: zum einen, wenn der Prozedur `diskus` weniger als zwei Parameter übergeben werden (Fehlermeldung: `ungueltige Parameteranzahl!!!`), und zum anderen, wenn keine der aufgerufenen Prozeduren Resultate ermitteln konnte (Fehlermeldung: `Funktion kann nicht ausgewertet werden. Bearbeitung abgebrochen!!!`).

Alle weiteren Probleme, die in den benutzten Prozeduren auftreten können, wurden abgefangen und in eine entsprechende Warnung übersetzt. Der Programmablauf wird dadurch nicht unterbrochen, der Benutzer erhält so aber dennoch Hinweise auf eventuell unerwartete Ergebnisse.

Die einzelnen Warnungen wurden je nach Relevanz in drei Stufen unterteilt. Die wichtigen Warnungen, die immer dann erscheinen, wenn Umstände aufgetreten sind, die das Verhalten von `diskus` nachhaltig verändern und unerwartete Ergebnisse hervorrufen können, sind in der Stufe 0 angesiedelt. Die Warnungen, die eher den Charakter einer Bemerkung haben sind, in der Stufe 2 angesiedelt. Die Warnungen der Stufe 1 können zwar Auswirkungen auf das Verhalten von `diskus` haben, diese sind aber in der Regel nicht gravierend.

Der Benutzer hat durch den optionalem Parameter `verbose=<num>` (siehe Abschnitt 3.3) die Möglichkeit, die Stufe der auszugebenden Warnungen einzustellen. Je höher die eingestellte Stufe, desto mehr Warnungsmeldungen werden gegebenenfalls ausgegeben. Im einzelnen können die folgenden Warnungen auftreten:

- `Warnung!! Ignoriere unbekannte Option: <name>` Stufe: 0

  Diese Warnung wird ausgegeben, wenn `diskus` die unbekannte Option `<name>` übergeben wurde. Dies kann auftreten, wenn `<name>` an sich unbekannt ist (z. B. durch einen Schreibfehler) oder aber der Wert der Option außerhalb des zulässigen Bereiches liegt (wie z. B. `verbose=3`).

  Wie schon aus dem Wortlaut der Warnungsmeldung zu entnehmen ist, wird in einem solchen Fall die unbekannte Option `<name>` ignoriert, der weitere Programmablauf wird dadurch nicht beeinträchtigt.

- `Warnung!! Unzulaessiger x-Plotbereich: <[x1,x2]> Bestimme x-Plotbereich autom.!` Stufe: 0

  Diese Warnung wird ausgegeben, wenn versucht wird, über den optionalen Parameter `xrange=[x1,x2]` einen x-Plotbereich anzugeben, für den `x1 = x2` gilt[16]. In diesem Fall wird wieder die automatische x-Plotbereichsbestimmung aktiviert.

- `Warnung!! Unzulaessiger y-Plotbereich: <[y1,y2]> Bestimme y-Plotbereich autom.!` Stufe: 0

  Analog zur vorherigen Warnung wird mit dieser Meldung auf einen ungültig gesetzten y-Plotbereich (bei dem `y1 = y2` gilt) hingewiesen. Auch hier wird dann wieder auf die automatische y-Plotbereichsbestimmung zurückgegriffen.

- `Warnung!! Unzulaessiger Wertetabellenbereich: <[t1,t2]> Bestimme Wertetabellenbereich autom.!` Stufe: 0

  Entsprechend den beiden zuvor erläuterten Warnungen gilt auch hier beim Erscheinen dieser Warnung `t1 = t2`. Die Option `tabrange` bleibt dann ebenfalls ohne Wirkung.

- `Warnung!! 'stepsize' UND 'tabnum' gesetzt. Ignoriere Option 'tabnum'!` Stufe: 1

  Werden `diskus` die beiden Optionen `stepsize` und `tabnum` gleichzeitig übergeben, bleibt, wie schon in Abschnitt 3.3 erwähnt, die Option `tabnum` ohne Auswirkung. Die Anzahl der Wertetabelleneinträge wird in diesem Fall ausschließlich auf Grund des Wertes von `stepsize` bestimmt.

16 Es sei darauf hingewiesen, daß eine Angabe von `xrange=[x1,x2]` mit `x1 > x2` durchaus erlaubt ist. In diesem Fall wird die übergebene Liste von `diskus` in die korrekte Reihenfolge gebracht.

- `Warnung!! Anzahl der zu berechnenden Funktionswerte: <num>`
  Stufe: 2

  Diese Warnung wird ausgegeben, wenn die Anzahl der für die Wertetabelle zu berechnenden Funktionswerte `<num>` einen gewissen Grenzwert überschreitet[17].

- `Warnung!! Fehler aufgetreten in: <procname>`
  `<name> konnten nicht ermittelt werden!!` Stufe: 2

  Diese Warnung wird nur dann ausgegeben, wenn in der genannten Prozedur `<procname>` ein Fehler aufgetreten ist. Durch `<name>` wird hier angegeben, welcher der zu ermittelnden Werte (Definitionslücken, Nullstellen, Extremwerte, Wendestellen, Symmetrieverhalten, Asymptoten oder das Verhalten im Unendlichen) nicht bestimmt werden konnte.

  Ursache für diese Warnung kann z. B. sein, daß die eingegebene Funktion nicht zu der Klasse von Funktionen gehört, welche von der die Warnung verursachenden Prozedur bearbeitet werden kann. Ein weiterer Grund für diese Warnung kann vorliegen, wenn die Funktionsdefinition nicht der Syntax entspricht, die von der in `<procname>` genannten Prozedur erwartet wird.

- `Warnung!! x-Plotbereichsbestimmung nur eingeschraenkt`
  `moeglich. Gewaehlter Bereich: <[x1,x2]>` Stufe: 1

  Wurde für die betrachtete Funktion nur eine kritische Stelle $\mathtt{x}_k$ ermittelt, so kann auf Grund dieses einen Wertes kein sinnvoller x-Plotbereich berechnet werden. In diesem Fall wird ein Defaultbereich gewählt, der den Wert $\mathtt{x}_k$ umschließt. Der so gewählte x-Plotbereich wird zu $[\mathtt{x}_k\mathtt{-xdef}_1,\ \mathtt{x}_k\mathtt{+xdef}_2]$ gesetzt[18].

  Es ist zu beachten, daß sich der durch die Warnung angezeigte x-Plotbereich `<[x1,x2]>` im weiteren Verlauf nochmal ändern kann, falls die betrachtete Funktion in diesem Bereich teilweise undefiniert ist.

- `Warnung!! x-Plotbereichsbestimmung nicht moeglich. Nehme`
  `Defaultwerte: <[x1,x2]>` Stufe: 1

  Liegen der automatischen x-Plotbereichsbestimmung keine Werte vor, anhand derer ein sinnvoller x-Plotbereich bestimmt werden könnte, so wird diese Warnung ausgegeben. Der x-Plotbereich wird dann auf die angezeigten Defaultwerte `<[x1,x2]>` gesetzt.

  Wie zuvor gilt auch hier, daß sich der angegebene x-Plotbereich `<[x1,x2]>` noch einmal ändern kann, falls die betrachtete Funktion in dem gewählten Bereich teilweise undefiniert ist.

---

17 In der vorliegenden Implementierung liegt diese Grenze bei 200.

18 Hier gilt zur Zeit $\mathtt{xdef}_1\mathtt{=xdef}_2$. Die kritische Stelle $\mathtt{x}_k$ erscheint somit im Plot bzgl. der x-Achse zentriert.

- `Warnung!! y-Plotbereichsbestimmung nicht moeglich. Nehme Defaultwerte: <[y1,y2]> (Eventuell Abhilfe durch manuelle x-Plotbereichswahl!)` Stufe: 1

  Diese Warnung kann auftreten, wenn der x-Plotbereich so unglücklich gewählt wurde, daß die Funktion in diesem Bereich undefiniert ist[19]. In diesem Fall muß die automatische Bestimmung des y-Plotbereichs fehlschlagen. Für den y-Plotbereich werden dann die angegebenen Defaultwerte `<[y1,y2]>` gewählt. Abhilfe kann unter Umständen dadurch geschaffen werden, daß der x-Plotbereich manuell über den optionalen Parameter `xrange=<[x1,x2]>` festgelegt wird.

- `Warnung!! Funktion total undefiniert. Plot nicht moeglich.` Stufe: 0

  Wenn sich herausstellt, daß die betrachtete Funktion total undefiniert ist, d. h. für kein $x \in \mathbb{R}$ definiert ist, so macht auch ein Plot dieser Funktion sicherlich wenig Sinn. Wenn nicht ohnehin schon die Plotausgabe durch den optionalen Parameter `noplot=true` unterdrückt wurde, erscheint in diesem Fall die genannte Warnung.

- `Warnung!! Funktion total undefiniert. Wertetabelle nicht moeglich!` Stufe: 0

  Analog zum vorangegangenen Fall ist es natürlich auch überflüssig, die Wertetabelle einer total undefinierten Funktion erstellen zu wollen. Ist die Ausgabe der Wertetabelle nicht über die Option `notab=true` unterbunden, so erscheint diese Fehlermeldung.

- `Warnung!! Fehler beim Plotversuch aufgetreten. Versuche Plot ohne 'discont=true'` Stufe: 2

  Wie bereits in Abschnitt 3.3 erwähnt, kommt es in einigen Fällen beim `plot`-Kommando im Zusammenhang mit der Plot-Option `discont` zu Problemen. Tritt ein solcher Fehler auf, so wird diese Warnung ausgegeben, die Option `discont` auf `false` gesetzt und der Plot der Funktion wiederholt.

- `Warnung!! Fehler beim Plotversuch aufgetreten. Plot nicht moeglich!` Stufe: 0

  Tritt beim Plotversuch trotz deaktivierter `discont`-Option weiterhin ein Fehler innerhalb der `plot`-Routine auf, wird diese Warnung ausgegeben.

---

19 Dies kann z. B. der Fall sein, wenn für den x-Plotbereich die Defaultwerte genommen werden mußten oder aber der x-Plotbereich von außen über den optionalen Parameter `xrange` gesetzt wurde.

3.6 Weiterverwendung der Ergebnisse

Wie schon in Abschnitt 3.4 erwähnt, besteht die Möglichkeit, auf die durch `diskus` ermittelten Ergebnisse auch nach Beendigung von `diskus` zurückzugreifen, da die in einer Maple-Prozedur als global deklarierten Variablen für die gesamte Dauer einer Maple-Session existieren. Der Benutzer kann dadurch mit den Ergebnissen weitere Berechnungen durchführen oder diese einfach nur durch diverse Transformationen, z. B. mittels `simplify`, `expand`, `collect` etc. in eine von ihm gewünschte Form bringen.

Von `diskus` werden nun eine Reihe solcher globaler Variablen, welche die berechneten Resultate beinhalten, zur Verfügung gestellt. Bei diesen Variablen handelt es sich grundsätzlich um zweidimensionale Arrays mit 4 Zeilen und n Spalten. In einer Spalte stehen dabei, sofern vorhanden, die folgenden vier Werte: symbolischer x-Wert, symbolischer Funktionswert, numerischer x-Wert und numerischer Funktionswert[20]. Die insgesamt n Spalten entsprechen den n ermittelten unterschiedlichen x-Werten pro Aufgabenstellung, d. h., wurden z. B. von `diskus` insgesamt drei verschiedene Maxima gefunden, dann hat das entsprechende Array die Dimensionierung 4×3.

Es ist zu beachten, daß die Felder der Arrays, die die numerischen Werte beinhalten, nicht immer belegt sein müssen. Diese Felder bleiben z. B. dann unbesetzt, wenn es sich bei den beiden zugehörigen symbolischen Werten um ganzzahlige Werte handelt. Ebenso sind natürlich auch nicht in allen Fällen die Felder mit den Funktionswerten belegt, da dies z. B. für Definitionslücken oder Nullstellen keinen Sinn macht. Gleiches gilt auch, falls es sich bei den ermittelten x-Werten um periodische Punkte oder Intervalle handelt.

Um einfacher auf diese bereitgestellten Arrays zugreifen zu können, wurden noch die vier Konstanten[21] `xsym`, `fxsym`, `xnum` sowie `fxnum` definiert. Sie ermöglichen den Zugriff auf die Arrays, ohne daß man sich merken muß, in welcher Zeile welcher Wert zu finden ist. Statt nun z. B. über `fnull[3,1]` den numerischen x-Wert einer ersten Nullstelle zu erfragen, kann man dies auch über `fnull[xnum,1]` erreichen. Entsprechend gelangt man durch die Verwendung der Konstante `xsym` an die symbolischen x-Werte, `fxsym` bezieht sich auf die symbolischen Funktionswerte und bei Verwendung von `fxnum` erhält man die numerischen Funktionswerte. Schematisch haben alle diese Arrays die folgende Struktur:

20 Diese Aussage ist so eigentlich etwas ungenau. An Stelle der symbolischen x-Werte können nämlich auch Intervalle, periodische Punkte, Gleichungen etc. stehen, d. h. auch mengenwertige Ausdrücke. Grundsätzlich sind in der ersten Zeile der Werte-Arrays die von den Diskussionsalgorithmen gelieferten Resultate zu finden, ganz gleich, von welcher Art sie sind.

21 Da Maple V Rel. 3 keine Konstanten kennt, sind dies tatsächlich vier globale Variablen.

Art des Wertes	Zugriff über	Wert 1	...	Wert i	...
symbolischer x-Wert	`xsym`	x_1	...	x_i	...
symbolischer $f(x)$-Wert	`fxsym`	$f(x_1)$	...	$f(x_i)$	...
numerischer x-Wert	`xnum`	x_1	...	x_i	...
numerischer $f(x)$-Wert	`fxnum`	$f(x_1)$	...	$f(x_i)$	...

Insgesamt stehen dem Benutzer die im folgenden aufgeführten 16 Array-Variablen zur Verfügung:

- `fdef` Definitionslücken

  Dieses Array enthält die Informationen über die Definitionslücken in symbolischer und gegebenenfalls auch in numerischer Form. Funktionswerte werden hierbei natürlich nicht berechnet, so daß die entsprechenden Felder dieses Arrays leer bleiben[22].

`fdef`:

Wert 1	...	Wert i	...
x_{def_1}	...	x_{def_i}	...
-	...	-	...
x_{def_1}	...	x_{def_i}	...
-	...	-	...

- `fnull` Nullstellen

  Über dieses Array hat man Zugriff auf die ermittelten Nullstellen der betrachteten Funktion. Eine Berechnung der Funktionswerte erübrigt sich hier ebenfalls.

`fnull`:

Wert 1	...	Wert i	...
x_{null_1}	...	x_{null_i}	...
-	...	-	...
x_{null_1}	...	x_{null_i}	...
-	...	-	...

- `fmax`, `fmin`, `fsattel`, `feunk` Extremwerte

  Über `fmax` erreicht man die symbolischen und numerischen Werte der Maxima sowie die zugehörigen Funktionswerte. Mittels `fmin` erhält man die entsprechenden Werte der Minima und über `fsattel` kann man analog auf die Sattelpunkte zugreifen. In `feunk` werden schließlich alle die Extremstellen gesammelt, die nicht klassifiziert werden konnten. Im Einzelfall kann es auch vorkommen, daß einige Funktionswerte nicht berechnet werden können.

22 Leer bedeutet hier, daß diese Felder uninitialisiert sind. Ein Auslesen eines uninitialisierten Array-Elements führt aber zu keinem Fehler, sondern es wird lediglich der Name des Array-Elements geliefert. So liefert z. B. `fdef[fxsym,1]` das „Ergebnis" $fdef_{2,1}$.

fmax:

Wert 1	...	Wert i	...
x_{max_1}	...	x_{max_i}	...
$f(x_{max_1})$	...	$f(x_{max_i})$	...
x_{max_1}	...	x_{max_i}	...
$f(x_{max_1})$	...	$f(x_{max_i})$	...

- `fpton`, `fntop`, `fwunk` Wendestellen

  Die symbolischen sowie numerischen Werte der PtoN-Wendestellen[23] sind samt der zugehörigen Funktionswerte über das Array `fpton` erreichbar. Analog kann man über `fntop` auf die jeweiligen Werte der NtoP-Wendestellen zugreifen. Nicht klassifizierbare Wendestellen sind über das Array `fwunk` erreichbar.

fpton:

Wert 1	...	Wert i	...
x_{pton_1}	...	x_{pton_i}	...
$f(x_{pton_1})$	...	$f(x_{pton_i})$	...
x_{pton_1}	...	x_{pton_i}	...
$f(x_{pton_1})$	...	$f(x_{pton_i})$	...

- `fachssym`, `fpunktsym` Symmetrieverhalten

  Mittels `fachssym` kann auf eine eventuell ermittelte Symmetrieachse zugegriffen werden, und über `fpunktsym` ist der Symmetriepunkt verfügbar. In beiden Fällen werden natürlich keine Funktionswerte berechnet. Die entsprechenden Felder der Arrays bleiben daher leer.

fachssym:

Wert 1
$x_{achssym_1}$
-
$x_{achssym_1}$
-

- `fasymp_h`, `fasymp_v`, `fasymp_s` Asymptoten

  Waagerechte Asymptoten werden durch das Array `fasymp_h` zur Verfügung gestellt. Über `fasymp_v` erreicht man die senkrechten Asymptoten und das Array `fasymp_s` enthält die Information über berechnete schiefe Asymptoten. Es werden weder Funktionswerte noch numerische Werte berechnet, daher sind in allen drei Fällen die jeweiligen Felder der Arrays leer.

23 Mit PtoN-Wendestellen bezeichnen wir die Wendestellen, bei denen sich die Kurvenkrümmung von positiv zu negativ ändert. Analog ändert sich die Kurvenkrümmung bei NtoP-Wendestellen von negativ zu positiv.

fasymp_h:

Wert 1	...	Wert i	...
$x_{asymp\_h_1}$	...	$x_{asymp\_h_i}$	...
-	...	-	...
-	...	-	...
-	...	-	...

- `finf_l`, `finf_r` Verhalten im Unendlichen

  Das Verhalten der betrachteten Funktion für $x \to -\infty$ ist über `finf_l` abrufbar. Entsprechend enthält das Array `finf_r` die Information über das Verhalten für $x \to \infty$. Funktionswerte und numerische Werte sind auch hier nicht verfügbar. Beide Arrays bestehen nur aus einer Spalte, welche wiederum nur jeweils einen Wert enthalten.

finf_l:

Wert 1
$x_{inf\_l_1}$
-
-
-

4 Details der Implementierung

Das Rahmenprogramm `diskus` ist ebenfalls, wie die zugrundeliegenden Implementierungen der Diskussionsalgorithmen, eine in Maple realisierte Prozedur. Die Quellen dazu finden sich in der Datei `diskus.m53` im Unterverzeichnis `cohrs/`. In dieser Datei sind alle das Rahmenprogramm betreffenden Routinen realisiert.

Die eigentliche Hauptprozedur `diskus` ist vom Ablauf her recht einfach strukturiert. Der Datenfluß erfolgt im wesentlichen linear. Im nachstehenden Grobalgorithmus von `diskus` auf Seite 32 wird dieser Umstand leicht ersichtlich.

Trotz der vergleichsweise einfachen Funktion von `diskus`, die sich auch im Grobalgorithmus niederschlägt, weist die vorliegende Implementierung mittlerweile einen beträchtlichen Umfang auf. Dies hat mehrere Gründe: So erwies sich z. B. eine der Hauptaufgaben von `diskus`, die übersichtliche Ausgabe der ermittelten Ergebnisse, als problematisch, da die von Maple hierfür bereitgestellten Prozeduren zur Ausgabe in ihren Möglichkeiten doch recht beschränkt sind. Folgerichtig nimmt dann auch die Behandlung der Ausgabe in `diskus` einen großen Raum ein.

Eine weitere Hürde war bei der Weiterverarbeitung der von den benutzten Prozeduren gelieferten Ergebnisse zu nehmen. Diese Daten liegen nicht immer in einer zur weiteren algorithmischen Bearbeitung geeigneten Form vor,

Algorithmus 1 Grobalgorithmus von `diskus`

`IN`: elementare Funktion f, Variable x
  Ermittlung und Ausgabe der Definitionslücken
  Ermittlung und Ausgabe der Nullstellen
  Ermittlung und Ausgabe der Extremwerte
  Ermittlung und Ausgabe der Wendestellen
  Ermittlung und Ausgabe des Symmetrieverhaltens
  Ermittlung und Ausgabe der Asymptoten
  Ermittlung und Ausgabe des Verhaltens im Unendlichen
  Bestimmung des x-Plotbereichs
  Bestimmung des y-Plotbereichs
  Ausgabe des Funktionsgraphen inkl. ermittelter Asymptoten
  Erstellung und Ausgabe der Wertetabelle
  Ausgabe der LaTeX-Datei und der PostScript-Datei
`OUT`: Globale Variablen mit den ermittelten Werten

so daß `diskus` hier immer wieder zunächst eine bisweilen umständliche Umsetzung in andere Datenstrukturen vornehmen muß. Zudem brachte auch die Reimplementierung der Prozeduren aus [Ban95] durch [Wär97] Probleme mit sich, da die ursprünglichen Prozeduren zwar vollständig ersetzt, aber dennoch leider nicht überflüssig wurden. Von diesen wird nämlich weiterhin von den in [Klo95] entwickelten Prozeduren Gebrauch gemacht. So liefern zwar die in [Wär97] entstandenen Prozeduren die Ergebnisse in zur Weiterverarbeitung durch `diskus` geeigneteren Datenstrukturen, doch mußten leider aus oben genannten Gründen in vielen Fällen dafür neue zusätzliche Bearbeitungsalgorithmen geschaffen werden. Bei der folgenden Übersicht über die in der Datei `diskus.m53` vorhandenen Prozeduren wird man daher auch des öfteren auf Prozeduren stoßen, die scheinbar doppelt vorhanden sind, einmal in einer Version zur Bearbeitung der Ergebnisse der Prozeduren aus [Ban95] und einmal in der entsprechenden Variante für die Prozeduren aus [Wär97].

Schließlich war es leider auch immer wieder nötig, Inkonsistenzen und Bugs in Maple zu umschiffen. So mußte z. B. eine eigene Prozedur zur korrekten Sortierung einer Liste mit symbolischen Ausdrücken, die auch die Werte ∞ bzw. $-\infty$ enthalten kann, geschrieben werden, da dies das vorhandene `sort`-Kommando in Maple V Rel. 3 nicht zu leisten vermag. Da Maple jedoch von sich aus in der Regel boolesche Vergleiche, welche die Werte ∞ und $-\infty$ enthalten, korrekt zu bewerten weiß, ist das oben geschilderte mangelhafte Verhalten des `sort`-Kommandos nicht vollständig nachvollziehbar. Leider gab es im Verlauf der Erstellung von `diskus` weitere ähnliche Probleme, die in Abschnitt 6 diskutiert werden.

4.1 Prozeduren des Rahmenprogramms

Bevor nun einige Details der Implementierung von `diskus` näher erläutert werden, soll an dieser Stelle zunächst ein Überblick über die weiteren Prozeduren, die in der Datei `diskus.m53` zu finden sind, gegeben werden. Außer der Hauptprozedur `diskus` sind dies noch eine Reihe von Hilfsprozeduren, von denen die meisten jedoch im folgenden nur kurz erwähnt werden sollen, da diese in den Quellen (in der Datei `diskus.m53`) ausführlich kommentiert sind. Einige der hier aufgeführten Prozeduren werden wir in späteren Abschnitten noch näher betrachten.

- Prozeduren zur Plotbereichsbestimmung
  - `get_xrange_sw`: Bestimmung des x-Plotbereichs.
  - `get_maxrange_sw`: Bestimmung des maximalen Definitionsbereiches.
  - `get_yrange_sw`: Bestimmung des y-Plotbereichs.
  - `CalcFunktionswerte`: Berechnung von Funktionswerten.
- Prozeduren zur Erstellung des Plots
  - `do_plot`: Erstellung des Funktionsplots.
- Prozeduren zur Erstellung der Wertetabelle
  - `gen_functab`: Erstellung der Wertetabelle.
  - `print_functab`: Ausgabe der Wertetabelle.
- Prozeduren zur Bildschirmausgabe
  - `printout`: Ausgabe der ermittelten Werte.
  - `printout_sw`: Ausgabe der ermittelten Werte.
  - `print_Punkt_sw`: Ausgabe der einfachen Punkte.
  - `print_P_Punkt_sw`: Ausgabe der periodisch auftretenden Punkte.
  - `print_Intervall_sw`: Ausgabe der Intervalle.
  - `print_P_Intervall_sw`: Ausgabe der periodisch auftretenden Intervalle.
  - `print_Gleichung`: Ausgabe der Gleichungen.
  - `print_groupsep`: Ausgabe eines „großen“ Trennstriches.
  - `print_simplesep`: Ausgabe eines „kleinen“ Trennstriches.
- Prozeduren zur LaTeX-Ausgabe
  - `print_LaTeX_Preamble`: Ausgabe der LaTeX-Präambel.
  - `print_LaTeX_Title`: Ausgabe der Zwischenüberschriften.
  - `print_LaTeX_MathDelim`: Ausgabe eines LaTeX-Environment-Befehls.
  - `print_LaTeX_ValArray`: Ausgabe der ermittelten Werte.
  - `‘latex/Punkt‘`: Ausgabe der einfachen Punkte.
  - `‘latex/P_Punkt‘`: Ausgabe der periodisch auftretenden Punkte.

- `'latex/Intervall'`: Ausgabe der Intervalle.
- `'latex/P_Intervall'`: Ausgabe der periodisch auftretenden Intervalle.
- `latexGleichung`: Ausgabe der Gleichungen.

• Diverse Hilfsprozeduren
  - `comp_yvals_sw`: Berechnung von Funktionswerten.
  - `comp_nums_sw`: Berechnung numerischer Werte.
  - `lessorder`: Hilfsprozedur zum Maple `sort`-Kommando.
  - `lessorder_sw`: Hilfsprozedur zum Maple `sort`-Kommando.
  - `list2array`: Übernahme der Werte einer Liste in ein Array.
  - `list2array_sw`: Übernahme der Werte einer Liste in ein Array.
  - `remove_complex`: Entfernen komplexer Werte aus einer Liste.
  - `remove_multiples`: Entfernen mehrfach auftretender Werte.
  - `remove_absmax`: Entfernen absoluter Maxima.
  - `short_symbolic`: Vereinfachen eines symbolischen Ausdrucks.

4.2 Die automatische Plotbereichsbestimmung

Um die von den Prozeduren des Packages `funcdisc` ermittelten Ergebnisse graphisch in einem Plot zu veranschaulichen, mußte im Rahmenprogramm `diskus` ein Algorithmus zur automatischen Plotbereichsbestimmung realisiert werden. Dabei galt die Maßgabe, daß die in der vorangegangenen automatischen Funktionsdiskussion ermittelten Werte auch im Plot der Funktion zu erkennen sein sollten. Hierzu war ein geeigneter x-Plotbereich zu bestimmen war. Nach einigen Tests mit der `plot`-Routine von Maple V Rel. 3 stellte sich heraus, daß die von Maple durchgeführte automatische Skalierung in y-Richtung nur selten zu anschaulichen Plots führte, so daß auch ein Verfahren zur y-Plotbereichsbestimmung entwickelt und in `diskus` implementiert werden mußte. Die beiden in `diskus` realisierten Algorithmen werden in den folgenden beiden Abschnitten näher erläutert.

4.2.1 Die x-Plotbereichsbestimmung

Wie bereits erwähnt, war bei der Bestimmung des x-Plotbereichs darauf zu achten, daß die für den Funktionsgraphen relevanten Werte, welche sich aus den Definitionslücken sowie den Null-, Extrem- und Wendestellen zusammensetzen, im Plot wiederzufinden sind. Um aber einen Gesamteindruck vom Verlauf der Funktion zu erhalten, reicht es jedoch nicht aus, den x-Plotbereich nur durch das Minimum und Maximum der oben genannten Werte zu definieren. So würden in der Regel wichtige Informationen über den Verlauf der Funktion über die kritischen Stellen hinaus fehlen. Anhand des folgenden Beispiels läßt sich dies gut verdeutlichen.

Beispiel 4.1: Für die Funktion

$$x \mapsto f(x) = \frac{x^2 - x - 1}{x^2 - 5}$$

lassen sich mittels `diskus` insgesamt sieben für die x-Plotbereichsbestimmung relevante Ergebnisse ermitteln. Es sind dies

- zwei Definitionslücken bei

$$\begin{aligned} & x_{def_1} = -\sqrt{5} \approx -2.236067978 \\ \text{und} \quad & x_{def_2} = \sqrt{5} \approx 2.236067978, \end{aligned}$$

- zwei Nullstellen bei

$$\begin{aligned} & x_{null_1} = \frac{1}{2} - \frac{\sqrt{5}}{2} \approx -0.6180339890 \\ \text{und} \quad & x_{null_2} = \frac{1}{2} + \frac{\sqrt{5}}{2} \approx 1.618033989, \end{aligned}$$

- zwei Extremwerte bei

$$\begin{aligned} & x_{max_1} = 4 - \sqrt{11} \approx 0.683375210 \\ \text{und} \quad & x_{min_1} = 4 + \sqrt{11} \approx 7.316624790, \end{aligned}$$

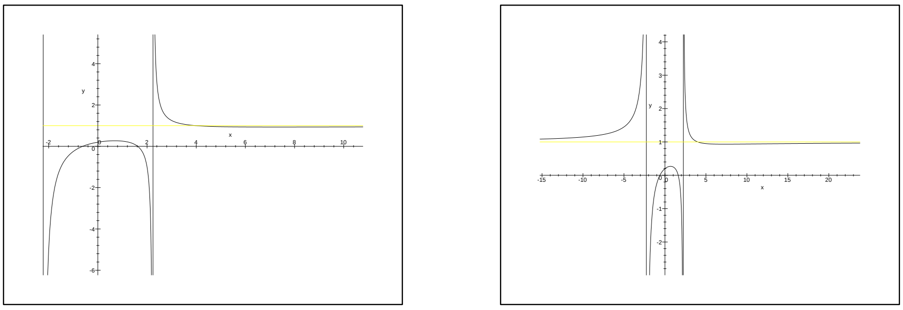
- sowie ein Wendepunkt bei

$$x_{pton_1} = \sqrt[3]{44 + 11\sqrt{5}} + \frac{11}{\sqrt[3]{44 + 11\sqrt{5}}} + 4 \approx 10.78070767.$$

Wählt man nun als Grenzen des x-Plotbereichs das Minimum und Maximum dieser x-Werte, also $x_{def_1} \approx -2.236067978$ und $x_{pton_1} \approx 10.78070767$, so erhält man einen Plot, wie er in Abbildung 3a dargestellt ist. Vergleicht man diesen mit dem tatsächlich von `diskus` zu dieser Funktion erstellten Plot (Abb. 3b), so erkennt man leicht, daß der Funktionverlauf in Abbildung 3a nur ungenügend wiedergegeben wird. △

Der x-Plotbereich war einerseits so groß zu wählen, daß darin alle kritischen Stellen enthalten sind und zudem der Funktionsverlauf deutlich wird, und andererseits so klein, daß die kritischen Stellen noch gut erkennbar bleiben. Es ist aufgrund dieser Vorgaben klar, daß das Minimum und das Maximum der x-Werte der kritischen Stellen lediglich einen minimalen x-Plotbereich festlegen, dieser aber, um einen anschaulichen Plot zu erhalten, sicherlich zu beiden Seiten vergrößert werden muß.

Eine frühe Idee dazu war, die Funktion außerhalb der kritischen Stellen zu bewerten und den x-Plotbereich dynamisch, d. h. in Abhängigkeit vom



(a) (b)

Abbildung 3: Beispiel zur x-Plotbereichsbestimmung

jeweiligen Verlauf der Funktion, zu wählen. Es wäre z. B. denkbar, die Steigungsänderung der Funktion außerhalb der kritischen Stellen mit in die x-Plotbereichsbestimmung einzubeziehen. So hätte man dann den x-Plotbereich umso mehr vergrößern können, je größer die Steigungsänderung ist, und andersherum, je weniger sich die Steigung ändert, desto geringer könnte auch die Vergrößerung des x-Plotbereichs ausfallen. Hier ergab sich allerdings die Frage, wie weit und in welchen Abständen man den Kurvenverlauf über die kritischen Stellen hinaus bewerten sollte, um einen guten x-Plotbereich zu erhalten. Sicherlich wäre es auch angebracht gewesen, die beiden „Seiten“ der Funktion zueinander in Beziehung zu setzen. Einige praktische Tests haben jedoch gezeigt, daß dieser Ansatz selten zu den gewünschten Resultaten führte, was daran liegen mag, daß diese Methode von zu vielen Parametern abhängt.

Der dann implementierte und zunächst nur heuristisch motivierte Algorithmus zur automatischen x-Plotbereichsbestimmung erwies sich in der Praxis trotz seines einfachen Aufbaus als überraschend leistungsfähig und lieferte bei fast allen Tests befriedigende bis gute Ergebnisse. Grundlage für diesen Algorithmus bildet wiederum das von den kritischen Stellen definierte Intervall, welches sich aus dem Minimum $x_{\min}$ und dem Maximum $x_{\max}$ der relevanten Daten ergibt. Die Grenzen des endgültigen x-Plotbereichs (x_{l}, x_{r}) werden durch

$$\begin{aligned} & x_{\mathrm{l}} := x_{\min} - (x_{\max} - x_{\min}) \\ \text{und} \quad & x_{\mathrm{r}} := x_{\max} + (x_{\max} - x_{\min}), \end{aligned}$$

festgelegt, womit der so bestimmte x-Plotbereich (x_{l}, x_{r}) dreimal so groß wie das zugrundeliegende Intervall ($x_{\min}$, $x_{\max}$) ist.

Damit ist der implementierte Algorithmus jedoch nur zum Teil skizziert, gibt es doch noch eine Reihe von Sonderfällen zu beachten. So kann es natürlich vorkommen, daß eine Funktion, wie z. B. $x \mapsto f(x) = x^2$, nur eine kritische Stelle aufweist, womit das obige Verfahren nicht angewendet werden kann. In diesem Fall wird um diesen einen Punkt x_1 ein Defaultbereich gelegt, der in der aktuellen Version durch

$$(x_\mathrm{l},\, x_\mathrm{r}) = (x_1 - x_{\mathrm{def}_l},\, x_1 + x_{\mathrm{def}_r})$$

mit $x_{\mathrm{def}_l} = x_{\mathrm{def}_r} = 2$ definiert ist. Es ist klar, daß die Qualität der so ermittelten x-Plotbereiche stark schwankt und von Funktion zu Funktion verschieden ist. Hier bietet der Algorithmus noch einigen Spielraum für Verbesserungen.

Ein ähnliches Bild ergibt sich, wenn die untersuchte Funktion keine kritischen Stellen aufweist, wie es z. B. bei der Funktion $x \mapsto f(x) = e^x$ der Fall ist, bzw. wenn diese nicht zu ermitteln waren. In diesem Fall stehen dem Algorithmus keine Informationen zur Verfügung, anhand derer sich ein x-Plotbereich festlegen lassen könnte, so daß hier auf einen Defaultbereich zurückgegriffen werden muß. Dieser Defaultbereich ist in der vorliegenden Version von `diskus` durch

$$(x_\mathrm{l},\, x_\mathrm{r}) = (-2,\, 2)$$

gegeben. Auch hier bietet sich sicherlich noch Raum zur Implementierung geeigneterer Verfahren.

Wie bereits erwähnt wurde, gehen in den Algorithmus die Informationen über die Definitionslücken, die Nullstellen sowie die Extrem- und Wendestellen ein. Zur Bestimmung des x-Plotbereichs werden allerdings nicht nur die hier auftretenden einfachen Punkte berücksichtigt, sondern auch die periodisch auftretenden Stellen sowie die einfachen Intervalle.

Um die periodisch wiederkehrenden Punkte mit in die Berechnungen einbeziehen zu können, müssen diese zunächst an einigen Stellen evaluiert werden. Das heißt, daß ein periodischer Ausdruck, wie z. B. $\pi/2 + k\pi$ mit $k \in \mathbb{Z}$, für einige k berechnet wird. In der aktuellen Version erfolgt die Berechnung für $k \in \{-1,\, 0,\, 1\}$. Man erhält somit aus jeder periodisch auftretenden Stelle drei einfache Stellen, was in aller Regel gewährleistet, daß die Periodizität der Funktion im Plot ersichtlich wird.

Bei der Berücksichtigung der einfachen Intervalle – dies können nur Definitionslückenbereiche sein – liegt nicht die Intention im Vordergrund, den x-Plotbereich genügend groß zu wählen, sondern vielmehr, diesen auf eine plausible Größe zu beschränken, denn schließlich macht es ja keinen Sinn, eine Funktion in undefinierten Bereichen plotten zu wollen. So ist z. B. die Funktion $f(x) = \ln(16 - x^2)$ undefiniert für alle x für die $-\infty < x \leq -4$

und $4 \leq x < \infty$ gilt. Nach obigem Algorithmus ergibt sich aber aufgrund der Nullstellen $x_{\mathrm{null}_1} = -\sqrt{15}$ und $x_{\mathrm{null}_2} = \sqrt{15}$ der Plotbereich zu $\left[-3\sqrt{15},\, 3\sqrt{15}\right] \approx [-11.619,\, 11.619]$, d. h., in weiten Teilen dieses Bereiches ist die Funktion gar nicht definiert. Da aber die Darstellung einer Funktion in ihren undefinierten Bereichen keine zusätzlichen Informationen bietet, wird der zunächst ermittelte Plotbereich abschließend noch mit den Definitionslückenbereichen abgeglichen. Dazu wird durch die Prozedur `get_maxrange_sw` der maximale Definitionsbereich der Funktion ermittelt. Dieser maximale Definitionsbereich ergibt sich aus den Daten der Definitionslückenintervalle und definiert sich durch den kleinsten bzw. größten x-Wert, für den die jeweilige Funktion definiert ist[24]. Der bisher bestimmte x-Plotbereich wird dann, wenn sinnvoll, auf den maximalen Definitionsbereich eingeschränkt. Für die obige Funktion $f(x)$ ergibt sich somit der tatsächliche x-Plotbereich zu $[x_{\mathrm{l}},\, x_{\mathrm{r}}] = [-4,\, 4]$. Bei diesem Vorgehen sind natürlich nur die Definitionslückenbereiche von Bedeutung, deren eine Grenze ∞ bzw. $-\infty$ ist. Alle anderen verändern den zunächst bestimmten x-Plotbereich nicht.

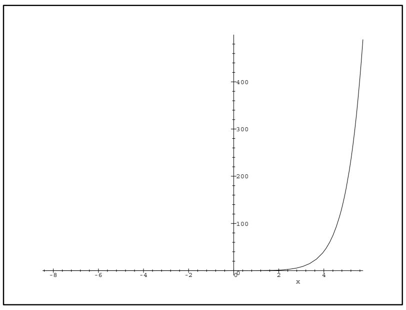
Nachdem damit der implementierte Algorithmus zur automatischen x-Plotbereichsbestimmung hinreichend erläutert wurde, sei abschließend noch erwähnt, daß ein derart einfacher Algorithmus natürlich nicht unter allen Bedingungen Optimallösungen liefern kann, was insbesondere für die auftretenden Sonderfälle gilt. Wie bereits angesprochen, liefert er in der Praxis dennoch zumeist befriedigende Resultate, und zudem hat es sich gezeigt, daß die Qualität des Funktionsplots nicht zuletzt auch entscheidend von der Wahl des y-Plotbereichs abhängt.

4.2.2 Die y-Plotbereichsbestimmung

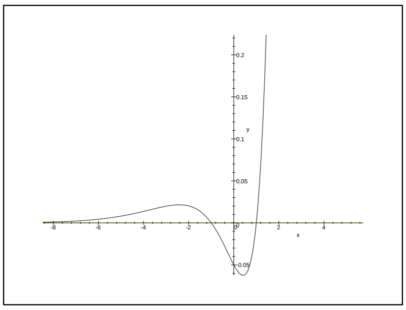
An den Algorithmus zur y-Plotbereichsbestimmung sind, ähnlich wie bereits bei der x-Plotbereichsbestimmung, zwei wesentliche Anforderungen zu stellen, damit der resultierende Plot den Funktionsverlauf aussagekräftig veranschaulicht. Zum einen ist der y-Plotbereich so groß zu wählen, daß auch tatsächlich alle kritischen Stellen im Plot auftauchen und andererseits darf der Bereich nicht zu groß gewählt werden, so daß kritische Stellen im Plot nicht mehr erkennbar sind. Da der vom Maple-Kommando `plot` realisierte Algorithmus zur automatischen y-Plotbereichsbestimmung diese Maßgaben (verständlicherweise) nicht befriedigend berücksichtigt, mußte ein Verfahren gefunden werden, welches den obigen Anforderungen genügt. Zur Veranschaulichung der Problematik sei auf Abbildung 4 hingewiesen, welche zwei Plots der Funktion $x \mapsto f(x) = (x^2 - 1) \cdot e^{x-3}$ zeigt und den Unterschied

24 Bei dieser speziellen Anwendung ist es nicht notwendig, zwischen offenen und geschlossenen Intervallen zu unterscheiden. Für die genannte Funktion f heißt das, daß sich der maximale Definitionsbereich zu $[-4,\, 4]$ ergibt, obwohl die Funktion sowohl für -4 als auch für 4 undefiniert ist.

zwischen einem vom `plot`-Kommando bestimmten y-Plotbereich (Abb. 4a) und einem von `diskus` errechneten y-Plotbereich (Abb. 4b) verdeutlicht.



(a) Von `plot` bestimmter y-Plotbereich

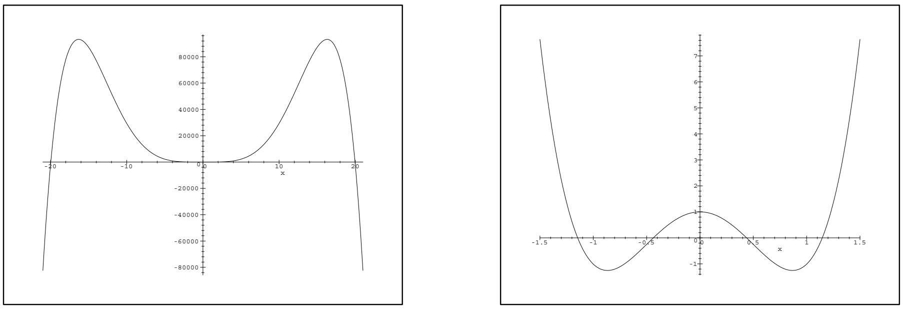


(b) Von `diskus` bestimmter y-Plotbereich

Abbildung 4: Beispiel zur y-Plotbereichsbestimmung für $f(x) = (x^2 - 1) \cdot e^{x-3}$

Es ist klar, daß den oben erwähnten Anforderungen an die y-Plotbereichsbestimmung allein schon theoretische Grenzen im Wege stehen und beide Maßgaben unter Umständen nicht immer gleichzeitig zu erfüllen sind. Dies ist z. B. dann der Fall, wenn ein stark ausgeprägtes Extremum, d. h. eines mit einem betragsmäßig großen Funktionswert, einem nur sehr schwach ausgeprägtem, also betragsmäßig kleinem, gegenübersteht. In diesem Fall führt die Bedingung, daß alle Extrema im Plot dargestellt werden sollen, zwangsläufig dazu, daß das nur schwach ausgeprägte Extremum nicht mehr erkennbar ist. Abbildung 5 verdeutlicht diese Problematik anhand des Polynoms $f(x) = -\frac{1}{100}x^6 + 4x^4 - 6x^2 + 1$. Konzipiert man den Plot wie in Abb. 5a, d. h.so, daß alle fünf Extrema dieser Funktion im Plot enthalten sind, so ergibt sich das Problem, daß drei der fünf Extrema nicht zu erkennen sind, da sie aufgrund ihrer nur schwachen Ausprägung verschwinden. Wie man in Abb. 5b erkennt, werden diese erst dann gut erkennbar, wenn man auf die Darstellung der beiden stark ausgeprägten Extrema verzichtet und den Plotbereich entsprechend einschränkt.

Natürlich sind in diesem Fall die dargestellten Plots unbefriedigend, lassen sie doch beide wichtige Informationen vermissen. Da allerdings die Informationen über die kritischen Stellen auch in schriftlicher Form von `diskus` ausgegeben werden und der Plot lediglich zur Veranschaulichung des allgemeinen Kurvenverlaufs gedacht ist, wurde in `diskus` der Darstellung eines Plots nach Abb. 5a der Vorzug gegeben, d. h., grundsätzlich wird der Plot so angelegt, daß, zumindest theoretisch, alle ermittelten kritischen Stellen im Plot erkennbar sind. Als Idee für spätere Erweiterungen und Verbesserungen



(a) (b)

Abbildung 5: Zwei Plots zu $f(x) = -\frac{1}{100}x^6 + 4x^4 - 6x^2 + 1$

von diskus, wäre es durchaus denkbar in einem solchen Fall mehrere Plots zu erstellen, also etwa einen Gesamtplot, der alle kritischen Stellen enthält und soviele Ausschnittsvergrößerungen wie nötig, um die kritischen Stellen, die im Gesamtplot nicht ausreichend zu erkennen sind, deutlich zu machen.

Nachdem damit die prinzipiellen Probleme bei der y-Plotbereichsbestimmung erläutert worden sind, soll nun näher auf die tatsächliche Implementierung durch die Prozedur get_yrange_sw eingegangen werden. Wie bereits in Abschnitt 3.3 erwähnt wurde, gibt es für den Benutzer mittels des optionalen Parameters yrmethod die Möglichkeit aus insgesamt drei verschiedenen Methoden zur y-Plotbereichsbestimmung zu wählen. Bei diesen drei Methoden handelt es sich jedoch eigentlich nur um drei verschiedene Varianten desselben Algorithmus, da alle drei Varianten in wesentlichen Teilen des Algorithmus übereinstimmen und zwei der drei Varianten nur geringfügige zusätzliche Berechnungen vornehmen.

Grob gesagt basiert der Algorithmus im wesentlichen auf der Berechnung einer gewissen Anzahl von Funktionswerten im zuvor ermittelten x-Plotbereich und der anschließenden, unter Umständen wiederholten Bewertung und Manipulation der so erhaltenen Liste. Die Erstellung der Liste der Funktionswerte erfolgt dabei gleichverteilt über dem zugrundeliegenden x-Plotbereich.

Betrachtet man Abb. 4a, so stellt man fest, daß hier der y-Plotbereich zu groß ausgelegt ist, da aufgrund der starken Steigung der Funktion für $x \to \infty$ die kritischen Stellen unkenntlich werden. Der Algorithmus zur automatischen y-Plotbereichsbestimmung muß nun diese Eigenarten des Funktionsverlaufes erkennen und entsprechend reagieren, d. h.den y-Plotbereich soweit einschränken, daß zwar diese Steigung ersichtlich bleibt aber nicht von den wesentlichen Punkten der Funktion ablenkt.

Aufgrund der gleichverteilt berechneten Funktionswerten ist klar, daß Kurvenabschnitte mit starker Steigung nur durch einige wenige Funktionswerte in der Liste repräsentiert werden. Für die Funktion aus Abb. 4a bedeutet dies aber, daß in der Liste der berechneten Funktionswerte nur einige wenige sehr große Werte vorkommen, während der Großteil der Funktionswerte relativ klein ist und obendrein dicht beieinander liegt. Entfernt man nach und nach den jeweils größten verbleibenden Funktionswert aus der Liste, so erhält man einen immer kleiner werdenden y-Bereich, der von den Werten dieser Liste aufgespannt wird. Das Problem, das hierbei bleibt, liegt in der Bewertung der Liste der Funktionswerte, d. h., es muß ein Maß zur Beurteilung gefunden werden, welches angibt, daß die Liste der Funktionswerte genügend dezimiert wurde.

Ein solches Maß zur Bewertung der Liste der Funktionswerte wurde mit der Funktion `kurtosis` aus Maple's Statistik-Paket gefunden. Diese Funktion bewertet eine gegebene Liste von Werten und liefert eine Maßzahl zurück. Definiert ist diese Funktion als das vierte Moment über dem Mittel dividiert durch die vierte Potenz der Standardabweichung der gegebenen Daten. Anschaulich gesprochen mißt die Funktion `kurtosis` den Grad, wie sehr eine Verteilung „flach“ bzw. „spitz“ ist. Da genauere Betrachtungen zu dieser Funktion den Rahmen dieser Dokumentation sprengen würden, sei hier auf die Online-Hilfe von Maple verwiesen, in der weiterführende Informationen zu finden sind.

In der Prozedur `get_yrange_sw` wird nun diese Funktion `kurtosis` benutzt, um die notwendige Bewertung der Liste der Funktionswerte vorzunehmen. Liefert die Funktion `kurtosis` einen Wert, der größer ist als ein festgelegter Schwellenwert, so wird der absolut gesehen größte Funktionswert aus der Liste herausgestrichen. Die verbleibende Restliste wird anschließend wieder neu bewertet. Dieser Vorgang wird solange wiederholt, bis entweder der Schwellenwert unterschritten ist oder aber in der Liste der Funktionswerte nur noch eine gewisse Mindestanzahl von Elementen verblieben ist. Nach Abbruch dieser Schleife ergibt sich dann der (vorläufige) y-Plotbereich durch das Minimum und das Maximum der verbliebenen Funktionswerte. Dieser y-Plotbereich kann sich nochmals ändern, falls die betrachtete Funktion waagerechte Asymptoten besitzt und diese nicht im soeben ermittelten Bereich enthalten sind. In dem Fall wird der y-Plotbereich wieder soweit vergrößert, daß auch die waagerechten Asymptoten im Plot erscheinen.

Das soeben beschriebene Verfahren bildet die Grundlage für alle drei Varianten der y-Plotbereichsbestimmung und beschreibt gleichzeitig die Default-Variante (`yrmethod=0`). Dieser Algorithmus reicht in der Regel aus, um einen anschaulichen Plot zu erhalten. Es hat sich jedoch in der Praxis gezeigt, daß insbesondere bei Polynomen höheren Grades, d. h. Polynomen mit Grad > 4, der y-Plotbereich häufig dennoch zu groß ausfällt. Aus diesem Grund wurden zusätzlich zwei weitere Varianten des Algorithmus implementiert, welche beim Aufruf von `diskus` durch die Option `yrmethod=1` bzw. `yrmethod=2` ausgewählt werden können. Diese beiden Varianten unterscheiden sich zunächst durch eine andere Wahl des Schwellenwertes und darüber hinaus durch zusätzliche Berechnungen. Dabei wird nach Beendigung der oben beschriebenen Methode der y-Plotbereich aufgrund der Informationen der Funktionswerte an den kritischen Stellen weiter verkleinert.

Ähnlich wie bei der x-Plotbereichsbestimmung, gibt es auch hier gewisse Sonderfälle zu beachten. Bei konstanten Funktionen besteht die Liste der Funktionswerte nur aus einem n-mal wiederholten Wert y_1. Natürlich kann in diesem Fall mit dem obigen Verfahren kein vernünftiger y-Plotbereich bestimmt werden, so daß hier der y-Plotbereich durch $[y_1-2,\ y_1+2]$ festgelegt wird. Ein weiterer seltener Sonderfall ergibt sich immer dann, wenn kein Funktionswert im zugrundeliegenden x-Plotbereich berechnet werden kann. Dies ist z. B. für die Funktion $x \mapsto f(x) = \ln((x-2)(x+2))$ gegeben. Da zu dieser Funktion nicht genügend Informationen zur Wahl eines geeigneten x-Plotbereichs ermittelt werden können, resultiert dieser im Default-Bereich $[-2,\ 2]$. Unglücklicherweise ist jedoch die Funktion $f(x)$ in genau diesem Bereich undefiniert, was zur Folge hat, daß auch kein plausibler y-Plotbereich bestimmt werden kann. Der dann gewählte y-Plotbereich ergibt sich aus dem Default-Bereich $[-100,\ 100]$.

Abschließend ist festzustellen, daß sicherlich auch das implementierte Verfahren zur y-Plotbereichsbestimmung noch recht weit davon entfernt ist, perfekte Ergebnisse zu liefern, was u. a. auch dadurch begründet ist, daß der Algorithmus aus einer vagen Grundidee heraus enstand und dann mittels empirischer Tests weiter verfeinert wurde. Wünschenswert wäre es daher, auch hier eine echte theoretische Grundlage zu schaffen.

Zudem muß gesagt werden, daß es im Falle einer ungenügenden Plotbereichsbestimmung durch `diskus`, dem Benutzer eher anzuraten ist, den Plotbereich durch manuelle Wahl geeigneter Grenzen festzulegen, als wiederholt die verschiedenen Varianten des Algorithmus auszuprobieren.

Wie bereits schon in Abschnitt 3.3 erwähnt, muß man den derzeitigen Stand der Plotbereichsbestimmung sowohl in x- als auch in y-Richtung als heuristisch ansehen. So scheint es z. B. nach den bisherigen Erfahrungen durchaus überlegenswert ein kombiniertes Verfahren zur Plotbereichsbestimmung, d. h. die Bestimmung des Plotbereichs in x- und in y-Richtung in wechselseitiger Abhängigkeit, anzudenken. Auf der anderen Seite zeigt jedoch die Praxis, daß in der Mehrzahl der bisher getesteten Fälle diese Algorithmen, trotz ihrer mangelnden theoretischen Grundlage, zu durchaus akzeptablen Plots führen. Dies gilt umso mehr, wenn man berücksichtigt, daß die von `diskus` gelieferten Plots selbst in ungünstigen Fällen immer noch einen zumindest groben Überblick über die Beschaffenheit der betrachteten Funktionen bieten und somit durchaus eine gewisse Hilfe darstellen.

5 Anwendungsbeispiele

In diesem Abschnitt sollen nun anhand einiger Beispiele die Fähigkeiten aber auch die Grenzen der Prozeduren des Packages `funcdisc` aufgezeigt werden. Dazu werden zunächst Funktionen betrachtet, die sich im wesentlichen problemlos mit `diskus` diskutieren lassen. Es sind dies eine gebrochen rationale Funktion, ein FOE-Term[25] sowie zwei exakt-invertierbare Funktionen. Anhand von drei weiteren Beispiele, einer irrationalen Funktion, einer periodischen Funktion und einem „einfachen“ Polynom, werden anschließend einige grundsätzliche Probleme bei der symbolischen Diskussion elementarer Funktion aufgezeigt.

Die Resulate zu den nachfolgenden Beispielen wurden den durch `diskus` erstellten LaTeX-Ausgaben entnommen. Zum Zwecke der Integration in dieses Dokument wurden diese jedoch formal, aber nicht inhaltlich, abgeändert. Im Anhang finden sich drei Beispiele für das Aussehen der Originalausgaben von `diskus`. Zwei der Beispiele veranschaulichen die verschiedenen Bildschirmausgaben von `diskus`, einmal den Prettyprint-Modus und einmal den Textmodus. Dazu wurden die entsprechenden Maple-Sessions direkt aus Maple als LaTeX-Datei gespeichert. Ein weiteres Beispiel zeigt dann eine originale und unveränderte, von `diskus` erstellte LaTeX-Ausgabe.

Beispiel 5.1: Diskussion der gebrochen rationalen Funktion

$$x \mapsto f(x) = \frac{x^2 - 3\,x - 4}{x + 2}.$$

Diese Funktion kann von `diskus` vollständig und korrekt diskutiert werden. Ausgegeben werden die Definitionslücke

$$x_{def_1} = -2,$$

die Nullstellen

$$\begin{aligned} & x_{null_1} = -1 \\ \text{und} \quad & x_{null_2} = 4, \end{aligned}$$

die Extrema

$$\begin{aligned} & (x_{max_1}, y_{max_1}) = \left(-2 - \sqrt{6},\ -7 - 2\sqrt{6}\right) \\ & \qquad \approx (-4.449489743,\ -11.89897949) \\ \text{und} \quad & (x_{min_1}, y_{min_1}) = \left(-2 + \sqrt{6},\ -7 + 2\sqrt{6}\right) \\ & \qquad \approx (0.449489743,\ -2.101020514)\,, \end{aligned}$$

25 Ein FOE-Term ist eine Funktion der Form $x \mapsto f(x) = p(x) \cdot e^{q(x)}$ mit reellwertigen Polynomen $p(x)$ und $q(x)$. Eine genauere Definition und die Beschreibung der Eigenschaften dieser Funktionenklasse findet man in [Coh97], Abschnitt 5.

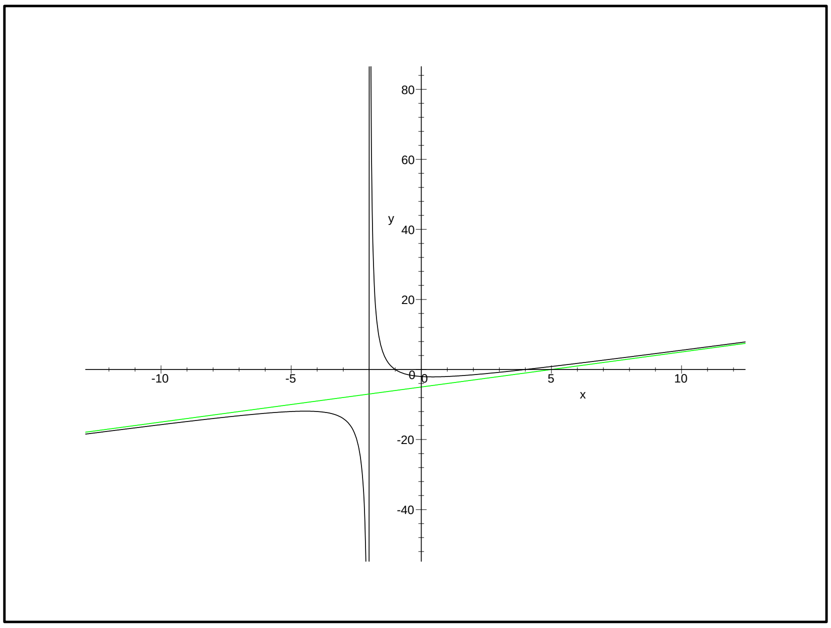


Abbildung 6: Plot zu $f(x) = \dfrac{x^2 - 3\,x - 4}{x + 2}$

der Symmetriepunkt

$$(x_{punktsym_1},\, y_{punktsym_1}) = (-2,\, -7)$$

die Asymptoten

$$\begin{aligned} & x = -2 && \text{(senkrechte Asymptote)} \\ \text{und}\quad & y = x - 5 && \text{(schiefe Asymptote)} \end{aligned}$$

sowie das Verhalten im Unendlichen

$$\begin{aligned} & \lim_{x\to-\infty} f(x) = -\infty \\ \text{und}\quad & \lim_{x\to+\infty} f(x) = \infty. \end{aligned}$$

Die Funktion besitzt keine Wendepunkte und keine waagerechte Asymptote und ist nicht achsensymmetrisch, so daß von `diskus` bei diesen Punkten eine entsprechende Meldung ausgegeben wird. Die Wertetabelle, auf deren Wiedergabe hier verzichtet werden soll, sowie der zugehörige Funktionsplot, zu sehen in Abbildung 6, runden die Ausgabe ab. △

Beispiel 5.2: Diskussion des FOE-Terms

$$x \mapsto f(x) = \left(x^2 - 1\right) e^{x-3}.$$

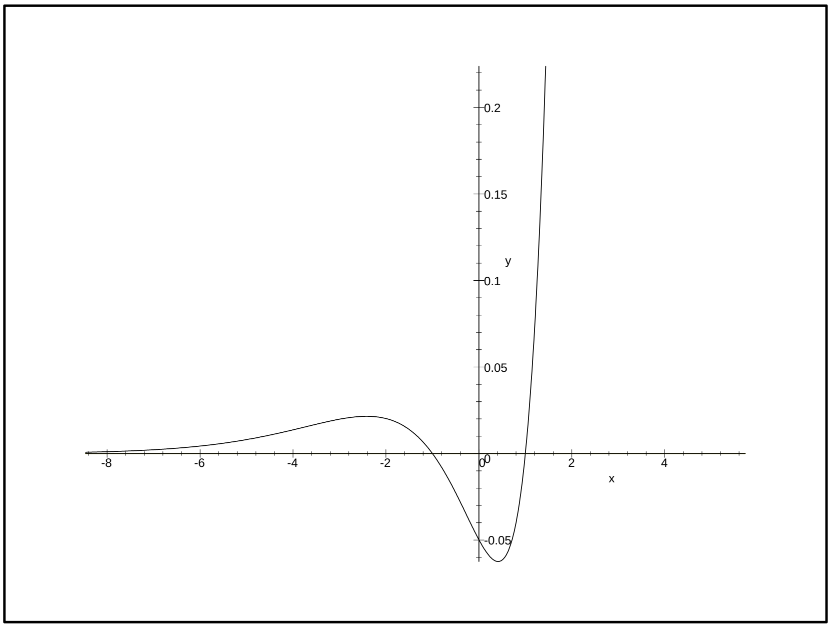


Abbildung 7: Plot zu $f(x) = \left(x^2 - 1\right) e^{x-3}$

Auch diese Funktion erschließt sich der Diskussion mittels `diskus` völlig. Der zu dieser Funktion von `diskus` automatisch erstellte Plot ist in Abbildung 7 zu sehen. Die ermittelten Resultate umfassen die beiden Nullstellen

$$\begin{aligned} & x_{null_1} = -1 \\ \text{und} \quad & x_{null_2} = 1, \end{aligned}$$

die Extrema

$$\begin{aligned} & (x_{max_1}, y_{max_1}) = \left(-1 - \sqrt{2},\, \sqrt{2}\left(2 + \sqrt{2}\right) e^{-4-\sqrt{2}}\right) \\ & \qquad \approx (-2.414213562,\, 0.02150020530) \\ \text{und} \quad & (x_{min_1}, y_{min_1}) = \left(\sqrt{2} - 1,\, \left(\sqrt{2} - 2\right) \sqrt{2} e^{\sqrt{2}-4}\right) \\ & \qquad \approx (0.414213562,\, -0.06241105574)\,, \end{aligned}$$

die zwei Wendestellen

$$\begin{aligned} (x_{pton_1}, y_{pton_1}) &= \left(-2-\sqrt{3},\, 2\left(3+2\sqrt{3}\right)e^{-5-\sqrt{3}}\right) \\ &\approx (-3.732050808,\, 0.01541152634) \\ \text{und} \quad (x_{ntop_1}, y_{ntop_1}) &= \left(-2+\sqrt{3},\, -2\left(-3+2\sqrt{3}\right)e^{-5+\sqrt{3}}\right) \\ &\approx (-0.267949192,\, -0.03535011046)\,, \end{aligned}$$

die waagerechte Asymptote

$$y = 0$$

und das Verhalten im Unendlichen

$$\begin{aligned} \lim_{x\to-\infty} f(x) &= 0 \\ \text{und} \quad \lim_{x\to+\infty} f(x) &= \infty. \end{aligned}$$

△

Beispiel 5.3: Diskussion der exakt-invertierbaren Funktion

$$x \mapsto f(x) = \operatorname{arcsinh}\left(\frac{x^2-x-1}{x^2+x+1}\right)$$

Bei dieser ansonsten für die symbolische Funktionsdiskussion besonders geeigneten Klasse von Funktionen (vgl. dazu auch [Ger92] und [Wär97]) wird es im allgemeinen nicht möglich sein, die Wendestellen symbolisch zu ermitteln. Dies hat seinen Grund darin, daß die zweite Ableitung einer geschachtelten Funktion in der Regel eine additive Struktur aufweist, weshalb von dieser im allgemeinen die Nullstellen nicht mehr symbolisch zu bestimmen sind. Wie man an diesem Beispiel jedoch sieht, werden alle weiteren Informationen wunschgemäß ermittelt. Dazu gehören die beiden Nullstellen

$$\begin{aligned} x_{null_1} &= 1/2 - \frac{\sqrt{5}}{2} \\ &\approx -0.6180339890 \\ \text{und} \quad x_{null_2} &= \frac{\sqrt{5}}{2} + 1/2 \\ &\approx 1.618033989, \end{aligned}$$

die zwei Extrema

$$\begin{aligned} (x_{max_1}, y_{max_1}) &= (-2,\, \operatorname{arcsinh}(5/3)) \\ &\approx (-2.0,\, 1.283795663) \\ \text{und} \quad (x_{min_1}, y_{min_1}) &= (0,\, -\operatorname{arcsinh}(1)) \\ &\approx (0,\, -0.8813735870)\,, \end{aligned}$$

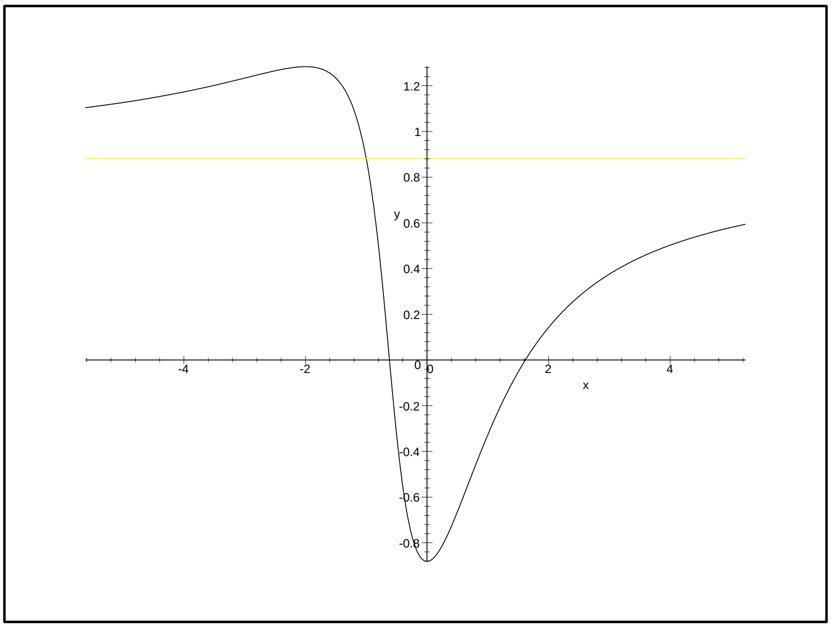


Abbildung 8: Plot zu $f(x) = \operatorname{arcsinh}\left(\frac{x^2 - x - 1}{x^2 + x + 1}\right)$

die waagerechte Asymptote bei

$$y = \operatorname{arcsinh}(1)$$

sowie das Verhalten im Unendlichen

$$\begin{aligned} &\lim_{x \to -\infty} f(x) = \operatorname{arcsinh}(1) \\ \text{und} \quad &\lim_{x \to +\infty} f(x) = \operatorname{arcsinh}(1). \end{aligned}$$

Im Plot zu dieser Funktion, dargestellt in Abbildung 8, erkennt man dann auch, daß diese Funktion tatsächlich Wendestellen besitzt, welche jedoch durch `diskus` nicht zu ermitteln waren. $\triangle$

Beispiel 5.4: Diskussion der exakt-invertierbaren Funktion

$$x \mapsto f(x) = \cosh(\ln(16 - x^2))$$

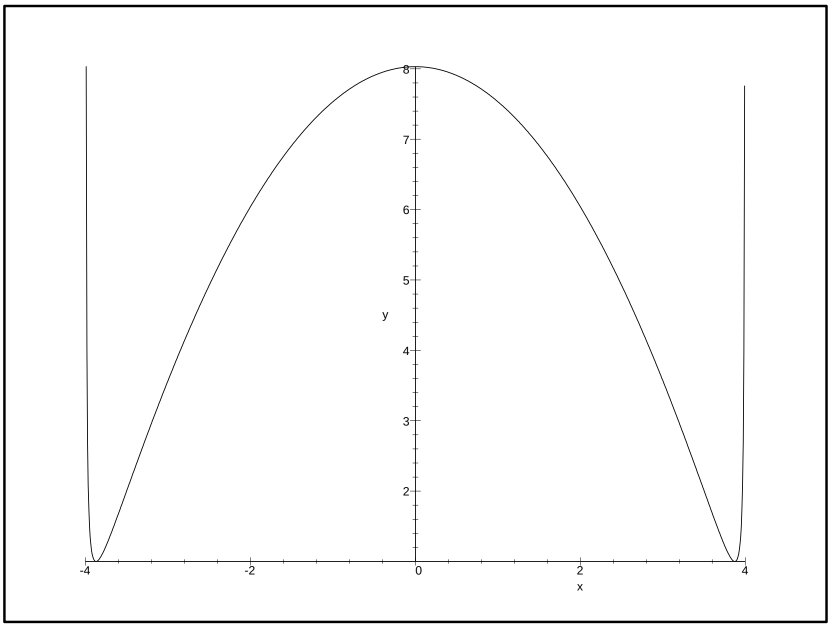


Abbildung 9: Plot zu $f(x) = \cosh(\ln(16 - x^2))$

Wie schon im letzten Beispiel, können auch für diese Funktion die Wendestellen nicht bestimmt werden. Ermittelt werden können jedoch die Definitionslückenintervalle

$$\begin{aligned} & \mathcal{M}_{def_1} = \{x \in \mathbb{R} \mid -\infty \leq x < -4\} \\ \text{und} \quad & \mathcal{M}_{def_2} = \{x \in \mathbb{R} \mid 4 \leq x < \infty\}, \end{aligned}$$

die Extrema

$$\begin{aligned} (x_{max_1}, y_{max_1}) &= (0, \cosh(\ln(16))) \\ &\approx (0, 8.031249998)\,, \\ (x_{min_1}, y_{min_1}) &= \left(-\sqrt{15}, 1\right) \\ &\approx (-3.872983346, 1.0) \\ \text{und} \quad (x_{min_2}, y_{min_2}) &= \left(\sqrt{15}, 1\right) \\ &\approx (3.872983346, 1.0)\,, \end{aligned}$$

die Symmetrieachse bei

$$x_{achssym_1} = 0$$

sowie die beiden senkrechten Asymptoten bei

$$\begin{aligned} x &= -\sqrt{16} \\ \text{und} \quad x &= \sqrt{16}. \end{aligned}$$

Das Verhalten im Unendlichen ist in beiden Fällen undefiniert. Der von `diskus` erstellte Plot zu dieser Funktion ist in Abbildung 9 zu sehen.

Das Bemerkenswerte an dieser Funktion ist, daß sie funktional äquivalent zur echt gebrochen rationalen Funktion

$$x \mapsto \frac{16 - x^2 + \frac{1}{16-x^2}}{2} = \frac{\left(16 - x^2\right)^2 + 1}{2 \cdot (16 - x^2)}$$

ist, auf ihrem Definitionsbereich sogar mit dieser übereinstimmt. Solch ein Phänomen wird bisher aus theoretischen Erwägungen[26] durch `funcdisc` nicht berücksichtigt, obwohl es im Beispiel ermöglichen würde, auch die Wendestellen zu berechnen. △

Anhand der nun folgenden Beispiele sollen einige grundsätzliche Probleme bei der symbolischen Diskussion elementarer Funktionen veranschaulicht werden. Im ersten Beispiel wird der Versuch unternommen eine irrationale Funktion der Form $f(x) \cdot g(x)$ zu diskutieren. Leider ist festzustellen, daß in diesem Fall nur wenige Aspekte der Kurvendiskussion erfolgreich bearbeitet werden können. Im letzten Beispiel wird gezeigt, daß auch ein einfaches Polynom zu unlösbaren Problemen führen kann und auch hier die Diskussion nur eingeschränkt möglich ist.

Beispiel 5.5: Diskussion der irrationalen Funktion

$$x \mapsto f(x) = \operatorname{arcsinh}(x^2 - 5\,x - 1)\sqrt{x}.$$

Aufgrund der Struktur dieser Funktion sind weder Extremwerte noch Wendestellen zu ermitteln. Hier beschränkt sich die Ausgabe von `diskus` lediglich auf die Definitionslücken

$$\mathcal{M}_{def_1} = \{x \in \mathbb{R} \mid -\infty < x < 0\},$$

26 Funktionale Äquivalenz ist für elementare Funktionen im allgemeinen nicht entscheidbar (vgl. dazu auch [Cav70]), so daß hier geeignete Teilklassen gefunden werden müßten.

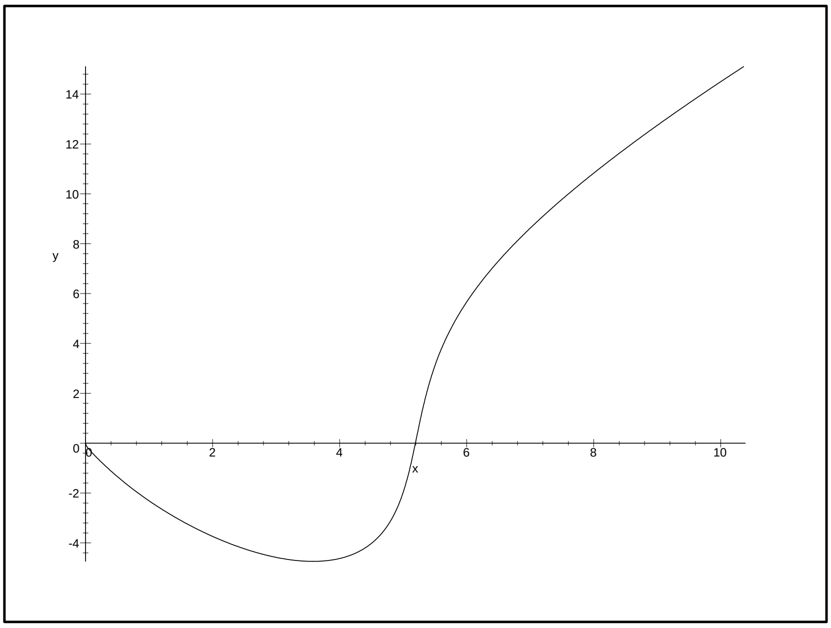


Abbildung 10: Plot zu $f(x) = \operatorname{arcsinh}(x^2 - 5\,x - 1)\sqrt{x}$

die beiden Nullstellen

$$\begin{aligned} x_{null_1} &= 0 \\ \text{und} \quad x_{null_2} &= 5/2 + \frac{\sqrt{29}}{2} \\ &\approx 5.192582404, \end{aligned}$$

sowie das Verhalten im Unendlichen

$$\begin{aligned} \lim_{x\to-\infty} f(x) &= \mathit{undefined} \\ \text{und} \quad \lim_{x\to+\infty} f(x) &= \infty. \end{aligned}$$

Wenn auch die exakten Angaben über das vorhandene Extremum und die Wendestellen fehlen, so sind diese zumindest im Plot der Funktion, welcher in Abbildung 10 gezeigt ist, zu erkennen. △

Beispiel 5.6: Diskussion der periodischen Funktion

$$x \mapsto f(x) = \sin(x)\cos(x)$$

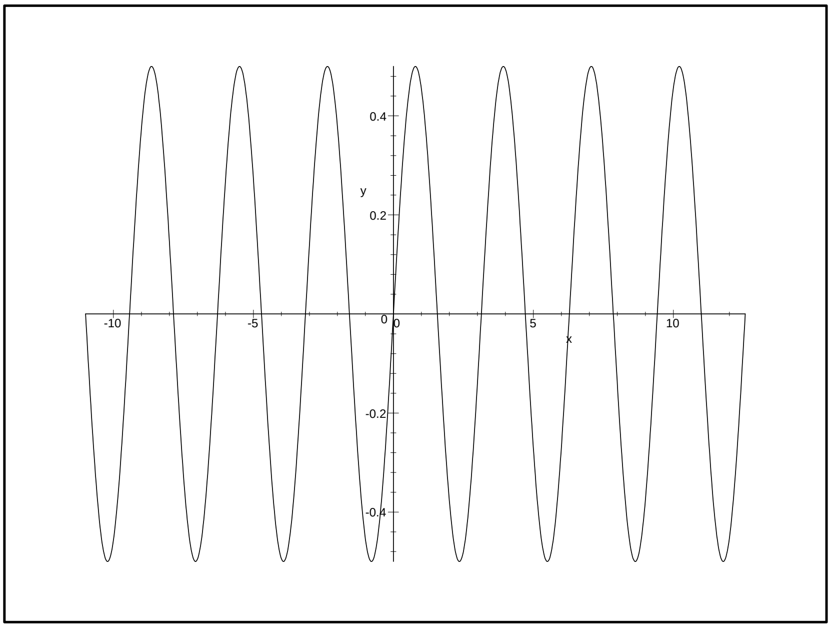


Abbildung 11: Plot zu $f(x) = \sin(x)\cos(x)$

Grundsätzlich sind die meisten der im Package `funcdisc` enthaltenen Prozeduren fähig, auch periodische Funktionen zu bearbeiten. Jedoch ergeben sich bei periodischen Funktionen gewisse Probleme, die unter Umständen eine vollständige Diskussion verhindern. So kann `diskus` zu obiger Funktion lediglich die beiden Mengen der Nullstellen

$$\begin{aligned} &\mathcal{M}_{null_1} = \left\{x \in \mathbb{R} \mid x = \frac{\pi}{2} + k\pi \text{ mit } k \in \mathbb{Z}\right\} \\ \text{und} \quad &\mathcal{M}_{null_2} = \{x \in \mathbb{R} \mid x = k\pi \text{ mit } k \in \mathbb{Z}\} \end{aligned}$$

ermitteln. Die sicherlich vorhandenen Extrem- sowie Wendestellen (siehe dazu auch den Plot zu f, dargestellt in Abbildung 11) werden bisher nicht bestimmt. Dies ist auch nicht weiter verwunderlich, handelt es sich doch bei der Ableitung f' um die additive Funktion

$$x \mapsto f'(x) = \cos(x)^2 - \sin(x)^2,$$

von welcher die symbolische Nullstellenbestimmung durch `diskus` nicht möglich ist. Andererseits läßt sich $f'(x)$ äquivalent umformen zu

$$f'(x) = 2\cos(x)^2 - 1$$

was sich in Maple ausdrücken läßt durch

```
df := (x->2*x^2-1)@cos;.
```

Es handelt sich damit also um eine geschachtelte Funktion, deren Nullstellenbestimmung kein Problem darstellt. Wendet man `diskus` auf diese Funktion `df` an, so erhält man dann auch die Nullstellen

$$\begin{aligned}
\mathcal{M}_{null_1} &= \left\{x \in \mathbb{R} \mid x = \frac{3\,\pi}{4} + 2\,k\pi \text{ mit } k \in \mathbb{Z}\right\}, \\
\mathcal{M}_{null_2} &= \left\{x \in \mathbb{R} \mid x = -\frac{3\,\pi}{4} + 2\,k\pi \text{ mit } k \in \mathbb{Z}\right\}, \\
\mathcal{M}_{null_3} &= \left\{x \in \mathbb{R} \mid x = -\frac{\pi}{4} + 2\,k\pi \text{ mit } k \in \mathbb{Z}\right\} \\
\text{und} \quad \mathcal{M}_{null_4} &= \left\{x \in \mathbb{R} \mid x = \frac{\pi}{4} + 2\,k\pi \text{ mit } k \in \mathbb{Z}\right\},
\end{aligned}$$

welche ja die potentiellen Extremwerte der Ausgangsfunktion f darstellen, sowie darüber hinaus auch noch die Extremwerte

$$\begin{aligned}
\mathcal{M}_{max_1} &= \{x \in \mathbb{R} \mid x = \pi + 2\,k\pi \text{ mit } k \in \mathbb{Z}\}, \\
\mathcal{M}_{max_2} &= \{x \in \mathbb{R} \mid x = 2\,k\pi \text{ mit } k \in \mathbb{Z}\}, \\
\mathcal{M}_{min_1} &= \left\{x \in \mathbb{R} \mid x = \frac{3\,\pi}{2} + 2\,k\pi \text{ mit } k \in \mathbb{Z}\right\} \\
\text{und} \quad \mathcal{M}_{min_2} &= \left\{x \in \mathbb{R} \mid x = \frac{\pi}{2} + 2\,k\pi \text{ mit } k \in \mathbb{Z}\right\}
\end{aligned}$$

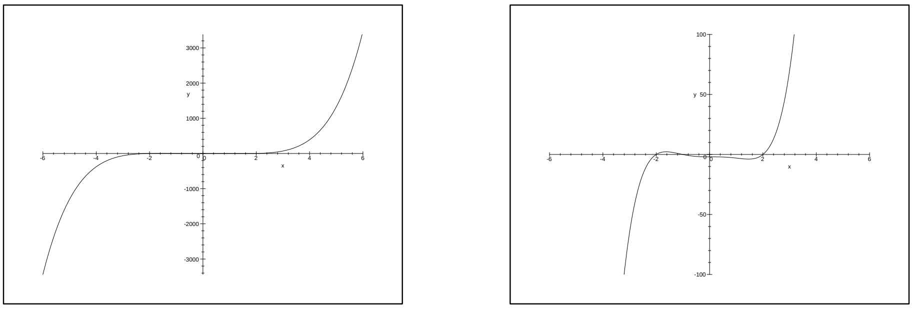
die natürlich nichts anderes als die möglichen Wendestellen von f sind.

Wie man an diesem Beispiel gesehen hat, hängt die Berechnung der kritischen Stellen unter Umständen auch von der Darstellung der Funktion ab. So kann es mitunter gelingen, eine Funktion durch eine oder mehrere Äquivalenzumformungen[27] in eine Form zu bringen, welche durch die Prozeduren des Packages `funcdisc` bewältigt werden kann. Das Problem an dieser Stelle bleibt allerdings wie oben, eine Methodik zur algorithmischen Durchführung der notwendigen Äquivalenzumformungen zu finden. Das beinhaltet zum einen die automatische Erkennung der Notwendigkeit von Äquivalenzumformungen und zum anderen das Finden einer geeigneten Sequenz von Äquivalenzumformungen, sprich die Bestimmung einer geeigneten Normalform △

27 Für dieses Beispiel leistete z. B. die Maple-Funktion `simplify` die nötige Äquivalenzumformung.

Beispiel 5.7: Diskussion des Polynoms

$$f(x) = \frac{1}{2}\left(x^3 + 1\right)\left(x^2 - 4\right).$$



(a) `yrmethod=0` **(b)** `yrmethod=2`

Abbildung 12: Automatisch erstellte Plots zu $f(x) = \frac{1}{2}\left(x^3 + 1\right)\left(x^2 - 4\right)$

Obgleich es sich bei dieser Funktion um ein vermeintlich einfaches Polynom handelt, muß man leider feststellen, daß bisher nicht alle kritischen Stellen durch **`diskus`** zu ermitteln sind. Ausgegeben werden nämlich lediglich die Nullstellen

$$\begin{aligned} x_{null_1} &= -2, \\ x_{null_2} &= -1 \\ \text{und} \quad x_{null_3} &= 2, \end{aligned}$$

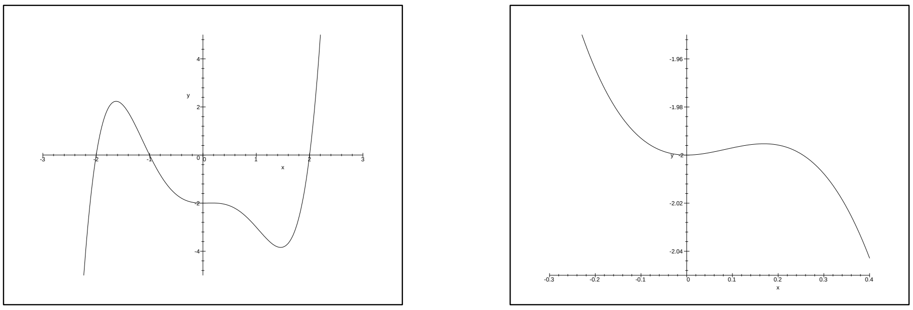
ein Minimum bei

$$(x_{min_1}, y_{min_1}) = (0, -2)$$

sowie das Verhalten im Unendlichen

$$\begin{aligned} \lim_{x\to-\infty} f(x) &= -\infty \\ \text{und} \quad \lim_{x\to+\infty} f(x) &= \infty. \end{aligned}$$

Die weiteren Extrema und die vorhandenen Wendestellen werden jedoch nicht ermittelt. Darüber hinaus ergibt sich erschwerend das Problem, daß bei dieser Funktion auch die automatische Plotbereichsbestimmung kein zufriedenstellendes Ergebnis liefert. In der Default-Einstellung (`yrmethod=0`) erhält man den in Abbildung 12a dargestellten Plot. Aufgrund der unglücklichen Wahl des y-Plotbereichs könnte man daraus zunächst einen Sattelpunkt in der Region um $x = 0$ vermuten. Das ausgegebene Minimum bei $x_{min_1} = 0$ ist in diesem Plot jedoch nicht nachvollziehbar. Der durch die Option `yrmethod=2` enstehende Plot, dargestellt in Abb. 12b, bietet zwar schon ein besseres Bild der Funktion, allerdings ist das angegebene Minimum immer noch nicht als solches erkennbar. Immerhin sind bei diesem Maßstab bereits zwei weitere Extrema, die in der ersten Ausgabe fehlten, vage zu erkennen. Doch erst eine weitere, manuell vorgenommene, Einschränkung des Plotbereichs zeigt ein akzeptables Bild der Funktion, wie es in Abbildung 13a zu sehen ist. Hier läßt sich zwar ansatzweise das angegebene Minimum bei $x_{min_1} = 0$ erahnen, jedoch wird dieses erst in einer Ausschnittsvergrößerung (Abbildung 13b) wirklich gut erkennbar, wobei gleichzeitig ein weiteres, bisher nicht in Erscheinung getretenes, Maximum sichtbar wird.



(a) Gesamtplot **(b)** Ausschnittsvergrößerung

Abbildung 13: Manuell erstellte Plots zu $f(x) = \frac{1}{2}\left(x^3+1\right)\left(x^2-4\right)$

Nachdem nun die Tücken dieser Funktion, zumindest bzgl. der Plotproblematik, aufgezeigt wurden, bleibt natürlich die Frage, wieso die Extrema und Wendestellen dieses vergleichsweise einfachen Polynoms durch `diskus` nicht ermittelt wurden. Das Problem an dieser Stelle wird deutlich, wenn man die Funktion in Maple „von Hand" diskutiert. Es ergeben sich nämlich dann die Nullstellen der ersten Ableitung

$$f'(x) = \frac{5}{2}x^4 - 6x^2 + x$$

zu

$$\begin{aligned}
x_1 =\ & 0\,,\\
x_2 =\ & \left(-\frac{1}{5}+\frac{1}{25}i\sqrt{295}\right)^{1/3}+\frac{4}{5}\left(\frac{1}{\left(-\frac{1}{5}+\frac{1}{25}i\sqrt{295}\right)^{1/3}}\right),\\
x_3 =\ & -\frac{1}{2}\left(-\frac{1}{5}+\frac{1}{25}i\sqrt{295}\right)^{1/3}-\frac{2}{5}\frac{1}{\left(-\frac{1}{5}+\frac{1}{25}i\sqrt{295}\right)^{1/3}}\\
& +\frac{1}{2}i\sqrt{3}\left(\left(-\frac{1}{5}+\frac{1}{25}i\sqrt{295}\right)^{1/3}-\frac{4}{5}\frac{1}{\left(-\frac{1}{5}+\frac{1}{25}i\sqrt{295}\right)^{1/3}}\right) \quad \text{und}\\
x_4 =\ & -\frac{1}{2}\left(-\frac{1}{5}+\frac{1}{25}i\sqrt{295}\right)^{1/3}-\frac{2}{5}\frac{1}{\left(-\frac{1}{5}+\frac{1}{25}i\sqrt{295}\right)^{1/3}}\\
& -\frac{1}{2}i\sqrt{3}\left(\left(-\frac{1}{5}+\frac{1}{25}i\sqrt{295}\right)^{1/3}-\frac{4}{5}\frac{1}{\left(-\frac{1}{5}+\frac{1}{25}i\sqrt{295}\right)^{1/3}}\right).
\end{aligned}$$

Obwohl es sich bei all diesen Ausdrücken um reelle Zahlen handeln muß, ist es Maple nicht möglich, die in diesen Ausdrücken enthaltenen komplexen Terme aufzulösen, um so die Ausdrücke weiter zu vereinfachen. Für die Nullstellen-Prozedur stellen sich die Werte x_2 bis x_4 damit also zunächst als komplexe Zahlen dar und werden deshalb nicht als Lösungen betrachtet. Selbst die numerische Evaluierung der obigen Werte hilft hier nicht weiter, denn wendet man die Maple-Funktion `evalf` auf diese Ausdrücke an, so erhält man

$$\begin{aligned}
& x_1 = 0,\\
& x_2 = 1.457958786 + 0.1\cdot 10^{-9}i,\\
& x_3 = -1.626624721\\
\text{und}\quad & x_4 = 0.1686659349 - 0.2\cdot 10^{-9}i.
\end{aligned}$$

Immer noch werden zwei der vier eigentlich reellen Zahlen als komplexwertig ausgegeben, ein Effekt, der auch durch die Erhöhung der Rechengenauigkeit nicht ausgeschaltet werden kann. Eine Beurteilung, ob es sich bei den ermittelten Werten um Extremwerte handelt ist nur für x_1 zweifelsfrei möglich. Die anderen Werte können nicht sicher als Lösungen der Extremwertberechnung angesehen werden. △

6 Erfahrungen und Ausblick

Die mehrjährige Entwicklung des Programmpakets `diskus` ermöglichte es uns, einen tieferen Einblick sowohl in den praktischen Umgang mit einem multifunktionellen Computeralgebrasystem wie Maple, als auch in weitreichende Probleme der algorithmischen Analysis zu gewinnen. Einige unserer Beobachtungen, Erfahrungen und Schlußfolgerungen sollen in diesem letzten Kapitel gesammelt werden.

6.1 Auswahl der zu diskutierenden Funktionen

Die Tatsache, daß für elementare Funktionen keine Normalform existiert, bedingt, wie in der Einleitung schon erwähnt, die Unmöglichkeit eines Algorithmus, welcher zu einer beliebig vorgegebenen elementaren Funktion eine vollständige Funktionsdiskussion durchführt (vgl. [GS96]). Folglich ist man, um befriedigende Ergebnisse zu erzielen, von vornherein dazu gezwungen, sich einzuschränken und geeignete Funktionsklassen zu finden, für die eine algorithmische symbolische Funktionsdiskussion mindestens in weiten Teilen möglich ist.

Einige dieser Klassen (z. B. die der rationalen Funktionen) haben – bisher eher aus praktischen Erwägungen, denn theoretisch begründet – Eingang in den Schulunterricht gefunden und dienen dort als Übungsmaterial. Die Tatsache, daß Computeralgebrasysteme, wie nun auch unser Programm `diskus` demonstriert, natürlich in der Lage sind, diese Funktionen vollständig zu diskutieren, hat daher bereits zu einer lebhaften Diskussion innerhalb der Mathematik-Didaktik geführt, in welcher Form das Thema „Funktionsdiskussion“ im Unterricht noch seine Berechtigung habe (vgl. [SW95]).

Mit den exakt-invertierbaren Funktionen und den durch FOE-Termen beschriebenen Funktionen konnten weitere geeignete Klassen bestimmt werden, welche ebenfalls in `diskus` berücksichtigt sind.

Wünschenswert wäre allerdings neben einer sicherlich noch notwendigen Verbesserung der bereits benutzten Algorithmen eine systematischere Vorgehensweise bei der Auswahl der Funktionsklassen. Im Hinblick auf die konkreten Bedürfnisse der Ingenieurwissenschaften, wie auch auf die angedeuteten Probleme im Bereich der Mathematik-Didaktik, scheint es uns notwendig – soweit möglich – gezielt diejenigen elementaren Funktionen „herauszupräparieren“, welche einer algorithmischen symbolischen Funktionsdiskussion zugänglich sind.

Dabei ist dann auch der von uns bisher weniger intensiv untersuchte Fall zu betrachten, daß in einer zu diskutierenden Funktion eine Kreisfunktion auftritt. Die Tatsache, daß hieraus sehr schnell mehr-parametrige Familien von Null-, Extrem- und Wendestellen oder Gleichungen resultieren, wirft eine Reihe von neuen Problemen auf, welche bisher in `diskus` nur in Ansätzen gelöst sind.

6.2 Die Notwendigkeit hybrid numerisch-symbolischer Methoden

Ein Effekt, der u. a. häufig bei der Diskussion exakt-invertierbarer Funktionen sichtbar wurde und den wir in Beispiel 5.7 demonstriert haben, war, daß Maple zwar polynomiale Gleichungen (auch höheren als 5. Grades) symbolisch auflösen kann, daß die erhaltenen Ergebnisse aber oft eine für Anwender und für die Weiterverarbeitung zu komplizierte Struktur besitzen (z. B. bei vielfach geschachtelten Wurzelausdrücken). Dadurch werden die Ergebnisse in zweierlei Weise „unlesbar": weder ist man in der Lage, solch einen komplizierten symbolischen Ausdruck unmittelbar mit einem konkreten Zahlenwert in Verbindung zu bringen, noch ist es Maple möglich, diesen befriedigend numerisch auszuwerten.

Da unser Ziel die Funktionsdiskussion reeller Funktionen ist, ist es zur weiteren Verarbeitung eines symbolischen Ausdruckes in `diskus` vielfach notwendig, zu entscheiden, ob es sich um einen reellen oder komplexen Wert handelt. Dies gelingt bisher, wie geschildert, weder auf rein symbolischem Wege noch mittels einer puren numerischen Auswertung.

Einfache Experimente mit Maple zeigen im letztgenannten Fall nicht nur, daß die numerisch ermittelten Werte extrem von der voreingestellten Präzision der Arithmetik abhängig sind, sondern auch, daß deren Aussagekraft nicht notwendig durch eine Erhöhung der Nachkommastellen verbessert wird: so kann es vorkommen, daß auch bei der Berücksichtigung weiterer Nachkommastellen fälschlich bei der Auswertung eines reellen Wurzelausdrucks ein nicht verschwindender Imaginärteil angegeben wird[28].

Andererseits ist bisher aber auch selbst im Fall geschachtelter Wurzelausdrücke noch kein Algorithmus bekannt, der in der Lage wäre nur aufgrund symbolischer Manipulationen die angesprochene Frage zu beantworten.

Wir sind inzwischen der Auffassung, daß hier ein typischer Punkt erreicht ist, wo es sich als vorteilhaft erweisen kann, eine Abkehr von den rein symbolischen Verfahren vorzunehmen und auch numerische, sowie hybrid numerisch-symbolische Methoden zu berücksichtigen. Hierbei ist speziell etwa an die Benutzung von Verfahren zur exakten Isolierung reeller und komplexer Nullstellen von Polynomen zu denken (vgl. [HTV84] oder [Mig92]). Die aktuelle Entwicklung in verschiedenen Bereichen, wie z.B. der reellen algebraischen Geometrie, scheint eine solche Vorgehensweise noch weiter zu rechtfertigen[29].

---

28 Interessanterweise kann in manchen Fällen eine Verbesserung erreicht werden, indem ein ursprünglicher Ausdruck mehrmals symbolisch mittels eines Operators etwa der Form `simplify` ∘ `expand` äquivalent umgeformt wird.

29 Bemerkenswert in diesem Zusammenhang ist, daß inzwischen selbst die Väter von Maple die Entwicklung von „hybrid symbolic-numeric algorithms for scientific computation" als erklärtes Ziel angeben

6.3 Probleme mit Maple

Die umfassenden Fähigkeiten des Computeralgebrasystems Maple im Bereich der symbolischen Behandlung von Problemen aus Algebra und Analysis sind unbestreitbar. Ebenso dürfte – mindestens aufgrund der Ansprüche sowohl der Schöpfer als auch der Gemeinde der Nutzer von Maple – die Korrektheit der hier benutzten Algorithmen weitgehend sichergestellt sein (obwohl auch hier Vorsicht geboten ist, vgl. [Wal96]). Problematischer dagegen sind die „accessories", die als weitere Dienstleistungen angeboten werden.

6.3.1 Die Maple-interne Programmiersprache

Sind gelegentliche Bugs in einem Programmpaket der Größe Maples noch zu verschmerzen und ist die Form von graphischen Displays und die Bildschirmausgabe von Ergebnissen oft auch eine Frage des persönlichen Geschmacks, so daß in diesem Bereich stets Kompromisse eingegangen werden müssen, so ist leider festzustellen, daß die Autoren von Maple die eingebaute Programmiersprache eher stiefmütterlich behandelt haben. Daher traten bei der Implementierung der verschiedenen Pakete von `diskus` stets Probleme auf, und es mußten, wie bereits in der Einleitung zu Abschnitt 4 erwähnt, mehr oder weniger umständliche Hilfskonstruktionen programmiert werden. Hier sei insbesondere nochmals auf zwei der gravierendsten Mankos hingewiesen: Zum einen ist die Maple-interne Programmiersprache nicht streng getypt, zum anderen war die unmittelbare Weiterverarbeitung von Ausgaben, welche durch die symbolischen Algorithmen ermittelt wurden, oft nicht ohne ein geeignetes Post-Processing möglich, da jene offensichtlich abhängig von internen Zuständen derjenigen Maschine sind, auf welcher Maple gerade betrieben wurde (so z. B. die Reihenfolge ermittelter Nullstellen u. a.).

Vielleicht am bezeichnendsten für die Schwierigkeiten, die einem bei der Benutzung der Maple-internen Programmiersprache begegnen, sind die unerwarteten Probleme, die sich für uns bei der Umstellung von Release 3 auf Release 4 ergaben: die unter Rel. 3 bereits implementierten Teile von `diskus` waren unter Rel. 4 nicht mehr lauffähig, weil sowohl Syntax als auch Semantik nicht abwärtskompatibel sind (vgl. auch [Wal96]).

6.3.2 Unerwünschte „Vereinfachungen"

Neben der internen Programmiersprache hat sich im Laufe unserer Untersuchungen ein weiterer Schwachpunkt von Maple gezeigt, diesmal allerdings, was erheblich unangenehmer ist, im symbolischen Teil. Wie bereits erwähnt, kann aus theoretischen Gründen kein Programm das Äquivalenzproblem für elementare Funktionen lösen. Daher sind bei beliebigen Äquivalenzumformungen, wie durch die Routinen `simplify` oder `expand` realisiert, stets Einschränkungen und Mehrdeutigkeiten zu erwarten. Unangenehm und für uns zunächst unerwartet war jedoch die Tatsache, daß Maple in einigen, wie sich erweist, nur scheinbar offensichtlichen Fällen Ausdrücke selbständig, d. h. ohne eine explizite externe Aufforderung, „vereinfachte", was im Fall elementarer Funktionen bedeutete, daß nicht mehr die vom Benutzer vorgegebene Funktion diskutiert wurde, sondern eine zu dieser funktional äquivalente. Das wiederum kann zum Verlust markanter Punkte führen. So behandelt Maple die Funktion

$$x \mapsto \frac{x}{x}$$

wie die Funktion

$$x \mapsto 1,$$

was natürlich den Verlust der Definitionslücke $x = 0$ mit sich zieht. Analoges gilt für Funktionen wie

$$x \mapsto \sqrt{x^2}.$$

Es ist offensichtlich, daß zwischen den ursprünglichen Funktionen und den „vereinfachten" Funktionen gravierende Unterschiede bestehen können.

Leider wird auch diese Form von „Vereinfachungen" nicht konsequent durchgeführt, sondern ist abhängig von der Form der Eingabe. So wird z. B. die Funktion

$$x \mapsto \frac{x^2 - 1}{x - 1}$$

korrekt von `diskus` diskutiert (was bedeutet, daß der Term $\frac{x-1}{x-1}$ nicht herausgekürzt und damit $x_0 = 1$ als Definitionslücke erkannt wird). Wird dagegen die Eingabe in der Form

$$x \mapsto \frac{(x + 1)(x - 1)}{x - 1}$$

gewählt, so kürzt Maple zunächst wieder, so daß nur noch die Funktion $x \mapsto x + 1$ diskutiert werden kann.

Dieses Verhalten hat, wie wir bei unserer Arbeit sehen konnten, mindestens zwei Folgen: eine offensichtliche, die wir gerade geschildert haben, daß also stets damit zu rechnen ist, daß u. a. hebbare Singularitäten einer Funktion nicht erkannt werden, weil sie bereits „weggekürzt" worden sind, und eine versteckte, daß nämlich Nebeneffekte bei der Anwendung numerischer Prozeduren auftreten, was daran liegt, daß die Genauigkeit numerischer Auswertungen bei funktional äquivalenten Funktionen variieren kann[30].

30 Man betrachte hierzu nur die erwähnte Funktion $x \mapsto \sqrt{x^2}$ und die hierzu funktional äquivalente Funktion $x \mapsto x$.

6.4 Schlußfolgerungen

Das zunächst vielleicht etwas anwendungsfern erscheinende Problem der algorithmischen symbolischen Funktionsdiskussion hat sich als überaus fruchtbar erwiesen, um Effekte, die in der Computeralgebra eine große Rolle spielen bzw. beim Umgang mit Computeralgebrasystemen auftreten „im Kleinen", d. h. in einer kontrollierten, durch die konkrete Aufgabenstellung genau begrenzten Umgebung, exemplarisch zu studieren.

Die Erkenntnisse, die wir so bei der Entwicklung des Prototyps `diskus` gewonnen haben, sind aber nicht auf diese Problemstellung beschränkt, sondern spiegeln verschiedenste typische Aspekte des in gewisser Hinsicht noch relativ jungen und wenig strukturierten Gebiets der Computeralgebra wieder: das schrittweise Vergrößern des Vorrats an symbolischen Lösungsverfahren durch Einschränkung auf geeignete (Funktions-)klassen, der manchmal geringe Informationsgehalt von exakten Ergebnissen aufgrund ihrer komplizierten Struktur, die Notwendigkeit, *ohne* Verzicht auf Exaktheit auch numerische Methoden berücksichtigen zu müssen etc.

Ebenso sind verallgemeinernde Schlüsse über die Art und Weise möglich, wie Computeralgebrasysteme anzuwenden sind: Es ist deutlich geworden, daß sich auch bei der Benutzung von symbolischen Algorithmen stets die Frage nach der Korrektheit stellt und daß sich gerade an vermeintlich offensichtlichen Stellen, wie z. B. der geschilderten „Äquivalenz"-Umformung „einfacher" Ausdrücke, schwerwiegende Fehler einschleichen können.

Die Probleme, die wir mit der Maple-internen Programmiersprache hatten, legen desweiteren den Schluß nahe, daß man selbst von scheinbaren Universal-Tools nicht alles erwarten sollte. Anstatt größere Anwendungen in der internen Programmiersprache eines Computeralgebrasystems zu realisieren, ist es wohl zweckmäßiger, zur Implementierung eine der klassischen höheren Programmiersprachen zu benutzen, um somit auch moderne Methoden des Software-Engineering berücksichtigen zu können. Die Anbindung an ein gegebenes Computeralgebrasystem sollte nur insoweit geschehen, daß ohne Schwierigkeiten gewisse Teilprobleme an dieses übergeben und Lösungen von diesem übernommen werden können. Aus diesem Grund sollte bei der Entwicklung solcher Systeme eher Wert auf die Bereitstellung geeigneter Schnittstellen gelegt werden als auf die ad-hoc-Entwicklung interner Programmiersprachen.

Man könnte meinen, daß es sich bei den von uns festgehaltenen Beobachtungen um Binsenweisheiten handelt. Nichtsdestotrotz hat die Erfahrung der letzten Jahre gezeigt, daß dem nicht so ist. Wir glauben daher, daß es verstärkt notwendig ist, potentielle Anwender, ebenso wie Entwickler von Computeralgebra(systemen) frühzeitig mit diesen Tatsachen zu konfrontieren. Es ist klar, daß sich hieraus dann auch Konsequenzen für die Art und Weise ergeben müssen, in welcher Computeralgebra – und damit letztlich auch Mathematik – in Universität und Schule zu unterrichten ist.

A Beispielausgaben

A.1 Die LaTeX-Ausgabe

Funktion:

$$f(x) = x\sqrt{x^2 - 9}$$

Definitionslücken:

$$\mathcal{M}_{def_1} = \{x \in \mathbb{R} \mid -3 < x < 3\}$$

Nullstellen:

$$\begin{aligned} x_{null_1} &= -\sqrt{9} \\ &\approx -3.0 \end{aligned}$$

$$\begin{aligned} x_{null_2} &= \sqrt{9} \\ &\approx 3.0 \end{aligned}$$

Maxima:

keine gefunden oder keine vorhanden

Minima:

keine gefunden oder keine vorhanden

Nicht klassifizierbare Extremwerte:

keine gefunden oder keine vorhanden

PtoN-Wendestellen:

keine gefunden oder keine vorhanden

NtoP-Wendestellen:

$$\begin{aligned} (x_{ntop_1}, y_{ntop_1}) &= \left(-\frac{3\sqrt{6}}{2}, -\frac{9\sqrt{3}}{2}\right) \\ &\approx (-3.674234615, -7.794228636) \end{aligned}$$

$$\begin{aligned} (x_{ntop_2}, y_{ntop_2}) &= \left(\frac{3\sqrt{6}}{2}, \frac{9\sqrt{3}}{2}\right) \\ &\approx (3.674234615, 7.794228636) \end{aligned}$$

Sattelpunkte:

keine gefunden oder keine vorhanden

Nicht klassifizierbare Wendestellen:

keine gefunden oder keine vorhanden

Achsensymmetrie:

$$x_{achssym_1} \quad = \quad 0$$

Punktsymmetrie:

keine gefunden oder keine vorhanden

senkrechte Asymptote:

keine gefunden oder keine vorhanden

waagerechte Asymptote:

keine gefunden oder keine vorhanden

schiefe Asymptote:

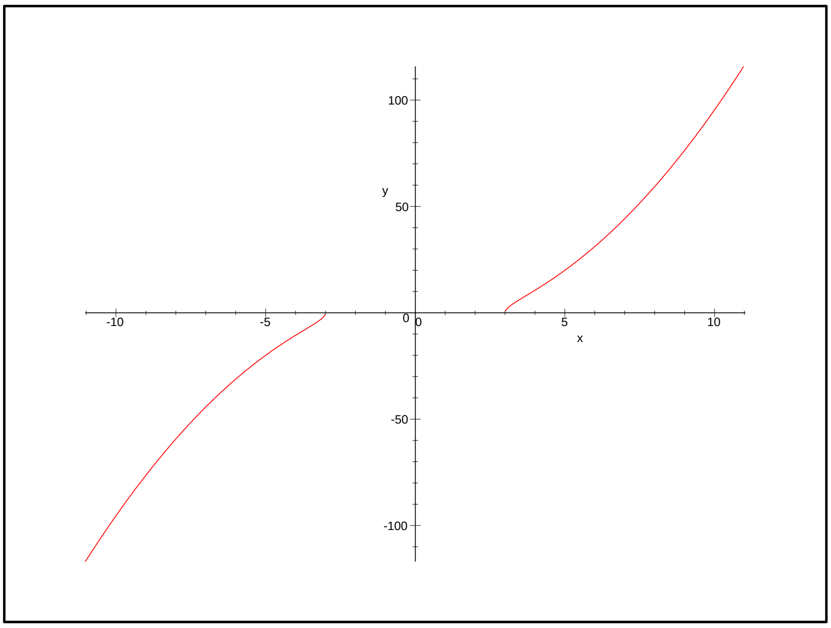
keine gefunden oder keine vorhanden

Verhalten im Unendlichen

$$\lim_{x \to -\infty} f(x) = -\infty$$
$$\lim_{x \to +\infty} f(x) = \infty$$

Wertetabelle für $f(x)$					
	+0	**+1**	**+2**	**+3**	**+4**
-12	-139.4274	-116.4131	-95.39392	-76.36753	-59.32959
-7	-44.27189	-31.17691	-20	-10.58301	0
-2	*complex*	*complex*	0	*complex*	*complex*
+3	0	10.58301	20	31.17691	44.27189
+8	59.32959	76.36753	95.39392	116.4131	139.4274

Plot der Funktion



A.2 Die Bildschirmausgabe im Prettyprint-Modus

```
> f:=(x->x)*((x->x^(1/2))@(x->x^2-9));
```

$$f := (x \to x)\left(x \to \sqrt{x}\right)@(x \to x^2 - 9)$$

```
> f(x);
```

$$x\sqrt{x^2 - 9}$$

```
> diskus(f,x);
```

$$\mathit{eingegebene\ Funktion:}$$

$$x\sqrt{x^2 - 9}$$

$$\#$$

$$\mathit{Definitionsluecken:}$$

$$-3 < x, x < 3$$

$$\#$$

$$\mathit{Nullstellen:}$$

$$-\sqrt{9}$$

$$\mathit{Numerisch:}$$

$$-3.000000000$$

$$- - - - - - - - - - - - - - - -$$

$$\sqrt{9}$$

$$\mathit{Numerisch:}$$

$$3.000000000$$

$$\#$$

$$\mathit{Extremwerte:}$$

$$\mathit{Maxima:\ keine\ vorhanden\ oder\ keine\ gefunden}$$

$$- - - - - - - - - - - - - - - - - - - -$$

$$\mathit{Minima:\ keine\ vorhanden\ oder\ keine\ gefunden}$$

$$- - - - - - - - - - - - - - - - - - - -$$

$$\mathit{Nicht\ klassifizierbare\ Extremwerte:\ keine\ vorhanden\ oder\ keine\ gefunden}$$

$$\#$$

$$Wendestellen:$$

$$PtoN: keine\ vorhanden\ oder\ keine\ gefunden$$

$$- -$$

$$NtoP:$$

$$-\frac{3}{2}\sqrt{6}, -\frac{9}{2}\sqrt{3}$$

$$Numerisch:$$

$$-3.674234615, -7.794228636$$

$$- - - - - - - - - - - - - - - - - - -$$

$$\frac{3}{2}\sqrt{6}, \frac{9}{2}\sqrt{3}$$

$$Numerisch:$$

$$3.674234615, 7.794228636$$

$$- -$$

$$Sattelpunkte: keine\ vorhanden\ oder\ keine\ gefunden$$

$$- -$$

$$Nicht\ klassifizierbare\ Wendestellen: keine\ vorhanden\ oder\ keine\ gefunden$$

$$\#$$

$$Symmetrieverhalten:$$

$$Achsensymmetrie:$$

$$0$$

$$- -$$

$$Punktsymmetrie: keine\ vorhanden\ oder\ keine\ gefunden$$

$$\#$$

$$senkrechte\ Asymptote: keine\ gefunden$$

$$- -$$

$$waagerechte\ Asymptote: keine\ gefunden$$

$$- -$$

$$schiefe\ Asymptote:\ keine\ gefunden$$

$$\#$$

$$Verhalten\ fuer\ x->\ -Unendlich:$$

$$-\infty$$

$$-\ -$$

$$Verhalten\ fuer\ x->\ +Unendlich:$$

$$\infty$$

$$\#$$

$$Wertetabelle:$$

```
|      || +0         | +1         | +2         | +3         | +4         |
|------||------------|------------|------------|------------|------------|
| -12  || -139.43    | -116.41    | -95.394    | -76.368    | -59.33     |
| -7   || -44.272    | -31.177    | -20        | -10.583    |  0         |
| -2   ||   complex  |   complex  |  0         |   complex  |   complex  |
|  3   ||  0         |  10.583    |  20        |  31.177    |  44.272    |
|  8   ||  59.33     |  76.368    |  95.394    |  116.41    |  139.43    |
-----------------------------------------------------------------------
```

A.3 Die Bildschirmausgabe im Textmodus

```
> f4:=(x->x)*((x->x^(1/2))@(x->x^2-9));
```

$$f := (x \to x)\left(x \to \sqrt{x}\right)@(x \to x^2 - 9)$$

```
> f(x);
```

$$x\sqrt{x^2-9}$$

```
> diskus(f,x,pretty=false);
```

$$\textit{eingegebene Funktion :}$$

$$x\sqrt{x^2-9}$$

```
########################################################################

Definitionsluecken:
   -3 < x < 3

########################################################################

Nullstellen:
   -3
    3

########################################################################

Maxima: keine vorhanden oder keine gefunden
------------------------------------------------------------------------
Minima: keine vorhanden oder keine gefunden
------------------------------------------------------------------------
Nicht klassifizierbare Extremwerte: keine vorhanden oder keine gefunden

########################################################################

PtoN: keine vorhanden oder keine gefunden
------------------------------------------------------------------------
NtoP:
   (-3.6742346, -7.7942286)
   ( 3.6742346,  7.7942286)
------------------------------------------------------------------------
Sattelpunkte: keine vorhanden oder keine gefunden
------------------------------------------------------------------------
Nicht klassifizierbare Wendestellen: keine vorhanden oder keine gefunden

########################################################################

Achsensymmetrie:
   0
```

```
-----------------------------------------------------------------------
Punktsymmetrie: keine vorhanden oder keine gefunden

#######################################################################

senkrechte Asymptote: keine vorhanden oder keine gefunden
-----------------------------------------------------------------------
waagerechte Asymptote: keine vorhanden oder keine gefunden
-----------------------------------------------------------------------
schiefe Asymptote: keine vorhanden oder keine gefunden

#######################################################################

Verhalten fuer x -> -Unendlich:
   -Unendlich
-----------------------------------------------------------------------
Verhalten fuer x -> +Unendlich:
   +Unendlich

#######################################################################

Wertetabelle:
|      || +0         | +1         | +2         | +3         | +4         |
|------||------------|------------|------------|------------|------------|
| -12  || -139.43    | -116.41    | -95.394    | -76.368    | -59.33     |
| -7   || -44.272    | -31.177    | -20        | -10.583    |  0         |
| -2   ||   complex  |   complex  |  0         |   complex  |   complex  |
|  3   ||  0         |  10.583    |  20        |  31.177    |  44.272    |
|  8   ||  59.33     |  76.368    |  95.394    |  116.41    |  139.43    |
-------------------------------------------------------------------------
```

Literatur

[Ban95] J. Banning. *Verfeinerung und Implementierung eines Algorithmus zur Diskussion elementarer Funktionen.* Diplomarbeit, Technische Universität Braunschweig, 1995.

[Cav70] B. F. Caviness. *On Canonical Forms and Simplification.* J. Assoc. Comput. Mach., **17** (1970), 385–396.

[Coh97] M. Cohrs. *Erweiterung eines Algorithmus zur Diskussion elementarer Funktionen.* Studienarbeit, Technische Universität Braunschweig, 1997.

[Ger92] E. H. A. Gerbracht. *Zur algorithmischen und exakten Diskussion elementarer Funktionen.* Diplomarbeit, Technische Universität Braunschweig, 1992.

[GS94] E. H. A. Gerbracht und W. Struckmann. *Zur algorithmischen Diskussion exakt invertierbarer elementarer Funktionen.* In DMV-Jahrestagung, Duisburg, 18.–24. September 1994, S. 216/206. Deutsche Mathematiker-Vereinigung (Herausgeber), 1994.

[GS96] E. H. A. Gerbracht und W. Struckmann. *Zur Diskussion elementarer Funktionen aus algorithmischer Sicht.* Informatik-Bericht Nr. 96-06, Technische Universität Braunschweig, Oktober 1996.

[GS97] E. H. A. Gerbracht und W. Struckmann. *Symmetrie elementarer Funktionen: Entscheidbarkeitsfragen und Algorithmen.* In GAMM-Jahrestagung, Regensburg, 24.–27. März 1997. Gesellschaft für angewandte Mathematik und Mechanik (Herausgeber), 1997.

[HTV84] W. Heitzinger, I. Troch und G. Valentin. *Praxis nichtlinearer Gleichungen.* Hanser Verlag, München/Wien, 1984.

[Klo95] A. Klos. *Entwurf und Implementierung von Algorithmen zur Bestimmung des Symmetrieverhaltens und der Asymptoten elementarer Funktionen.* Diplomarbeit, Technische Universität Braunschweig, 1995.

[Mig92] M. Mignotte. *Mathematics for Computer Algebra.* Springer-Verlag, Berlin, 1992.

[MW85] J. Marsden und A. Weinstein. *Calculus I.* Springer-Verlag, New York, 2. Auflage, 1985.

[SW95] R. Schmidt und C. Wagenknecht. *Abitur 1994: Moderne Mathematikwerkzeuge verboten!* Prax. Math. **2**, Schwerpunktthemenheft 12: Analysis, (1995), 84–89.

[Wal96] A. Walz. *MAPLE V – Eine Einführung.* http://www.physik.tu-muenchen.de /cip/local/maple_tut/mplintro.html, 1996.

[Wär97] S. Wärther. *Implementierung von Algorithmen zur Diskussion exakt-invertierbarer Funktionen.* Diplomarbeit, Technische Universität Braunschweig, 1997.

Corrigendum zu [GS96]

- Satz 4.3 zur Charakterisierung exakt invertierbarer Funktionen muß korrekt folgendermaßen lauten:

  Sei f eine exakt invertierbare Funktion. Dann existieren elementare Funktionen $\psi^{(1)}, \ldots, \psi^{(m)}$, so daß f funktional äquivalent zur Hintereinanderschaltung $\psi^{(1)} \circ \ldots \circ \psi^{(m)}$ ist. Für $i \in \{1, \ldots, m\}$ tritt dabei einer der nachstehenden Fälle ein:

  1. *$\psi^{(i)}$ ist elementar transzendent, d. h., $\psi^{(i)} \in \{\exp, \ln, \sin, \arctan\}$ mit den bekannten lokalen Umkehrfunktionen (das sind $\ln$, $\exp$, $\arcsin + 2k\pi$ mit $k \in \mathbb{Z}$ bzw. $\tan$),*
  2. *$\psi^{(i)}$ ist exakt invertierbares Polynom,*
  3. *$\psi^{(i)}$ ist exakt invertierbare gebrochen rationale Funktion,*
  4. *$\psi^{(i)}$ ist exakt invertierbare algebraische Funktion oder*
  5. *$\psi^{(i)}$ ist die lokale Umkehrfunktion einer der in den Fällen 2.–4. genannten Funktionen.*

  Die lokalen Umkehrfunktionen von f sind dabei von der Form

  $$f^{-1} = \left(\psi_{k_m}^{(m)}\right)^{-1} \circ \ldots \circ \left(\psi_{k_1}^{(1)}\right)^{-1},$$

  wobei die Indizes k_j geeignete ganzzahlige Parameter sind und $\left(\psi_{k_j}^{(j)}\right)^{-1}$ eine lokale Umkehrfunktion von $\psi^{(j)}$ bezeichnet.

- Eine Klassifikation aller exakt invertierbaren algebraischen Funktionen ist bisher noch nicht erfolgt, so daß auch Folgerung 4.9 entsprechend zu modifizieren ist.

Technische Universität Braunschweig
Informatik-Berichte ab Nr. 95-02

95-02	G. Snelting	Reengineering of Configurations Based on Mathematical Concept Analysis
95-03	A. Zeller	A Unified Configuration Management Model
95-04	H. Bickel, W. Struckmann	The Hoare Logic of Data Types
95-05	F.-J. Grosch	No Type Stamps and No Structure Stamps – a Referentially-Transparent Higher-Order Module Language
95-06	V. S. Cherniavsky	Über semantische und formalistische Beweismethoden in den exakten Wissenschaften
95-07	A. Zeller, D. Lütkehaus	DDD - A Free Graphical Front-End for UNIX Debuggers
95-08	A. Zeller	Smooth Operations with Square Operators – The Version Set Model in ICE
95-09	P. Funk, A. Lewien, G. Snelting	Algorithms for Concept Lattice Decomposition and their Application
96-01	A. Zeller, G. Snelting	Unified Versioning Through Feature Logic
96-02	M. Goldapp, U. Grottker, G. Snelting	Validierung softwaregesteuerter Meßsysteme durch Program Slicing und Constraint Solving
96-03	C. Lindig, G. Snelting	Modularization of Legacy Code Based on Mathematical Concept Analysis
96-04	J. Adámek, J. Koslowski, V. Pollara, W. Struckmann	Workshop Domains II (Proceedings)
96-05	F.-J. Grosch	A Syntactic Approach to Structure Generativity
96-06	E. H. A. Gerbracht, W. Struckmann	Zur Diskussion elementarer Funktionen aus algorithmischer Sicht
96-07	H.-D. Ehrich	Object Specification
97-01	A. Zeller	Versioning Software Systems through Concept Descriptions
97-02	K. Neumann, R. Müller	Implementierung von Assertions durch Oracle7-Trigger
97-03	G. Denker, P. Hartel	TROLL – An Object Oriented Formal Method for Distributed Information System Design: Syntax and Pragmatics
97-04	F.-J. Grosch	M - eine typisierte, funktionale Sprache für das Programmieren-im-Grossen
97-05	J. Küster Filipe	Putting Synchronous and Asynchronous Object Modules together: an Event-Based Model for Concurrent Composition
97-06	J. Küster Filipe	A categorical Hiding Mechanism for Concurrent Object Systems
97-07	G. Snelting, U. Grottker, M. Goldapp	VALSOFT Abschlussbericht
98-01	J. Krinke, G. Snelting	Validation of Measurement Software as an application of Slicing and Constraint Solving
98-02	S. Petri, M. Bolz, H. Langendörfer	Transparent Migration and Rollback for Unmodified Applications in Workstation Clusters
98-03	M. Cohrs, E. H. A. Gerbracht, W. Struckmann	DISKUS - Ein Programm zur symbolischen Diskussion reeller elementarer Funktionen