

Graph2D

Kai Nickel

Copyright © CopyrightÂ©1994-95 Kai Nickel

COLLABORATORS

	<i>TITLE :</i> Graph2D		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Kai Nickel	July 22, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Graph2D	1
1.1	Inhaltsverzeichnis Graph2D	1
1.2	Kurzbeschreibung	2
1.3	Autor	3
1.4	Copyright	3
1.5	Registrierung von Graph2D	4
1.6	Installation	5
1.7	Fragen & Antworten	6
1.8	Bedienung	7
1.9	Projekt-Menü	7
1.10	Funktionseditor	8
1.11	Notation	9
1.12	Diskussion	10
1.13	num. Integration	11
1.14	Wertetabelle	12
1.15	Tangente anlegen	12
1.16	Taylorreihe	12
1.17	2D-Graph-Fenster	13
1.18	2D-Graph-Einsteller	14
1.19	2D-Graph Sonstiges-Einsteller	16
1.20	SIRDS-Graph-Fenster	16
1.21	SIRDS-Graph-Einsteller	17
1.22	Textanzeigefenster	18
1.23	Voreinstellungseeditor	18
1.24	Drucken	19
1.25	Funktion	19
1.26	Interne Funktionen und Operatoren	20
1.27	Einbindung	21
1.28	EBNF	22
1.29	Entwicklungsgeschichte	22
1.30	Benutzeroberfläche Mui	27
1.31	SIRDS	28
1.32	Index	29

Chapter 1

Graph2D

1.1 Inhaltsverzeichnis Graph2D

G r a p h 2 D

Funktionsplotter mit Analysefunktionen

Version 2.10 (03.03.1995)

Autor: Kai Nickel

1. Allgemeines

1.1	~Kurzbeschreibung~~~	-	Was ist Graph2D?
1.2	~Autor~~~~~	-	Wer und wo ist der Autor?
1.3	~Copyright~~~~~	-	Rechtliche Anmerkungen
1.4	~Registrierung~~~~~	-	Um Graph2D dauerhaft zu benutzen
1.5	~Installation~~~~~	-	So kommt Graph2D auf die Platte
1.6	~Fragen&~Antworten~	-	Beantwortung häufiger Fragen

2. Bedienung

2.1	~Allgemeines~~~~~	-	Allgemeines zur Bedienung
2.1.1	~~~Projekt-Menü~~~~~	-	Das Projekt-Menü
2.2	~Funktionseditor~~~~~	-	Der Funktionseditor
2.2.1	~~~math,~Notation~~~	-	Die Notations-Fenster
2.2.2	~~~Kurvendiskussion~	-	Das Kurvendiskussionsfenster
2.2.3	~~~num,~Integral~~~~~	-	Berechnung des Simpsonintegrals
2.2.4	~~~Wertetabelle~~~~~	-	Erstellen einer Wertetabelle
2.2.5	~~~Tangente~~~~~	-	Anlegen einer Tangente
2.2.6	~~~Taylorreihe~~~~~	-	Approximationen mit dem Taylorpolynom
2.3	~2D-Graph~~~~~	-	Die 2D-Graph-Fenster
2.3.1	~~~2D-Einst.~Achsen	-	2D-Graph Achseneinstellungen
2.3.2	2D-Einst. Sonst.	-	Sonstige 2D-Graph Einstellungen
2.4	~SIRDS-Graph~~~~~	-	Die SIRDS-Graph-Fenster
2.4.1	~~~SIRDS-Einsteller~	-	Der SIRDS-Grapheinsteller
2.5	~Textanzeiger~~~~~	-	Die Textanzeigefenster
2.6	~Voreinstellungen~~~	-	Der Voreinstellungseditor
2.7	~Drucken~~~~~	-	Der Druck-Requester

3. Anhang

3.1	~Funktionen~~~~~	-	Was ist eine Funktion?
-----	------------------	---	------------------------

3.1.1	~~~Interne~Fkts&Ops~	-	Eingebaute Funktionen und Operatoren
3.1.2	~~~Einbindung~~~~~	-	Funktionen gegenseitig einbinden?
3.1.3	~~~EBNF~~~~~	-	Funktionssyntax in Backus-Naur-Form
3.2	~Geschichte~~~~~	-	Änderungen zu älteren Versionen
3.3	~MUI~~~~~	-	Die Benutzeroberfläche
3.4	~SIRDS~~~~~	-	Die verwendete 3D-Technik
3.5	~Index~~~~~	-	Index dieser Dokumentation

1.2 Kurzbeschreibung

1.1 Kurzbeschreibung

Graph2D ist ein Funktionsplotter, der in mehreren Arbeitsfenstern Graphen von reellen Funktionen darstellen und Kurvendiskussionen durchführen kann. Dazu gehören die Bestimmung von Nullstellen, Extrema, Wendepunkten, Monotoniebereichen und der Symmetrie. Es ist möglich, Funktionen abzuleiten, zu vereinfachen, Wertetabellen auszugeben, numerische Integrale zu berechnen, Tangenten anzulegen und Taylor-Approximationen zu berechnen. Außerdem können Funktionen als SIRDS dargestellt werden, wodurch bei richtiger Betrachtungstechnik der Eindruck "echter" Dreidimensionalität entsteht. Graph2D hat eine systemkonforme MUI-Oberfläche, enthält ein Installationsskript, eine Online-Hilfe und liegt lokalisiert in Deutsch und Englisch vor

Die Features in Stichpunkten:

- Ausgabe von Funktionen in stark konfigurierbaren Koordinatensystemen
- Erzeugung von SIRDS für dreidimensionale Funktionen
- Arbeiten mit vielen Funktionen in mehreren Fenstern gleichzeitig
- Erstellung von Kurvendiskussionen (Nullstellen, Extrema, Wendepunkte, Monotonie, Symmetrie)
- symbolisches Ableiten und numerisches Integrieren von Funktionen
- Taylor-Reihen Entwicklung
- Erzeugung von Wertetabellen und Anlegen von Tangenten
- gegenseitige Einbindung von Funktionen problemlos möglich
- Darstellung von Funktionen auch in "mathematischer Notation"
- Ausdruck von Graphen, Funktionen und Diskussionen in wählbarer Größe und Auflösung
- Graphen und Funktionen als IFF-Graphiken beliebiger Größe speicherbar
- systemkonforme und komfortable MUI-Oberfläche
- Online-Hilfe im AmigaGuide-Format
- Installation mit Commodore-Installer
- Lokalisierung Deutsch und Englisch
- Shareware mit vollem Funktionsumfang

Systemanforderungen:

- Amiga mit Kickstart2.0 oder höher
- installiertes MUI-System in der Version 2.1 oder höher

1.3 Autor

1.2 Autor

Graph2D und diese Anleitung wurden geschrieben von

Kai Nickel

Post : Herzogstraße 29
D-67435 Neustadt

eMail : kai@rpsbbs.pfalz.de (Kai Nickel)
un7x@rz.uni-karlsruhe.de

Bankverbindung: Kontonummer 247 014
BLZ 546 600 24 (Raiffeisenbank Neustadt/Wstr. Nord)

Über Fehlerberichte oder Verbesserungsvorschläge wäre ich sehr dankbar, ebenso über Menschen, die Graph2D oder die Anleitung in eine andere Sprache übersetzen möchten. Bitte setzt Euch aber vorher mit mir in Verbindung!

Bezugsquellen:

Updates von Graph2D werden von mir ins FRAS und ins Aminet geposted und wenn möglich auch auf der SaarAG - FD-Serie veröffentlicht. Außerdem gibt es sie unter anderem in folgenden Mailboxen. Dort sollte auch eine einigermaßen aktuelle Version von MUI zu finden sein.

- RPSBBS (06329-1624)
Login "GAST", Brett "/Lokal/Support/Amiga/Graph2D"
(ich selbst heiße dort "Kai")
- SchneeII (06347-92071)
Login "GAST", Brett "Bin-Filebase/Programmier-Brett"
(ich selbst heiße dort "Kai")

1.4 Copyright

1.3 Copyright

Graph2D ist Copyright © 1994-95 Kai Nickel.

Graph2D ist Shareware. Der Autor behält sich das Copyright für das Programm vor. Wer Graph2D einigermaßen ernsthaft bzw. über längere Zeit hinweg benutzen möchte, muß sich registrieren lassen.

Die Weitergabe der unregistrierten Version bleibt erlaubt, soweit damit keine kommerziellen Interessen verbunden sind. Konkret darf der Preis einer Diskette, die Graph2D enthält, 5.- DM nicht übersteigen. Eine Aufnahme in PD-Serien ist hiermit unter diesen Bedingungen explizit erlaubt, ebenso die

Verbreitung mittels DFÜ. Bei jeder Art der Weitergabe müssen alle zum Originalarchiv von Graph2D gehörenden Dateien unverändert und gemeinsam übernommen werden.

Der Autor gewährt keine Garantie für die Richtigkeit und Funktionsfähigkeit von Graph2D und übernimmt keinerlei Haftung für irgendwelche negativen Folgen, die durch Graph2D entstehen könnten. Spätere Updates und Fehlerbeseitigungen werden nicht garantiert.

Graph2D wurde mit dem Amiga-Oberon-Compiler V3.20 der A+L AG erstellt. Die Lizenzbedingungen des Compilers verbieten eine Nutzung des Programmes im militärischen Bereich.

Graph2D benutzt für seine Oberfläche das MUI-System von Stefan Stuntz.

MagicWB und somit auch einige der mitgelieferten Icons von Graph2D stehen unter Copyright von Martin Huttenloher.

Der Algorithmus zur 3D-Darstellung von Bildern stammt von Kilian Singer und wurde mit seiner Genehmigung in Graph2D angewandt.

1.5 Registrierung von Graph2D

1.4 Registrierung

Graph2D ist Shareware. Wer das Programm einigermaßen ernsthaft nutzt, ist aufgefordert, sich beim Autor registrieren zu lassen.

Die Registrierungsgebühr beträgt 20,- DM

Wer Graph2D über eine gewisse Testphase hinaus benutzt, ohne sich registrieren zu lassen, verstößt gegen die Nutzungsbedingungen. Die unregistrierte Version ist zwar uneingeschränkt funktionsfähig, jedoch erscheinen von Zeit zu Zeit "Nerv-Requester", die an die Registrierung erinnern sollen.

Durch die Registrierung bekommt man ein persönliches "Keyfile", daß einen zur dauerhaften Benutzung von Graph2D und eventueller zukünftiger Updates berechtigt. Ich kann aber natürlich nicht zusichern, ob und wie viele Updates es noch geben wird - in der Vergangenheit waren es aber schon etliche...

Nach der Installation des Keyfiles verschwinden natürlich die Info-Fenster und Nerv-Requester!

Bedenkt bitte, daß in Graph2D viele Monate Arbeit stecken und daß gerade auch Rückmeldungen der User in Form von Registrierungen über mögliche Weiterentwicklungen und Fehlerbeseitigungen von Graph2D entscheiden.

Und so geht die Registrierung:

Ihr überweist die 20,- auf mein Konto und schreibt mir dann eine e-Mail oder einen Brief, daß Ihr Euch für Graph2D registrieren lassen möchtet. Dazu benutzt Ihr am besten das mitgelieferte Formular Graph2D.RegForm. Wer

einen Brief abschickt, kann diesem natürlich auch das Geld in bar oder Euroscheck beilegen. Wie auch immer – nachdem das Geld bei mir angekommen ist, werde ich Euch das Keyfile per E-Mail (falls angegeben) senden. Wer keine e-Mail Adresse hat, dem werde ich das Keyfile auf einer Disk mit der Post zuschicken.

Probleme?

Da bekanntermaßen durchaus einmal eine e-Mail verloren gehen kann, schreibt mir ruhig noch einmal, wenn meine Antwort länger als z.B. zwei Wochen aussteht. Ich werde dann natürlich noch einmal das Keyfile absenden. Sollte es so gar nicht funktionieren, dann schickt mir einfach einen Brief mit der Post und ich werde dann genauso antworten. Um Pannen zu vermeiden ist es ganz wichtig, daß Ihr im "Verwendungszweck" des Überweisungsdruckes Euere e-Mail und am besten auch reale Adresse angebt – dann kann eigentlich nichts mehr passieren!

Nach der Registrierung:

Das empfangene Keyfile sollte am besten in das Verzeichnis kopiert werden, in dem sich das Hauptprogramm "Graph2D" befindet. Falls es dort aus unerfindlichen Gründen nicht erkannt werden sollte – oder gar bei jedem Start eine Meldung "Please insert Volume PROGDIR:" erscheint – dann kopiert man das Keyfile in das 'S:'-Verzeichnis des Systems. Dort wird es sicher erkannt.

Vielen Dank für die Unterstützung des Shareware-Gedankens!

1.6 Installation

1.5 Installation

Automatisch:

Man installiert Graph2D am besten, indem man das mitgelieferten Skript "Graph2D-Install" anklickt. Alles weitere läuft dann im Dialog mehr oder weniger von selbst ab. Graph2D wird danach einfach durch Anklicken von der Workbench aus gestartet.

Für die Ängstlichen: In der Einstellung "Expert User" darf man wirklich jede Aktion selbst bestätigen!

Manuell:

Wer den Installer von Commodore nicht hat, kann Graph2D auch folgendermaßen von Hand installieren (genau das macht auch das Skript):

- Kopieren von Hauptprogramm, Guide und Registrierungsformular in ein selbstgewähltes Verzeichnis.
 - Kopieren der "GarbageCollector.library" in das "LIBS:" Verzeichnis des Systems oder nach PROGDIR:Libs/. Diese Library wird zum Ablauf unbedingt benötigt!
-

- optional: Unterverzeichnis "Functions" erzeugen und Beispielfunktionslisten hineinkopieren
- optional: Unterverzeichnis "Graphs" erzeugen und Beispiel-Grapheinstellungen hineinkopieren
- optional: Kopieren des engl. Katalogs nach Locale:Catalogs/english/ oder PROGDIR:Catalogs/english/
- optional: ersetzen der Icons durch die aus dem Verzeichnis "MWBIcons"

Systemvoraussetzungen:

Um Graph2D benutzen zu können benötigt man einen Amiga mit Kickstart 2.0 (das ist V36) oder höher. Außerdem muß das Mui-System von Stefan Stuntz in der Version 2.1 oder höher installiert sein, ohne das Graph2D nicht laufen wird.

Originalarchiv:

Zum original Graph2D-Paket gehören folgende Dateien:

Graph2D	- Das Hauptprogramm
Graph2D-Install	- Das Installationsskript
Docs/Deutsch/Graph2D.guide	- Dokumentation/Onlinehilfe
Docs/Deutsch/Graph2D.RegForm	- Registrierungsformular
Docs/Deutsch/Graph2D.LiesMich	- Kurzbeschreibung
Docs/English/Graph2D.guide	- engl. Dokumentation/Onlinehilfe
Docs/English/Graph2D.RegForm	- engl. Registrierungsformular
Docs/English/Graph2D.ReadMe	- engl. Kurzbeschreibung
Docs/English/Product-Info	- Beschreibung im "Fish-Format"
Catalogs/english/Graph2D.catalog	- engl. Katalog
Libs/garbagecollector.library	- benötigte Library
Libs/GarbageCollector.LiesMich	- Info zur Library
Graphs/Standard.2D	- 2D-Standardkoordinatensystem
Graphs/Standard.SIRDS	- SIRDS-Standardkoordinatensystem
Graphs/Trigonometrie.2D	- 2D-Systembeispiel für Trigonometrie
Functions/Bsp2D.fkt	- Beispielfunktionen
Functions/Bsp3D.fkt	- Beispielfunktionen für 3D-Ansicht
MWBIcons/ClickForColors	- MagicWB-Palette (zum Ausprobieren!)
MWBIcons/Graph2D.info	- Icon im MagicWB-Stil
MWBIcons/Graph2D.guide.info	- Icon im MagicWB-Stil
MWBIcons/Graph2D.RegFrom.info	- Icon im MagicWB-Stil
MWBIcons/Drawer.info	- Icon im MagicWB-Stil
MWBIcons/Graphs.info	- Icon im MagicWB-Stil
MWBIcons/Functions.info	- Icon im MagicWB-Stil
MWBIcons/Catalogs.info	- Icon im MagicWB-Stil
MWBIcons/Libs.info	- Icon im MagicWB-Stil

1.7 Fragen & Antworten

1.6 Fragen und Antworten

Wie stelle ich jetzt eigentlich eine Funktion als Graph dar?

Zuerst erzeugt man eine neue Funktion mittels "Erzeugen", gibt ihren Text ein (z.B. " $f(x)=\sin(x)^2$ ") und öffnet einen neuen Graphen mittels "Neuer Graph". Um die neue Funktion im neuen Graphen erscheinen zu lassen, markiert man sie als "Gezeichnet" und bringt den Graphen mittels "Aktualisieren" dazu, sich neu aufzubauen. Die Funktion sollte nun zu erkennen sein, falls ihre Werte in den im Graphen sichtbaren Bereich fallen. Gegebenenfalls muß man den Graph noch "Einstellen" um ein angemessenes Darstellungsintervall zu wählen.

1.8 Bedienung

2.1 Bedienung

Graph2D bietet diese Datei hier auch als kontextsensitive Onlinehilfe an. Durch Drücken der Help-Taste auf der Tastatur kann sie jederzeit geöffnet werden. Dabei wird automatisch in das zum aktiven Fenster passende Kapitel gesprungen.

Nach dem Start von Graph2D öffnet sich zunächst das Hauptfenster, der sog. Funktionseditor, in dem die Funktionen, mit denen gearbeitet werden soll, eingegeben und bearbeitet werden können.

Sind Funktionen vorhanden, können sie in Graph-Fenstern dargestellt werden, oder mathematisch analysiert oder verändert werden.

1.9 Projekt-Menü

2.1.1 Das Projekt-Menü

Das Projekt-Menü steht im Funktionseditor, den Graph-Fenstern und dem Notationsfenster zur Verfügung. Es bietet folgende Funktionen:

Projekt

- | | |
|---------------------|--|
| Neuer Graph... | - Öffnet ein neues Graph-Fenster. Der Typ (2D-Graph oder SIRDS-Graph) muß nun noch in einem kleinen Requester ausgewählt werden. |
| Voreinstellungen... | - Öffnet den Voreinstellungsektor |
| Hilfe... | - Zeigt die Online-Hilfe an, identisch mit der Funktion der HELP-Taste |
| Über... | - Öffnet ein Info-Fenster |
| Ende | - Beendet Graph2D nach Sicherheitsabfrage |

1.10 Funktionseditor

2.2 Funktionseditor

Der Funktionseditor ist ein Fenster, in dem alle im Speicher befindlichen Funktionen aufgelistet und bearbeitet werden können.

Eine Funktion aus dem List-Gadget erscheint durch Anklicken im Funktionen-Gadget und kann nun editiert werden. Wenn die Eingabe bzw. Änderung der Funktion durch <RETURN> beendet ist, wird sie automatisch interpretiert. (Der Auswahlbalken kann übrigens auch aus dem String-Gadget heraus mit den Cursortasten bewegt werden.)

Tritt beim Interpretieren ein Fehler auf, wird dieser darunter im Textfeld "Fehler" angezeigt, und die Funktion wird in der Funktionsliste kursiv dargestellt. Eine solche fehlerhafte Funktion kann natürlich nicht gezeichnet oder analysiert werden!

Die HELP-Taste über dem String-Gadget oder dem Fehler-Feld gedrückt bewirkt, daß direkt das "Funktionen"-Kapitel der Anleitung angezeigt wird, mit dessen Hilfe man schnell eventuelle Fehler in der Syntax erkennen kann.

Im String-Gadget "Graph" mit beigeordneter Popup-Liste werden alle existierenden Graphen mit ihrem Namen angezeigt. Jedem Graphen kann ein individueller Name gegeben werden, unter dem er in der Liste erscheinen wird. Ein Doppelklick auf einen Graphen in der Liste bringt diesen sofort in den Vordergrund.

Von jeder augenblicklich ausgewählten Funktion kann eingestellt werden, ob und mit welcher Farbe sie im ausgewählten Graphen eingezeichnet werden soll. Ein Doppelklick auf eine Funktion schaltet ihren Zustand zwischen gezeichnet und nicht-gezeichnet hin und her. Dies ist identisch mit einem Klick auf das "Gezeichnet"-Gadget. Im aktiven Graphen gezeichnete Funktionen werden in der Liste durch ein vorangestelltes ">" gekennzeichnet. Die Farbe einer Funktion wird im Paletten-Gadget "Farbe" ausgewählt, ebenso wie ein "Pattern" (die "Strichelung") gewählt werden kann.

An einer eingezeichneten Funktion gemachte Änderungen werden erst dann sichtbar, wenn im Graphen "aktualisieren" angeklickt wird! Auf diese Weise wird nicht nach jeder einzelnen Änderung der ganze Graph aufwendig neu berechnet, sondern erst, wenn dies gewünscht wird.

Es stehen folgende Bild- bzw. Textbuttons zur Verfügung, mit denen man mit der ausgewählten Funktion folgende Dinge tun kann:

math. Notation	- Öffnet das Notations-Fenster
Ableitung	- Erzeugt die (erste) Ableitung
Vereinfachen	- Eine äquivalente Vereinfachung wird erzeugt, d.h. die Funktion wird möglichst weit "ausgerechnet".
Diskussion	- Unternimmt eine konfigurierbare Kurvendiskussion
num. Integral	- Berechnet das bestimmte~Integral
Wertetabelle	- Erzeugt eine Wertetabelle
Tangente	- Erzeugt eine Tangente

Solange der Funktionseditor das aktive Fenster ist, stehen folgende Menüs zur Verfügung. Die oft benötigten Punkte "Notation", "Ableiten", "Diskussion" und "Neuer Graph" sind zusätzlich im Fenster auch als Gadgets vorhanden, um eine schnellere Anwahl zu ermöglichen:

Projekt - Das Projekt-Menü

Funktionen

Alle löschen	- Alle Funktionen werden gelöscht
Laden ...	- Eine Funktionsliste wird geladen
Laden dazu...	- Eine Funktionsliste wird geladen und an die im Speicher vorhandene angehängt
Speichern als...	- Die Funktionsliste im Speicher wird unter wählbarem Namen abgespeichert

Edit

Erzeugen	- Eine neue, "leere" Funktion wird an ausgewählter Stelle in die Funktionsliste eingesetzt und kann sofort editiert werden.
Ausschneiden	- Die ausgewählte Funktion wird aus der Liste ausgeschnitten.
Einfügen	- Eine zuvor ausgeschnittene Funktion wird an ausgewählter Stelle wieder in die Funktionsliste eingefügt.

Sonstiges

Vereinfachen	- wie der entsprechende Bildbutton (s.o.)
Ableitung	- s.o.
Diskussion	- s.o.
num. Integral...	- s.o.
Wertetabelle...	- s.o.
Tangente ...	- s.o.
math. Notation...	- s.o.

1.11 Notation

2.2.1 Notation

In einem Fenster wird die ausgewählte Funktion in einer etwas lesbareren, "normaleren" Form gezeigt. Divisionen werden hierbei mit Bruchstrichen dargestellt und Exponenten sind richtig hochgestellt. Die ganzen Sache heißt (vielleicht etwas unglücklich) "mathematische Notation"...

Ist die math. Notation einer Funktion größer als das Fenster, kann der sichtbare Ausschnitt mit den Verschiebepalken bewegt werden. Geschlossen wird das Fenster durch das Schließ-Gadget.

Von den Notations-Fenstern können mehrere gleichzeitig geöffnet sein. Auch während sie offen sind, kann anderweitig weitergearbeitet werden.

Folgende Menus können vom Notations-Editor angewählt werden:

Projekt	- Das Projekt-Menü
Notation	
Drucken	- Druckt die sichtbare Notation aus. Die Druckdichte kann durch die Workbench-Prefs eingestellt werden.
Speichern IFF...	- Speichert die Notation als IFF-Graphik ab. Sie kann so von allen Programmen verwendet werden, die das IFF-ILBM-Format unterstützen.

1.12 Diskussion

2.2.2 Kurvendiskussion

Graph2D ist in der Lage, Funktionen einer Kurvendiskussion zu unterziehen. Voraussetzung dafür ist eine ganzrationale Funktion mit einer Laufvariablen.

In diesem Fenster muß zunächst das sogenannte Untersuchungsintervall eingegeben werden, in dessen Grenzen die Diskussion durchgeführt werden soll. Das Untersuchungsintervall sollte stets so klein wie möglich gehalten werden - je größer es gewählt wird, desto schlechter findet Graph2D die gesuchten Stellen der Funktion. Das Ergebnis einer jeden Diskussion wird in Klartext in einem Textanzeigefenster ausgegeben und kann von dort aus auch ausgedruckt werden.

Eine Kurvendiskussion wird mit Start begonnen und gegebenenfalls mittels Abbruch vorzeitig beendet werden.

Die Diskussion einer Funktion kann (besonders auf 68000er Amigas) mitunter recht lange dauern, auch sind die Ergebnisse meist mit Vorsicht zu genießen. Es empfiehlt sich immer, sich zur Kontrolle den Graphen der betreffenden Funktion zeichnen zu lassen und die Ergebnisse der Diskussion mit seiner Hilfe zu verifizieren!!! Man wird schnell feststellen, daß bspw. bestimmte Nullstellen leider einfach nicht gefunden werden.

Eine komplette Diskussion besteht aus folgenden Teilen, die alle einzeln zuschaltbar sind:

Nullstellen: $f(x) = 0$

Die Nullstellen der ausgewählten Funktion im Untersuchungsintervall werden bestimmt. Zu jeder Nullstelle bekommt man auch den Typ der Nullstelle angezeigt. Mögliche Typen sind:

-+	(von unten nach oben)	: $f'(x) > 0$
+-	(von oben nach unten)	: $f'(x) < 0$
++	(Berührstelle von oben)	: $f'(x) = 0$
--	(Berührstelle von unten)	: $f'(x) = 0$

Auf die Nullstellenbestimmung beruhen auch alle anderen Diskussionsteile. Momentan arbeitet der verwendete Algorithmus leider jedoch nicht ganz zufriedenstellend, einige Nullstellen, besonders bei einer größeren Anzahl im Untersuchungsintervall, werden nicht erkannt – das fälschliche Erkennen einer Nullstelle allerdings kommt so gut wie nicht vor.

Extrema: $f'(x) = 0$

Hierbei werden die lokalen Minima (Tiefpunkte) und Maxima (Hochpunkte) der Funktion im Untersuchungsintervall bestimmt.

min (Minimum): $f''(x) > 0$
max (Maximum): $f''(x) < 0$

Wendepunkte: $f''(x) = 0$

Die Wendepunkte der ausgewählten Funktion werden ausgegeben, dazu erhält man noch die eine der folgende zusätzlichen Information:

links->rechts: $f'''(x) < 0$
rechts->links: $f'''(x) > 0$

Monotonie:

Das Untersuchungsintervall wird in Bereiche unterteilt, in denen die ausgewählte Funktion folgende Monotonieeigenschaften zeigt:

monoton steigend : $f'(x) \geq 0$
monoton fallend : $f'(x) \leq 0$
streng monoton steigend: $f'(x) > 0$
streng monoton fallend : $f'(x) < 0$

Symmetrie:

Hierbei wird die ausgewählte Funktion geprüft auf

Punktsymmetrie (zum Ursprung): $f(x) = -f(-x)$
Symmetrie zur y-Achse : $f(x) = f(-x)$

Trifft beides nicht zu, wird "keine Symmetrie" erkannt.

Die Berechnung der Symmetrie erfolgt durch Einsetzen einer größeren Anzahl von Testwerten.

1.13 num. Integration

2.2.3 Numerische Integration

Funktionen mit einer Laufvariablen können mittels der Simpsonformel numerisch integriert werden. Dazu muß mit "von" und "bis" ein Integrationsintervall angegeben und durch "Zerlegungen" die Genauigkeit der Integration eingestellt werden. Ist dies geschehen, wird in einem Textanzeigefenster ein Report erstellt, der das Resultat der Integration bekannt gibt. Es ist zu beachten, daß "Flächen" die unterhalb der x-Achse

liegen mit negativem Vorzeichen bedacht werden, so daß z.B. das Integral von "sin(x)" im Bereich [0, 2*Pi] genau 0 beträgt, da sich die positive und die negative "Fläche" gegenseitig aufheben. Ist man allerdings nur an der Maßzahl dieser Flächen interessiert (also am nichtorientierten Flächeninhalt), so könnte man das in diesem Beispiel durch die Betrachtung von "abs(sin(x))" erreichen.

Simpsonsche Regel:

a, b: Integrationsintervallgrenzen
k : Anzahl der Zerlegungen

$$\int_a^b f(x) dx = \frac{b-a}{k*3} * (f(x_0) + 4*f(x_1) + 2*f(x_2) + \dots + 2*f(x_{k-2}) + 4*f(x_{k-1}) + f(x_k))$$

1.14 Wertetabelle

2.2.4 Wertetabelle

Von Funktionen mit max. einer Laufvariablen können Wertetabellen errechnet werden. Die Grenzen der Tabelle werden mit "von" und "bis" eingestellt, in "Schrittweite"-breiten Abständen erfolgt die Berechnung der Werte. Die Ausgabe erfolgt in einem Textanzeigefenster.

1.15 Tangente anlegen

2.2.5 Tangente anlegen

An Funktionen mit max. einer Laufvariablen können Tangenten an beliebigen Punkten "x" angelegt werden. Dabei wird eine neue Funktion erzeugt, die die Gleichung der gewünschten Tangente besitzt. Diese Funktion besitzt den Anhang "tan" in ihrem Namen.

1.16 Taylorreihe

2.2.6 Taylorreihe

Funktionen mit max. einer Laufvariablen können durch ein Taylorpolynom in einer Umgebung um einen wählbaren Punkt x0 angenähert werden. Je höher der Grad gewählt wird, desto genauer wird die Approximation. Im Normalfall sollte man den Grad jedoch nicht unbedingt > 10 wählen, da sonst der Speicher- und Zeitbedarf sprunghaft gegen unendlich strebt... Die erzeugte Polynom-Funktion wird in die Funktionsliste aufgenommen und bekommt ein vorgestelltes "T" in ihrem Namen.

Taylorreihe: x_0 = gewünschter Antragspunkt

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(x_0) \cdot (x - x_0)^n}{n!}$$

Graph2D summiert natürlich nicht unendlich (∞) und bringt die Ausgabe in die gewohnte Polynom-Form. Deshalb wird folgende, angepasste Berechnung durchgeführt (n = gewünschter Grad):

$$\sum_{k=0}^n \frac{f^{(k)}(x_0) \cdot (x - x_0)^k}{k!}$$

1.17 2D-Graph-Fenster

2.3 2D-Graph-Fenster

2D-Graph-Fenster werden in beliebiger Anzahl vom Benutzer geöffnet. In ihnen können Funktionen (auch mehrere gleichzeitig) mit maximal einer Laufvariablen betrachtet werden.

Welche Funktionen in einem Graph-Fenster angezeigt werden, läßt sich im Funktionseditor einstellen. Die Graph-Fenster und der Funktionseditor laufen parallel, so daß zwischen ihnen hin- und hergeschaltet und "gleichzeitig" in ihnen gearbeitet werden kann. Graph-Fenster sind in der Größe veränderbar.

Die Bild- bzw. Textbuttonleiste bietet folgende Funktionen an, die auch durch Menüs und deren Shortcuts erreichbar sind:

Aktualisieren - der Graph wird mit allen eventuell in der Zwischenzeit erfolgten Änderungen neu gezeichnet. An einer eingezeichneten Funktion gemachte Änderungen werden also erst dann sichtbar, wenn der Graph aktualisiert wird!
Auf diese Weise wird nicht nach jeder einzelnen Änderung der ganze Graph aufwendig neu berechnet, sondern erst, wenn dies auch gewünscht wird. Auch das Vergrößern und Verkleinern eines Graphfensters geht relativ schnell vonstatten, da alle Funktionswerte gepuffert sind und erst durch "Aktualisieren" neu berechnet werden.

Einstellen - Öffnet den 2D-Graph Achsen-Einsteller, von dem aus

alle Einstellungen des Koordinatensystems festgelegt werden können.

- | | |
|------------------|---|
| Zoom in | - Mit der Maus kann nach Anwahl dieses Punktes im Graphen ein Bereich markiert werden. Nach Bestätigung öffnet sich dann ein neues Graph-Fenster, das genau diesen Bereich anzeigt. |
| Zoom out | - Ein neuer Graph öffnet sich, der den vierfachen Darstellungsbereich des Ursprungsgraphen umfaßt. |
| Drucken | - Druckt den Graphen aus |
| Speichern IFF... | - Speichert den Graphen als IFF-Bild, so daß er von allen Programmen, die das IFF-ILBM-Format unterstützen verwendet werden kann. Die Pixel-Größe in der der Graph, abgespeichert werden soll kann in den Voreinstellungen festgelegt werden. Der Graph wird dafür intern in der vollen Pixelgröße berechnet und aufgebaut. |

Das momentane Darstellungsintervall und der Name des Graphen werden in der Titelleiste des Graph-Fensters angezeigt. Solange man die linke Maustaste gedrückt hält, wird die Position des Mauszeigers in Graph-Koordinaten umgerechnet in der Textzeile oberhalb des Graphen angezeigt.

Solange ein 2D-Graph-Fenster aktiv ist, stehen folgende Menüs zur Verfügung:

- | | |
|---------|--------------------|
| Projekt | - Das Projekt-Menü |
|---------|--------------------|

Graph

- | | |
|------------------|--|
| Aktualisieren | - Wie entsprechender Bildbutton (s.o.) |
| Zoom in | - s.o. |
| Zoom out | - s.o. |
| Drucken... | - s.o. |
| Speichern IFF... | - s.o. |

Einstellen

- | | |
|--------------|--|
| Achsen... | - Öffnet den 2D-Graph Achsen-Einsteller, in dem die Achsen und deren Skalierung für den Graphen eingestellt werden können. |
| Sonstiges... | - Öffnet den 2D-Graph Sonstiges-Einsteller, in dem verschiedene weitere Einstellungen für den Graph vorgenommen werden können. |
| Seichern... | - Speichert die Einstellungen des Graphen zur späteren Verwendung unter wählbarem Namen ab. |
| Laden... | - Lädt eine Graph-Einstellungsdatei und wendet sie auf den Graphen an. |

1.18 2D-Graph-Einsteller

2.3.1 2D-Graph Achsen-Einsteller

In diesem Einsteller wird der Darstellungsbereich, die Achsenskalierung und einige weitere Merkmale eines Graphen verändert. Für die x- und die y-Achse können jeweils unabhängig voneinander folgende Angaben gemacht werden:

von ... bis - Das Darstellungsintervall für die betreffende Achse. Die gemachte Angabe ist abhängig von der gewählten Einheit!

Einheit - Die Basiseinheit der Achse, mit der alle anderen Angaben skaliert werden. Üblich ist eine Einheit von "1". Wenn man aber z.B. ein Koordinatensystem speziell für trigonometrische Funktionen erstellen will, so würde als Einheit sicher auch "3.141" sinnvoll sein. Ein Darstellungsintervall von z.B. [-2; 4] würde in diesem Fall bedeuten, daß das tatsächliche Intervall von $-2 * 3.141$ bis $4 * 3.141$ reichen würde, also effektiv x-Werte aus [-6.282;12.564] umfaßt. Die Einheit ist frei wählbar, solange sie >0 ist.

Achse zeichnen - Legt fest, ob die Achse überhaupt sichtbar sein soll. Auch wenn sie nicht gezeichnet wird, sind die Angaben über den Darstellungsbereich natürlich trotzdem nötig.

Titel - Der Anwender kann einer Achse einen Titel geben, der neben der Achse angezeigt wird. Wählt man z.B. für die x-Achse eine Einheit von "3.141" so ist es sicher wichtig, dies auch im Titel kenntlich zu machen, in dem man sie z.B. "x in pi" nennt.

Striche - Die Achse kann optional durch kleine Striche unterteilt werden. Hier kann man nun festlegen, nach wieviel Einheiten jeweils ein Strich gezeichnet werden soll. Außerdem können zwischen jeweils zwei dieser Striche eine beliebige Anzahl an Zwischenstrichen gezeichnet werden, die dann etwas kleiner als die eigentlichen Striche sind. Eine Zwischenstrichanzahl von 0 bedeutet keine Zwischenstriche. Folgen die Striche zu eng aufeinander, daß man sie nicht mehr unterscheiden könnte, so werden sie automatisch einfach nicht dargestellt.

Beschriftung - Die Striche können optional mit Zahlen beschriftet werden. Es kann festgelegt werden, nach wievielen Strichen wieder eine Zahl geschrieben werden soll. Die geschriebene Zahl ist auch wiederum abhängig von der gewählten Einheit. Falls die Beschriftung so eng gewählt wird, daß sich Zahlen überschneiden würden, so wird das automatisch verhindert, indem manche einfach ausgelassen werden.

Mit dem Button Sonstiges... gelangt man von hier direkt in den 2D-Graph Sonstiges-Einsteller, in dem noch weitere Einstellungen für den Graphen gemacht werden können.

Nach Verlassen des Achsen-Einstellers mit "Ok" wird der aktive Graph aktualisiert, um die evtl. gemachten Änderungen sichtbar zu machen. Beim Verlassen durch "Abbruch" oder Schließen des Fensters bleibt es natürlich bei den vorherigen Werten.

1.19 2D-Graph Sonstiges-Einsteller

2.3.2 2D-Graph Sonstiges-Einsteller

In diesem Fenster können verschiedene Einstellungen für den aktuellen Graphen gemacht werden:

Für beide Achsen kann angegeben werden, ob ein "Raster" in Form von Punkten oder Linien gezeichnet werden soll und welchen Abstand es in x- und y-Richtung (Wert wieder in Einheiten) haben soll. Der Typ legt fest, ob das Raster nur aus einzelnen Punkten, aus durchgezogenen Linien oder aus gestrichelten Linien bestehen soll.

Für die eingezeichneten Funktionen kann eine "Genauigkeit" ausgewählt werden, die festlegt, wie viele Funktionswerte für die Ausgabe des Graphen tatsächlich berechnet werden sollen. Bei geringerer Genauigkeit werden weniger Funktionswerte berechnet und die Ausgabe erfolgt in höherer Geschwindigkeit – jedoch wird der Funktionsgraph zuweilen etwas "eckiger" und unpräziser im Verlauf.

Mit "Punkte verbinden" werden die einzelnen berechneten Funktionswerte durch Linien miteinander verbunden. Wenn ein Funktionsgraph größere Sprünge macht (z.B. bei geringer Genauigkeit), werden so die entstehenden Lücken verhindert. Andererseits entstehen so eventuell Verbindungen, die in Wirklichkeit gar nicht existieren (Beispiel: $\tan(x)$)

"Schrift" erlaubt die Wahl des Zeichensatzes für den Graphen, in der die Einheitenzahlen und die Achsenbeschriftung dargestellt werden.

Nach Verlassen des Sonstiges-Einstellers mit "Ok" wird der aktive Graph aktualisiert, um die evtl. gemachten Änderungen sichtbar zu machen. Beim Verlassen durch "Abbruch" oder Schließen des Fensters bleibt es natürlich bei den vorherigen Werten.

1.20 SIRDS-Graph-Fenster

2.4 SIRDS-Graph-Fenster

SIRDS-Graph-Fenster werden in beliebiger Anzahl vom Benutzer geöffnet. In ihnen kann jeweils eine Funktionen mit maximal zwei Laufvariablen betrachtet werden. Zur Art der Darstellung siehe Kapitel SIRDS.

Welche Funktion in einem SIRDS-Graph-Fenster angezeigt wird, läßt sich im Funktionseditor einstellen. Die Graph-Fenster und der Funktionseditor laufen parallel, so daß zwischen ihnen hin- und hergeschaltet und "gleichzeitig" in ihnen gearbeitet werden kann. Graph-Fenster sind in der Größe veränderbar. Das momentane Darstellungsintervall und der Name des Graphen werden in der Titelleiste des Graph-Fensters angezeigt.

Die drei Bild- bzw. TextButtons haben folgende Bedeutung (v.l.n.r.):

Aktualisieren – der Graph wird mit allen eventuell in der Zwischenzeit

erfolgten Änderungen neu gezeichnet. An der eingezeichneten Funktion gemachte Änderungen werden also erst dann sichtbar, wenn der Graph aktualisiert wird! Auf diese Weise wird nicht nach jeder einzelnen Änderung der ganze Graph aufwendig neu berechnet, sondern erst, wenn dies auch gewünscht wird. Auch das Vergrößern und Verkleinern eines Graphfensters geht so relativ schnell vonstatten, da alle Funktionswerte gepuffert sind und erst durch "Aktualisieren" neu berechnet werden müssen.

- Einstellen - Öffnet den SIRDS-Graph-Einsteller
- Drucken - Druckt den Graphen aus
- Speichern IFF... - Speichert den Graphen als IFF-Bild, so daß er von allen Programmen, die das IFF-ILBM-Format unterstützen verwendet werden kann. Die Pixel-Größe in der der Graph, abgespeichert werden soll kann in den Voreinstellungen festgelegt werden. Der Graph wird dafür intern in der vollen Pixelgröße berechnet und aufgebaut.

Solange ein SIRDS-Graph-Fenster aktiv ist, stehen folgende Menüs zur Verfügung:

- Projekt - Das Projekt-Menü
- Graph
 - Aktualisieren - wie der entsprechende Bild-Button (s.o.)
 - Einstellen ... - s.o.
 - Drucken - s.o.
 - Speichern IFF... - s.o.

1.21 SIRDS-Graph-Einsteller

2.4.1 SIRDS-Graph-Einsteller

In diesem Einsteller wird der Darstellungsbereich, die Achsenskalierung und einige weitere Merkmale eines SIRDS-Graphen verändert.

Für die x-, y- und z-Achse kann jeweils ein Darstellungsintervall festgelegt werden. x- und y-Achse verlaufen später im Graphen auf einer imaginären Ebene, wohingegen die z-Achse dem Betrachter "auf magische Weise" entgegenzukommen scheint.

Mit Farben kann festgelegt werden, mit wievielen der verfügbaren Farben der Graph aufgebaut werden soll. Viele empfinden mehrere Farben angenehmer als einen monochromen Graphen (Farben=2), der seinerseits zum Ausdrucken besser geeignet wäre. Auf die Geschwindigkeit hat die Farbanzahl keinen Einfluß.

Die Transparenz bestimmt, wieviele Weiß-Anteile der Graph enthalten soll. Je höher der Wert, desto lichter und heller erscheint der Graph, was besonders beim Ausdrucken wichtig sein könnte.

Außerdem kann eine Genauigkeit ausgewählt werden, mit der ein Graph gezeichnet werden soll. Bei geringerer Genauigkeit werden weniger Funktionswerte berechnet und die Berechnung erfolgt in deutlich höherer Geschwindigkeit – jedoch wird der Funktionsgraph zuweilen etwas "eckiger" und der dreidimensionale Effekt erscheint "stufig". Trotzdem sind die geringen Genauigkeiten von großer Bedeutung, da das Berechnen eines SIRDS mitunter sehr lange dauern kann...

Graph-Einstellungen können mit Laden... und Speichern... unter beliebigen Namen gespeichert und später wieder geladen werden.

Nach Verlassen des Koordinatensystem-Einstellers mit Ok wird der aktive Graph aktualisiert, um die evtl. gemachten Änderungen sichtbar zu machen. Beim Verlassen durch Abbruch oder Schließen des Fensters bleibt es natürlich bei den vorherigen Werten.

1.22 Textanzeigefenster

2.5 Textanzeigefenster

Das Textanzeigefenster zeigt in Graph2D alle Arten von Texten in einem Listen-Gadget an. Mit dem Rollbalken kann man sich im Text bewegen und diesen mittels "Drucken" auch ausdrucken. Geschlossen wird das Fenster durch "Ok" oder dem Schließ-Symbol.

Ein Textanzeigefenster muß nicht unbedingt geschlossen werden, um mit Graph2D weiterzuarbeiten.

1.23 Voreinstellungseeditor

2.6 Voreinstellungseeditor

Hier können folgende Voreinstellungen für Graph2D festgelegt werden:

Verzeichnis	- Dieses Verzeichnis wird von den Laden/Speichern-Requestern in den Grapheditoren vorgegeben
2D-Graph	- Diese 2D-Grapheinstellungsdatei wird immer verwendet, wenn ein neuer Graph geöffnet wird
SIRDS-Graph	- Diese SIRDS-Grapheinstellungsdatei wird immer verwendet, wenn ein neuer SIRDS-Graph geöffnet wird
IFF-Breite und IFF-Höhe	- Ausmaße (in Pixel), die ein Graph bekommt, wenn er als IFF-Graphik abgespeichert wird
Schrift	- Diese Schrift wird in der math.~Notation verwendet
Druck-Schrift	- Schrift, in der die Notation ausgedruckt wird

Buttonleiste im Funktionseditor oder Graphen - Da die Bildbuttons nicht jedermanns Sache sind, können sie hier für die nämlichen Programmteile abgestellt werden - statt dessen erscheinen dann Text-Buttons oder auch gar keine Buttonleiste, was etwas Platz spart.

Die Voreinstellungen können mit "Benutzen" bis zum Programmende verwendet werden. Mit "Speichern" werden sie dauerhaft gemacht. Bei "Abbruch" passiert natürlich gar nichts.

1.24 Drucken

2.7 Druck-Requester

Mit diesem Requester kann ein Graph über die in den Workbench-Printer-Settings getätigten Einstellungen ausgedruckt werden.

Selbst wählen darf man hier die Breite und Höhe des auf dem Papier entstehenden Bildes in Millimetern (mm). Außerdem kann hier die Dichte des Ausdrucks eingestellt werden, was Vorrang vor dem gleichen Einsteller in den Workbench-Preferences hat.

Der Graph wird (abhängig von der gewählten Dichte) mehr oder weniger in der vollen Drucker-Auflösung intern erstellt und ausgedruckt. Bei hohen Auflösungen werden dabei die einzelnen Pixel mitunter sehr klein, so daß man z.B. für die Beschriftung auch schon mal einen sehr großen Font verwenden muß, um das ganze auch noch lesen zu können.

1.25 Funktion

3.1 Funktionen

Zum Thema Funktionen gibt es noch die folgenden Kapitel

- 3.1.1 ~Interne~Operatoren~&~Funktionen~
- 3.1.2 ~Gegenseitige~Einbindung~~~~~
- 3.1.3 ~EBNF-Syntax~~~~~

Eine Funktion muß in Graph2D aus diesen Teilen bestehen:

Name

Eine Funktion besteht zunächst einmal aus ihrem Namen. Dieser darf keine führenden Zahlen oder Leerzeichen enthalten und eine Größe von 30 Zeichen nicht überschreiten. Erlaubt sind Buchstaben (keine Umlaute), Zahlen (ab zweiter Stelle) und das einzelne Hochkomma (').

Argumentenliste

Direkt an den Namen schließt sich die Argumentenliste an. Sie besteht aus mehreren durch Kommata getrennten Variablennamen. Für Variablennamen gilt das gleiche wie für Funktionsnamen, in einer Argumentenliste dürfen sie

nicht doppelt vorkommen! Die Argumentenliste kann übrigens auch leer sein, wobei man sich dann sogar die Klammern sparen kann. Name und Argumentenliste bilden zusammen den sogenannten "Funktionsheader".

Das Zeichen "="

Zwischen Funktionsheader und Funktionstext steht das Zeichen "=". Wer hätte das gedacht...

Funktionstext

Die Definition des Funktionstextes erfolgt in der "üblichen" Notation für mathematische Terme, zwischen Groß- und Kleinschreibung wird (übrigens auch bei Variablen- und Funktionsnamen) unterschieden! Der Funktionstext besteht aus:

- Operatoren siehe Liste der internen~Operatoren
- Funktionen siehe Liste der internen~Funktionen

Außerdem können eigene Funktionen aus der Funktionsliste eingebunden werden.

- Konstanten

Eine Konstanten eine reelle Zahlen im Bereich $\pm 9.22337177E18$

- Variablen

Natürlich dürfen in einer Funktion nur die Variablen verwendet werden, die in der Argumentenliste namentlich aufgeführt sind.

Beispiele für richtige Funktionen:

```
"p(x)=x^2+98",
"Summe(a,b)=a+b",
"g=9.81",
"f(x)=sin(x^2)+e^(1/2*x)",
"funktion23a(iks, yps, zett)=iks^yps*zett+(-1.23+1.0356E-5)"
```

1.26 Interne Funktionen und Operatoren

3.1.1 Interne Funktionen und Operatoren

Bekannt sind folgende Funktionen und Operatoren:

```
sin(x), arcsin(x), sinh(x),
cos(x), arccos(x), cosh(x),
tan(x), arctan(x), tanh(x),
arctanh(x)    - Dies sind die trigonometrische Funktionen und ihre
                Umkehrfunktionen
abs(x)        - Der Absolutwert des Arguments, auch Betrag |x| genannt
int(x)        - Der ganzzahlige Teil des Arguments
```


<code>ln(x)</code>	- natürlicher Logarithmus (zur Basis 2)
<code>sqrt(x)</code>	- Die Quadratwurzel, also $x^{(1/2)}$
<code>sgn(x)</code>	- Die "Vorzeichenfunktion", liefert je nach $x > 0$, $x = 0$ oder $x < 0$ die Werte 1, 0 oder -1
<code>fak(x)</code>	- Die Fakultät ($x!$) z.B. <code>fak(4)=4*3*2*1=24</code> (nur sinnvoll bei positiven, ganzzahligen Werten)
<code>if(a, b)</code>	- liefert b zurück, wenn Argument a WAHR (=1) ist, ansonsten ist der Funktionswert 0. (nützlich für partiell definierte Funktionen)
<code>binom(n, k)</code>	- Binomialkoeffizient "n über k"
<code>exp(x)</code>	- Exponentialfunktion e^x
<code>e</code>	- Euler'sche Zahl 2.71...
<code>pi</code>	- 3.141...
<code>+</code> , <code>-</code>	- Addieren und subtrahieren wie gewohnt
<code>*</code> , <code>/</code>	- Multiplikation und Division
<code>^</code>	- Potenz-Operator (Vorsicht: $a^b^c = (a^b)^c$)
<code>=</code> , <code>></code> , <code><</code> , <code>>=</code> , <code><=</code> , <code><></code>	- Vergleichsoperatoren, z.B. gilt $a > b = 1$ falls $a > b$, ansonsten gilt $a > b = 0$
<code>AND</code> , <code>OR</code> , <code>NOT</code>	- Bool'sche Operatoren, momentan nur sinnvoll auf die boolschen Werte 1(=WAHR) und 0(=FALSCH) anwendbar, nicht bitweise.

1.27 Einbindung

3.1.2 Einbindung

Funktionen können sich gegenseitig einbinden, d.h. zu ihrer Definition andere Funktionen heranziehen. Dabei ist folgendes zu beachten:

Spätere Änderungen in der eingebundenen Funktion haben keine Auswirkungen mehr auf die einbindende Funktion, da sie zum Zeitpunkt des Interpretierens in die einbindende Funktion (intern) hineinkopiert wird. Um Änderungen auch in der einbindenden Funktion wirksam zu machen, muß diese lediglich neu interpretiert werden. (Im Textgadget einfach Return drücken.) Auf diese Weise wird ausgeschlossen, daß versehentlich "endlose" Rekursionen eingegeben werden.

Existieren mehrere Funktionen gleichen Namens, so ist eine eindeutige Zuweisung nicht mehr gewährleistet - es wird dann die erste Funktion aus der Liste der möglichen angenommen.

Beispiele:

<code>"Summe(a,b)=a+b"</code>	
<code>"f(x)=2^Summe(x,5)+x^2"</code>	richtig
<code>"k=5.6"</code>	
<code>"Blabla(x,y)=k*x+y+1"</code>	richtig
<code>"f(x)=g(x)/6"</code>	
<code>"g(x)=2^f(x)"</code>	falsch! (Zirkel!)

"u(x)=u(1/x)+3.14"

falsch! (Zirkel!)

1.28 EBNF

3.1.3 EBNF-Syntax

Leerzeichen sind (außer innerhalb von Bezeichnern) im gesamten Funktionstext erlaubt und können auch der Übersichtlichkeit dienen.

Zur Erläuterung: [x] : kein- oder einmaliges Vorkommen von x
 { x } : kein- ein- und mehrmaliges Vorkommen von x
 "xyz" : explizites Vorkommen der Zeichenkette xyz

Funktion = Bezeichner ["(" [Bezeichner { "," Bezeichner }] ")"]
 "=" Ausdruck .

Ausdruck = Priol [("=" | ">" | "<" | "<=" | ">=" | "<>")
 Priol] .

Priol = ["+" | "-"] Prio2 { ("+" | "-" | "OR") Prio2 } .

Prio2 = Prio3 { ("/" | "*" | "AND") Prio3 } .

Prio3 = ["NOT"] Prio4 { "^" Prio4 } .

Prio4 = Zahl | "(" Ausdruck ")" | Bezeichner | Aufruf .

Aufruf = Bezeichner ["(" Ausdruck { "," Ausdruck } ")"] .

Zahl = Ziffer { Ziffer } ["." Ziffer { Ziffer }]
 ["E" ["-"] Ziffer { Ziffer }] .

Bezeichner = Buchstabe { Buchstabe | Ziffer } .

Ziffer = "0" | .. | "9" .

Buchstabe = "A" | .. | "Z" | "a" | .. | "z" | "'".

1.29 Entwicklungsgeschichte

3.2 Entwicklungsgeschichte

V2.10 (03.03.95):

- Druck jetzt in wählbarer Größe und in voller Auflösung möglich
- Graphen und Notation können als IFF-Bild abgespeichert werden
- Buttonleiste abschaltbar bzw. gegen Textbuttons austauschbar
- eingebaute Konstanten "e" und "pi"
- es werden jetzt überall mehr Nachkommastellen ausgegeben

- Aufteilung des 2D-Grapheinstellers in zwei Fenster
- Unäres Vorzeichen: Ausgabe von z.B. "-x" anstatt "0-x"
- eindeutigere Klammerung bei Ausgabe von Potenzketten
- Graph-Fenster erinnern sich jetzt an ihre alte Position
- ein paar kosmetische Änderungen im Layout
- SIRDS-Berechnung mit ESC abbrechbar
- Überarbeitung der Anleitung

intern: - code-cleanup
- Mathematischer Teil überarbeitet, eingebundene Funktionen werden jetzt einkopiert, so daß keine fatalen Rekursionen mehr entstehen können

V2.00 (04.02.95):

- Kurvendiskussion ist jetzt schneller und erfolgreicher
neu: Kurvendiskussionsrequester
- neues Layout in vielen Teilen des Programms, Requester haben nun illustrierende Bilder, in Graphen und Funktionseditor gibt es platzsparende Bildbuttons
- neu: Funktionen mit Taylorreihe approximierbar
- einige schreckliche Fehler im mathematischen Teil behoben (falsche Ergebnisse, Klammern und Ableitungen), EBNF geändert
- neue interne Funktionen "binom(n, k)" und "exp(x)"
- optimiertes Speicherhandling und Geschwindigkeitssteigerung beim Berechnen von Funktionswerten
- Online-Hilfe neu gegliedert und erweitert, die HELP-Taste führt jetzt auch von den Requestern aus zum passenden Kapitel
- math. Notation behält jetzt ihre Funktion - auch wenn sie zwischenzeitlich geändert wurde
- Koordinatensystem zeichnet keine Striche mehr, wenn es zu viele sind
- ein weiterer Fehler nach dem Laden von Funktionen behoben...

V1.60 (08.01.95):

- Voreinstellungseeditor: Der Notations-Font, der Grapheinstellungspfad und Voreinstellungsdateien für die Graphtypen können angegeben werden
- Fehler nach Laden von Funktionslisten behoben
- 2D-Graph: der Font für Beschriftungen kann gewählt werden, neue Rasterart, Beschriftungen und Zwischenstriche werden schlauer gezeichnet und sollten nun fast immer gut aussehen
- math. Notation: benutzt jetzt den Punkt "." für die Multiplikation und das Ausrufezeichen "!" für die "fak"-Funktion. Außerdem stehen Menus zur Verfügung und die Notation kann gedruckt werden.
- Graph-Ausdrucke ohne Fensterrahmen
- alle Fenster passen wieder auf einen NTSC 640*200 / Topaz 8 Screen
- Änderungen in der Archivzusammensetzung

V1.50 (10.12.94):

- Lokalisierung: Es liegt nun ein englischer Katalog und eine englische Anleitung bei
 - Verbesserte Nullstellenfindung -> bessere Kurvendiskussion
 - Funktionen können nun mit verschiedenen Pattern gezeichnet werden
-

- Statusfenster an mehreren Stellen in Graph2D trösten über die Wartezeit und bieten Abbruchmöglichkeiten
- Fehler bei Ableitung von einbindenden Funktionen und bei der Klammerung von abgeleiteten/vereinfachten Funktionen behoben
- In der math. Notation werden für die Funktionen 'sqrt' und 'abs' nun auch Wurzelzeichen und Betragsstriche gezeichnet
- Behebung einiger kleinerer Fehler, die lieber ungenannt bleiben möchten
- Überarbeitung der Anleitung
(Für die Ausdrucker: es sind nun max. 75 Zeichen pro Zeile)
- Doppelklick auf Funktion toggelt nun "Gezeichnet"-Marke
- HELP bei Funktionseingabe zeigt sofort relevantes Kapitel der Anleitung an

intern: - Definitionsmodul für Applikation

V1.41 (16.10.94):

- Verbesserungen bei der Achseskalierung und -beschriftung
- Ergänzen von Funktionsästen, die sichtbaren Bereich verlassen
- bessere Vermeidung von Skalierungsfehlern bei Funktionsdarstellung
- Keine überflüssigen Nachkomma-Nullen mehr am Ende von Zahlen
- neue interne Funktionen nun auch ableitbar
- Keyfile darf endlich wieder in "PROGDIR:" sein :-)
- Beispielkoordinatensysteme mitgeliefert und Verzeichnisvoreinstellungen sowie Dateinamepattern in den Laderequestern
- kein unnötiges Text-Gadget mehr in SIRDS-Graph
- Tastaturshortcuts nun auch in Requestern
- Gadgetdisabling in 2D-Grapheinsteller
- Nach abgebrochenem Laden von Graphsettings bleiben die alten Werte bestehen
- Farbslider im Funktionseditor nun korrekt disabled und steuerbar
- Dokumentation hat wieder einen Index

V1.40beta (12.10.94):

- 2D-Graph und SIRDS-Graph sind nun zwei getrennte Graph-Typen mit verschiedenen Einstellungsfenstern und Menus
- stark erweiterte Gestaltungsmöglichkeiten für die Skalierung der Achsen bei 2D-Graphen
- neue interne Funktionen: abs, int, sinh, cosh, tanh, arctanh
- Funktionsbezeichner dürfen keine verrückten Zeichen mehr enthalten
- Anleitung um neue Funktionen erweitert und geändert
- Product-Info Datei erstellt
- Betaversion: längeres, langsames Hauptprogramm mit allen Checks

intern: - Modulumbenennungen und Modulheaderkommentare
- \$CopyArrays- bei einigen Prozeduren

V1.32 (18.09.94):

- "Darstellung" heißt nun "(math.) Notation", da es Verwechslungen mit der Darstellung einer Funktion als Graph gab.
 - neuer Menüpunkt "Hilfe..." (gleichet der HELP-Taste)
-

- Die Zahlen neben den Farbslider können jetzt zweistellig sein.
- Änderungen in der Dokumentation: Umordnung, neues Kapitel "Fragen & Antworten" und Benutzung von weiteren AmigaGuide3.0-Features

intern: - endlich mit V40-Interfaces kompiliert
- OS-Versionsstrings nun auch in Guide und Installskript

V1.31 (02.09.94):

- Ab 1.30 wurden Keyfiles im Programmverzeichnis nicht mehr erkannt. Ab sofort müssen Keyfiles im S: Verzeichnis des Systems liegen, damit es sicher funktioniert.
- Verbesserungen im Install-Skript und dem Install-Kapitel in der Dokumentation

V1.30 (01.09.94):

- Bereichsfehler beim Zoomen behoben
- korrekte Anzeige des Darstellungsintervalls in Graph-Titelzeile
- MagicWB-Icons für Graph2D hinzugefügt, die optional installiert werden
- Änderungen im Install-Skript (z.B. werden Drawer erzeugt)
- In der Popup-Graphenliste funktionieren nun auch die Cursortasten
- von nun an Compilation mit AmigaOberon V3.2, daher auch eine neue garbagecollector.library
- stellenweise Mißverständlichkeiten in der Anleitung verbessert, Änderungen an der .LiesMich und .ReadMe Datei

intern: - Verzicht auf Objektbindung an Hook -> globale Variable
- drawinfos ist nun ul.ListPtr anstatt ul.List

V1.22 (16.08.94):

- geringfügige Änderungen in Anleitung und Archivzusammensetzung

V1.21beta (15.08.94):

- Graph2D läuft nun erst ab MUI 2.1 oder höher. Die Menüs funktionieren nun (auch) unter dieser MUI-Version.
- Der Recorerable Alert bzw. Absturz beim Verlassen des Grapheinstellers ist jetzt hoffentlich endgültig beseitigt.
- Help-Taste funktioniert nun auch im Graph und im Darstellungsfenster
- Restrukturierung der Anleitung. Lest sie doch mal durch! :)
- Kleine Änderungen im Installskript
- Ein wenig Kosmetik an der Oberfläche und den Icons
- Nervrequester modifiziert
- längerer Code mit allen Checks, da Betaversion

intern: - neue MetaGUI, deshalb mehr statische Windows
- Änderungen der Koordinatensystem-Klasse

V1.20 (20.06.94):

- Graph2D ist nun SHAREWARE -> Nervrequester eingebaut
- Die Graph-Fenster werden jetzt zentral in einer Liste vom Funktionseditor aus verwaltet. Hier wird ab jetzt auch eingestellt, wie eine Funktion in welchem Graph erscheint. Das macht den alten "Funktionseinsteller" überflüssig. Außerdem hat jeder Graph hat jetzt einen eigenen Namen.
- Funktionswerte werden gepuffert. Das erhöht die Graph-Resize Geschwindigkeit enorm und ermöglicht flüssigeres Arbeiten. Die Einstellung der Darstellgenauigkeit ist noch abgestufter möglich.
- Auswahl, ob bei der Funktionsdarstellung Linien zwischen den einzelnen errechneten Funktionswerten gezeichnet werden soll.
- Das Drucken eines Graphen sollte nun wirklich funktionieren.
- Neues Dateiformat bei den Koordinatensystemeinstellungen, die alten Settings können leider nicht mehr geladen werden.
- Restrukturierung und Erweiterung der Dokumentation

intern: - Probleme (auf einigen Rechnern) mit Funktionsdarstellung beseitigt

- komplettes Rewrite der Koordinatensystem-Klasse
- Verbesserungen des CustomClass-Handlings und der Hooks

V1.10beta (25.05.94):

- 3D-Ansicht eingebaut, deswegen auch Änderung im Koords-Einsteller
- veränderbare Genauigkeit bei Graph-Zeichnung eingeführt
- Änderungen bei den bekannten Funktionen und Operatoren
- Änderung in der Behandlung von gegenseitiger Funktionseinbindung
- Überarbeitete Dokumentation im AmigaGuide-Format, ebenso Online-Hilfe eingebaut
- Installer-Skript geschrieben
- Rewrite des Parsers: strengerer Syntaxcheck, Leerzeichen erlaubt, Änderungen der EBNF
- div. kosmetische Änderung und neue Buttons, Verschiebepalken bei der Funktionsdarstellung

V1.00 (16.04.94):

- Skalierbarer Ausdruck eines Graphen ist jetzt möglich, auch der Ausdruck des Textanzeigefensters funktioniert hoffentlich
- nicht implementierte Funktionen nicht mehr anwählbar

intern: - Fehler bei y-Rasterung beseitigt

- Stackgröße nun auf 50000

V0.10beta (28.02.94):

- Umstellung auf MUI-Oberfläche ;-)
- Überarbeitung der Anleitung
- neues Programmicon

intern: - Funktionen ohne Namen verursachen keinen Müll mehr

- Verbesserungen bei der Graph-Rasterung

V0.01beta (19.01.94):

- Basisversion in Oberon

1.30 Benutzeroberfläche Mui

3.3 Benutzeroberfläche MUI

MUI-Copyright:

This application uses

MUI - MagicUserInterface

(c) Copyright 1993/94 by Stefan Stuntz

MUI is a system to generate and maintain graphical user interfaces. With the aid of a preferences program, the user of an application has the ability to customize the outfit according to his personal taste.

MUI is distributed as shareware. To obtain a complete package containing lots of examples and more information about registration please look for a file called "muiXXusr.lha" (XX means the latest version number) on your local bulletin boards or on public domain disks.

If you want to register directly, feel free to send

DM 30.- or US\$ 20.-

to

Stefan Stuntz
Eduard-Spranger-Straße 7
80935 München
GERMANY

Anmerkungen zu MUI in Graph2D:

Um eine (hoffentlich) optisch ansprechende und komfortable Oberfläche zu bieten, benutzt Graph2D das MUI-System von Stefan Stuntz. Für Graph2D muß deshalb MUI in der Version 2.1 oder höher installiert sein.

Man darf MUI benutzen, ohne sich dafür registrieren zu lassen. In der registrierten Version hat man allerdings den Vorteil, daß alle Einstellungen in dem zu MUI gehörende Preferences Programm abgespeichert werden können und so jede MUI-Applikation (also auch Graph2D ;-)) in ihrem Aussehen dauerhaft den eigenen Wünschen angepaßt werden kann! Die wichtigsten Einstellungen lassen sich allerdings auch in der nicht registrierten Version von MUI tätigen.

Möchte man beispielsweise, daß sich Graph2D nicht mehr auf dem Workbench-Screen öffnet, so kann mit den MUI-Preferences leicht ein anderer Public-Screen oder auch ein eigener Bildschirm eingestellt werden, auf dem Graph2D dann in Zukunft erscheinen wird.

Es empfiehlt sich also auf jeden Fall, auch die Anleitung(en) des MUI-Systems – insbesondere der MUI-Preferences – gründlich zu lesen. Besonders möchte ich an dieser Stelle noch auf ein paar Eigenarten aller MUI-Applikationen hinweisen, die natürlich auch für Graph2D zutreffen:

Fenster von MUI-Applikationen sind in der Größe veränderbar und vollständig fontsenstiv, d.h. sehen mit jedem eingestellten Zeichensatz gut aus!

MUI-Applikationen lassen sich durch ein Extra-Gadget in der Fenster-Titelleiste jederzeit ikonifizieren, d.h. alle zum Programm gehörenden Fenster schließen sich sofort, und auf der Workbench erscheint das Programmicon, durch dessen Aktivierung man jederzeit die Applikation wieder zum Leben erwecken kann...

MUI-Applikationen sind auch vollständig über die Tastatur bedienbar. Durch Tab-Cycling und Shortcuts lassen sich alle Gadgets aktivieren und einstellen, ohne die Maus zur Hand nehmen zu müssen. Das Gadget, das gerade Tastatureingaben empfängt, ist durch Umrahmung o.ä. markiert. Fenster lassen sich mittels Tastatur schließen (normalerw. durch ESC).

MUI-Applikationen sind als Commodities ins System eingebunden und lassen sich deshalb mit dem CommodityExchange-Programm steuern.

Vieles mehr, das in der MUI-Anleitung ausführlich beschrieben ist...

1.31 SIRDS

3.4 SIRDS

Graph2D verwendet zur 3D-Darstellung von Funktionen eine Technik, die üblicherweise als SIRDS (= Single Image Random Dot Stereogramm) bezeichnet wird. Man braucht keine Spezialbrille oder sonstige Hilfsmittel, um den 3D-Effekt zu erreichen, auch kann man diese Bilder ausdrucken oder kopieren, ohne daß sie an Wirkung verlieren.

Ich möchte an dieser Stelle nicht die Funktionsweise der Einzelbildstereogramme erläutern – wichtig erscheint mir für den Anwender von Graph2D nur das Erreichen des 3D-Eindrucks zu sein:

Das errechnete Bild besteht auf den ersten Blick nur aus einem wirren, mehr oder weniger farbigen Punkt-Muster. Man muß nun versuchen, den Blick ganz entspannt hinter die Bildebene zu lenken, z.B. auf die Wand hinter dem Monitor oder vielleicht auch auf das eigene Spiegelbild auf dem Bildröhrenglas. Natürlich sieht man dann die Bildpunkte, die gesamte Bildebene nicht mehr scharf, sie verschwimmt (auch sollte ein fester Punkt, z.B. der Mauszeiger doppelt erscheinen, doch das nur nebenbei). Beginnt dann bei richtigem Blick eine neue, dreidimensionale Ebene "im" Monitor zu erscheinen, darf man nicht den Fehler machen, sich sofort darauf zu

konzentrieren, denn dann verschwindet sie sofort wieder. Vielmehr muß man sich bemühen, weiterhin die momentane Augeneinstellung beizubehalten. Erst mit der Zeit ist man in der Lage, im Bild etwas "herumzuschauen" oder auch einmal zu zwinkern...

Verzweifelt nicht, wenn sich der Effekt nicht so schnell einstellen will! Das erste Mal – ich weiß es aus eigener Erfahrung – kann sehr lange dauernd und recht frustrierend sein. Doch wenn man es einmal geschafft hat, lohnt es sich wirklich, und der Effekt wird von nun ab auch immer leichter und schneller erreichbar sein. Die Augen sind bei der beschriebenen Methode im Grunde sehr entspannt, es strengt also auch bei längerem Betrachten nicht übermäßig an. Es sollte aber auch erwähnt werden, daß man auch durch Schielen den beschriebenen Effekt erzielen kann. Davon rate ich allerdings ab, da dies in der Regel sehr anstrengend ist, und man bedingt durch die Technik die Bilder invertiert sieht, also "hoch" und "tief" vertauscht wird.

Ich möchte mich an dieser Stelle bei Kilian Singer bedanken, dessen kleine, aber wirkungsvolle Routine diese Bilder in Graph2D ermöglicht!

1.32 Index

3.5 Index

- 2D-Graph
- 2D-Graph-Einsteller
- 3D-Darstellung

- A Achsen-Einsteller
- Autor

- B Bedienung
- bestimmte~Integral

- C Copyright

- D Druck

- E eingebunden

- F Fragen~&~Antworten
- Funktionen
- Funktionseditor

- G Geschichte
- Graph
- Graph2D.RegForm

- I Installation
- internen~Funktionen
- internen~Operatoren

- K Konto
- Kurvendiskussion

Kurzbeschreibung

M	math.-Notation MUI-System
N	Notations-Fenster
O	Originalarchiv
P	Projekt-Menü
R	Registrierung
S	SIRDS SIRDS-Graph SIRDS-Graph-Einsteller Sonstiges-Einsteller Syntax
T	Tangente Textanzeigefenster
V	Voreinstellungseditor
W	Wertetabelle
