Handbuch über die Bedienung und Funktion des Programms.

Programmierpraktikum 2011
Heinrich-Heine Universität Düsseldorf
Funktionsplotter

Gruppe 1:

Hatice Tavli
Tran Viet Dung
Michael Pott
Christian Wolf
Huu Tung Nguyen

Inhaltsverzeichnis:

6. „BestFit“ – technische Umsetzung

1. Vorwort

Diese Software wurde im Rahmen des Programmierpraktikums 2011 der Heinrich-Heine Universität Düsseldorf von den oben genannten Personen erstellt und dokumentiert. Das Projekt besteht aus der Umsetzung eines Funktionsplotters.
Die verwendete Programmiersprache ist Java. Zur Realisierung wurde die Open-Source-Entwicklungsumgebung „Eclipse“ verwendet.

Ziel der Anwendung ist es, bei mathematischen Problemstellungen schnell und unkompliziert einen Funktionsplotter nutzen zu können. Auch stellt dieses Programm eine kostenlose Alternative zu kommerziellen Produkten dar.

2. Einführung

1. Was ist ein Funktionsplotter?

Der ProPra-Plotter 2011 dient in erster Linie der graphischen Darstellung mathematischer Funktionen.

Die einzelnen Funktionswerte für verschiedene Werte von x aus einem bestimmten Intervall ergeben unterschiedliche Punkte im zweidimensionalen Koordinatensystem, an dem an der Abszisse die Werte für X und an der Ordinate die Funktionswerte (Y-Werte) aufgetragen werden.

Die Summe der einzelnen Punkte (=Graph) ergibt meist eine zusammenhängende Kurve.

Der ProPra-Plotter 2011 rechnet für bestimmte X-Werte aus einem angegebenen Intervall die Funktionswerte aus und zeichnet die berechneten Punkte in das Koordinatensystem.

Eine Wertetabelle, welche die eindeutige Zuordnung einzelner Funktionswerte zu ihren X-Werten ermöglicht kann ebenfalls erstellt werden.

2. Installation und technische Mindestanforderungen

Diese Software arbeitet, bedingt durch Java, plattformunabhängig. D. h., dieses Programmsollte auf verschiedenen Computersystemen mit Unterschieden in Architektur, Prozessor und Betriebssystem, bei installiertem Java (JRE), lauffähig sein.

Es ist lediglich die Installation von Java (mindestens JRE) notwendig um das Programm zu bedienen. Für den Funktionsplotter bedarf es keiner Installation, es muss lediglich eine ausführbare Datei gestartet werden.

Der ProPra-Plotter 2011 wurde erfolgreich auf unterschiedlichsten Rechnern und den Betriebssystemen Windows XP, Windows Vista, Windows 7 und Linux auf Funktionalität getestet.

3. Grundlagen zur Funktionsweise des Programms

Der Funktionsplotter basiert auf der objektorientierten Programmiersprache Java. Die Ausführung aller Operationen erfolgt durch den Anwender. Parameter, Einstellungen oder die Darstellung des Funktionsgraphen werden durch einen Klick auf den entsprechenden Button („Zeichnen“) ausgelöst. Für alle Berechnungen sind die gegebenen technischen Mittel verantwortlich.

3. Verwendung

1. Allgemeines

Der Funktionsplotter ist möglichst intuitiv gestaltet und sollte im Normalfall keine größeren Probleme bei der Bedienung bereiten.
Die gewünschten Werte und Paramter in den Feldern „Funktion“ , „Funktion mit Bedingung“, „Intervalloption“, „Plotteroptionen“ und „Optionsfeld“ eingegeben bzw. verändern und folgend auf den Button „Zeichnen“ am unteren Ende des Programms klicken. Daraufhin erhält man den errechneten Graph im Koordinatensystem , eine Wertetabelle und die automatisch ermittelten Nullstellen.

Der Button “Daten entfernen“ entfernt den / die Graphen aus dem Koordinatensystem. Somit ist ein neuer Vorgang mit anderen Werten möglich.

Der Button „Zeichnen“ führt die Berechnung aus und erstellt den / die Graphen im Koordinatensystem.

Der Button „Gitter“ blendet Gitternetzlinien im Koordinatensystem ein bzw. aus.

Auch stehen dem Nutzer die Menüpunkte „Datei“, „Bearbeiten“ und “ Hilfe„ zur Verfügung.

1. Das Programm beenden

Ein Klick auf den Menüpunkt „Datei“ und dessen Unterpunkt „Beenden“ beendet das Programm.

2. Einen neuen Plottvorgang starten

Ein Klick auf den Menüpunkt „Datei“ und dessen Unterpunkt „neuer Plotter“ entfernt den / die Graphen aus dem Koordinatensystem und ermöglicht so einen neuen Vorgang.
Hinweis: Der Button „Daten entfernen“ hat dieselbe Funktion.

3. Speichern von Ergebnissen

Ein Klick auf den Menüpunkt „Bearbeiten“ und dessen Unterpunkt „Bild exportieren“ ermöglicht es, den Kurvenverlauf als JPG-Bild auf der Festplatte zu speichern. Dazu sucht sich der Benutzer den gewünschten Speicherort auf seiner Festplatte aus und vergibt einen willkürlichen Dateinamen mit der Endung JPG z.B. „Grah.jpg“. Dieser Graph ist nun gespeichert und kann nun in anderen Programmen weiter verarbeitet werden.

4. BestFit

Durch einen Klick auf den Menüpunkt „Bearbeiten“ und dessen Unterpunkt „BestFit“, wird die BestFit-Option ausgelöst. Hiermit ist es durch eine Wertetabelle möglich, den Graphen des bestmöglichen approximierenden Polynoms $n-ten$ Grades sowie der betreffenden Polynomparameter auszugeben.
Um dies zu ermöglichen, stehen das Textfeld „Polynom“, der Menüpunkt „Anzahl der Startpunkte“ und die Buttons “Daten einlesen“ bzw. „Polynom bestimmen“ zur Verfügung.
Der Button “Daten einlesen“ ermöglicht das Öffnen der Textdatei „wert.txt“, welche es ermöglicht, eine Funktion aus vorgegebenen Werten ermitteln zu lassen.

Wird die Wertetabelle innerhalb der BestFit-Option editiert, ist es für einen reibungslosen Ablauf zwingend notwendig , diese Werte zuerst mit der Enter-Taste zu bestätigen!
Die Ermittlung des Polynoms wird in beiden Fällen durch einen anschließenden Klick auf den Button „Polynom bestimmen“ ausgelöst.
Das errechnete Ergebnis erscheint oben im Textfeld „Polynom“. Der Ausgabewert des Polynoms variiert, sobald man die Anzahl der Startpunkte unter „Anzahl der Startpunkte“ verändert.

5. Bedienungsanleitung

Mit einem Klick auf den Menüpunkt „Hilfe“ und dessen Unterpunkt „Handbuch“, kommt man zu diesem Dokument.

4. Einzelne Funktionen der Anwendung

1. Gruppe: Funktion

Hier werden die Eigenschaften für die Funktionsgraphen festgelegt, wobei drei Zeilen für die Graphen zur Verfügung stehen. Die Checkboxen am Anfang der Zeilen „Funktionsschar“ und „Geschaltete Funk“ legen fest, ob die jeweilige Funktion aktiviert ist oder nicht. Nicht aktivierte Funktionen werden von der Anwendung ignoriert.
Im Texteingabefeld wird für jede Funktion der Funktionsterm eingegeben. Die zu beachtenden Syntaxregeln (unter mathematische Elemente) sind unten beschrieben.
Ist für eine Funktion kein Term angegeben, so wird die entsprechende Checkbox automatisch deaktiviert.

2. Gruppe: Funktion mit Bedingung

Hierstehen dem Nutzer vier Zeilen zur Verfügung. In den Texteingabefeldern wird für jede Funktion der Funktionsterm eingegeben.
Die Checkbox am Anfang der Zeile „$f(x)$ mit zwei Bedingungen: a=“ legt fest, ob die jeweilige Funktion aktiviert ist oder nicht.
Des Weiteren stehen dem Nutzer die Felder „$F1(x < a)$“, „$F2(x > a)$“ und „$F(x = a)$“ für weitere Berechnungen zur Verfügung.
Es kann eine partiell definierte Funktion angegeben werden. D.h. für verschiedene Bereiche auf der X-Achse kann eine eigene Funktion definiert werden.

3. Gruppe Intervalloptionen

Mit Hilfe der Felder „Beginn“, „Ende“, „Minimum“ und „Maximum“ wird der Intervall des darzustellenden Graphen, je nach Option, beeinflusst. Die ganzzahligen Werte sind jeweils in das Texteingabefeld einzutragen. Die gewünschten Intervallwerte werden als Achsenbeschriftungen im Koordinatensystem dargestellt.

4. Gruppe: Plotteroptionen

Hier stehen die Optionen „Wendepunkte“ und „lokale Extremwerte“ zur Verfügung. Die Checkboxen am Anfang der Zeilen aktivieren die Optionen. Die aktivierte Option wird im Graph durch eine Markierung sichtbar gemacht.

5. Gruppe: Optionsfeld

Der Menüpunkt „Funktion wählen“ erlaubt es, eine zuvor eingegebene Funktion auszuwählen und diese, mit Hilfe der anderen Menüpunkte , zu editieren.

Unter „Farbe wählen“ ist es möglich, Funktionen unterschiedlich zu kolorieren (Rot, Blau, Grün, Gelb…) und somit voneinander abzuheben.

Der Programmpunkt „Funktion ausblenden“ blendet eine ausgewählte Funktion im Koordinatensystem aus bzw. ein.

Eine Ableitung einer zuvor ausgewählten Funktion wird durch den Programmpunkt „Ableitung“ ermöglicht.

Der Menüpukt „Integralfunktion“ ermittelt das Integral einer zuvor ausgewählten Funktion. Es können somit Flächen unter einem Graphen berechnet werden.

6. Das Koordinatensystem

Ein Klick auf den Bereich „Graphen“ (rechte Seite des Programmfensters) macht diesen Tab sichtbar. Mit ihm ist die Visualisierung der errechneten Werte in den vier Quadranten möglich. Somit können Probleme einer rechnerischen Behandlung an Hand ihrer Koordinaten an der x- und y-Achse zugänglich gemacht werden.

7. Wertetabelle

Ein Klick auf den Bereich „Wertetabelle“ (rechte Seite des Programmfensters) macht diesen Tab sichtbar. Anhand dieser Tabelle werden in zwei Zeilen die Argumente und die dazugehörigen Funktionswerte einer Funktion eingetragen. Ein Klick auf den Button „Berechnen“ errechnet den Wertebereich. Die Schrittweite kann im Texteingabefeld eingetragen werden.

8. Nullstellen

Ein Klick auf den Bereich „Nullstellen“ (rechte Seite des Programmfensters) macht diesen Tab sichtbar. Die Nullstellen einer Funktion werden automatisch errechnet und in diesem Tab dargestellt.

5. Mathematische Elemente

1. Funktionen

Der Propra-Plotter 2011 analysiert den Funktionsterm Zeichen für Zeichen von links nach rechts. Zeichenketten, die keine Übereinstimmungen in der Elementliste haben, werden ignoriert.

Syntax	**Name**	**Beschreibung**
$sin(x)$	Sinus	Liefert Sinus des Bogenmaßwertes *x*
$cos(x)$	Cosinus	Liefert Cosinus des Bogenmaßwertes *x*
$tan(x)$	Tangens	Liefert Tangens des Bogenmaßwertes *x*
cot	Cotangens	Liefert Cotangens von x
$arctan(x)$	Arcustangens	Liefert Arcustangens von *x*
$arcsin(x)$	Arcussinus	Liefert Arcussinus von *x*
$arccos(x)$	Arcuscosinus	Liefert Arcuscosinus von *x*
$arccot(x)$	Arcus Cotangens	Liefert Arkuscotangens von x
$sinsh(x)$	Sinus Hyperbolicus	Liefert Sinus Hyperbolicus von *x*
$cosh(x)$	Cosinus Hyperbolicus	Liefert Cosinus Hyperbolicus von *x*

$tanh(x)$	Tangens Hyperbolicus	Liefert Tangens Hyperbolicus von *x*
$coth(x)$	Cotangens Hyperbolicus	Liefert Cotangens Hyperbolicus von x
$arsinh\ (x)$	Areasinus Hyperbolicus	Umkehrfunktionen von Sinus Hyperbolicus
$arcosh(x)$	Areakosinus Hyperbolicus	Umkehrfunktionen von Kosinus Hyperbolicus
$artanh(x)$	Areatangens Hyperbolicus	Umkehrfunktionen von Tangens Hyperbolicus
$arcoth(x)$	Areakotangens Hyperbolicus	Umkehrfunktionen Kotangens Hyperbolicus
$sec(x)$	Secans	Winkelfunktion
$csc(x)$	Cosecans	Winkelfunktion
$arcsec(x)$	Arkussekans	Umkehrfunktion der Sekansfunktion
$arccsc(x)$	Arkuskosekans	Umkehrfunktionen der Kosekansfunktion
$sqrt(x)$	Quadratwurzel	Liefert Quadratwurzel aus *x*
pow	Potenzrechnung	Ermittlung von Potenzen
$log(x)$	Logarithmus	Liefert Logarithmus von *y* zur Basis *x*
$ln(x)$	Natürlicher Logarithmus	Liefert Logarithmus von *x* zur Basis e
$lg(x)$	Zehner-Logarithmus	Liefert Logarithmus von *x* zur Basis 10
$abs\ (x)$	Betrag	Absolutbetrag einer reellen Zahl

2. Variablen

Syntax	**Beschreibung**
x	Variable
a	Variable wird interpretiert, wenn Funktionsscharen aktiviert sind.

3. Konstanten

Syntax	**Name**	**Beschreibung**
E	Eulersche Zahl	2.7182818284590452354
PI	Kreiszahl PI	3.14159265358979323846

4. Klammern

Syntax	**Name**	**Beschreibung**
$(..)$	Klammern	Teile des Funktionsterms können zusammengefasst werden
$[..]$		
$\{..\}$		

5. Operatoren

Syntax	**Name**	**Beschreibung**
$x + y$	Addition	Addiert Summanden x und y
$x - y$	Subtraktion	Subtrahiert Subtrahend y vom Minuend x
$x * y$	Multiplikation	Multipliziert Faktoren x und y miteinander
x / y	Division	Dividiert Dividend x durch Divisor y
x ^ y	Exponent [1]	Potenziert Basis x mit Exponenten y

6. BestFit – technische Umsetzung

Die Idee hinter BestFit ist die Newtonsche Interpolation. Dazu eine kleine mathematische Erläuterung:

Gegeben sind n Punkte $(x0, y0), . . . (xn, yn)$, zu denen ein Polynom vom Grad n mit $p(xi) = yi$ für $i = 0; 1 n$ bestimmt werden soll.

Dazu betrachtet man das Newtonsche Interpolationsverfahren.

Betrachtet wird der Fall $n = 1$.

Gegeben sind die Punkte $(x0; y0); (x1; y1)$. Nun wird ein Polynom vom Grad 1, welches beide Punkte verbindet, gesucht:

$$p(x)y0 + (x - x0)\frac{y1 - y0}{x1 - x0}$$

Für den Fall $n = 2$ mit den Punkten $(x0; y0); (x1; y1); (x2; y2)$ bestimmen wir das Polynom 2.ten Grades mit Hilfe der obigen Gerade. Diese wird mit einem weiteren Summanden ergänzt und man erhält somit:

$$p(x) = y0 + (x - x0)\frac{y1 - y0}{x1 - x0} + a(x - x0)(x - x1)$$

Nun bestimmt man a so, dass $p(x2) = y2$ erfüllt ist:

$$p(x1) = y1 = y0 + (x1 - x0)\frac{y1 - y0}{x1 - x0}$$

$$p(x2) = y2 = y0 + (x2 - x0)\frac{y1 - y0}{x1 - x0} + a(x2 - x =)(x2 - x1)$$

Durch die Subtraktion beider Gleichungen erhält man also:

$$y2 - y1 = (x2 - x1)\frac{y1 - y0}{x1 - x0} + a(x2 - x0)(x2 - x1)$$

Dividiert man nun durch $(x2 - x0)(x2 - x1)$, so ergibt sich:

$$a = \frac{1}{x2 - x0}\left[\frac{\mathrm{y2 - y1}}{x2 - x1} - \frac{y1 - y0}{x1 - x0}\right]$$

Newtonsche Interpolationsformel

Das Tableau mit den dividierten Differenzen:

$$
\begin{array}{c|lllll}
x_0 & y_0 = y[x_0] & & & \\
 & & \delta y[x_0, x_1] & & \\
x_1 & y_1 = y[x_1] & & \delta^2 y[x_0, x_1, x_2] & \\
 & & \delta y[x_1, x_2] & & \delta^3 y[x_0, x_1, x_2, x_3] \\
x_2 & y_2 = y[x_2] & & \delta^2 y[x_1, x_2, x_3] & \\
 & & \delta y[x_2, x_3] & & \\
x_3 & y_3 = y[x_3] & & &
\end{array}
$$

Die Interpolationsformel dazu lautet:

$$
\begin{aligned}
p(x) = \; & y[x0] + (x - x0)\delta y[x0, x1] + (x - x0)(x - x1)\delta^2\, y[x0, x1, x2] \\
& + \cdots + (\mathrm{x} - \mathrm{x0})(\mathrm{x} - \mathrm{x1}) \ldots (\mathrm{x} - \mathrm{xn1})\delta^{\mathrm{n}}\, \mathrm{y}[\mathrm{x0}, \mathrm{x1}, \ldots\, \mathrm{xn}]
\end{aligned}
$$

Eine kleine Erläuterung zum verwendeten Algorithmus:

Im ersten Schritt werden die Eingabevektoren x, y auf die gleiche Länge überprüft. Ist dies nicht der Fall, gibt das Programm die Fehlermeldung „Eingabewerte überprüfen“ heraus.

Im zweiten Schritt werden die Länge der Eingabevektoren und folgende Fälle betrachtet:

Fall 1: Länge von $\vec{x} = 0$

Das Interpolationspolynom ist in diesem Fall ein Nullpolynom, d.h. return polynom = 0

Fall 2: Länge von $\vec{x} = 1$

Das Interpolationspolynom ist hier ein Konstantespolynom, d.h. return polynom = konstant = y

Fall 3: Länge von $\vec{x} > 1$

Berechne das Interpolationspolynom mit

1. double[ ] koeff = Interpolation(x, y)
2. String[ ] polynom = IPolynom$(koeff, x)$