

Marie Curie Schule
Am Sportplatz 1
30952 Ronnenberg

Facharbeit
im Seminarfach

Kleines L^AT_EX Tutorial Teil III

Verfasser: Henrik Maack und Paul Dawidow

Fachlehrer: Karsten Hohaus

Abgabetermin: 22. Dezember 2006

Vorwort

Sehr geehrte Leser, Leserinnen.

In der folgenden Facharbeit erfahren Sie einige wichtige Anwendungsmöglichkeiten, die Ihnen das Programm \LaTeX zur Verfügung stellt.

Wir möchten Ihnen gerne durch dieses Tutorial das Ergänzungsprogramm PSTricks zu \LaTeX vorstellen, welches die Erstellung von einfachen Koordinatensystemen und Einbindung von Funktionstermen ermöglicht. Des Weiteren wird in dem zweiten Teil dieser Facharbeit die Erstellung eines Literaturverzeichnisses erklärt.

Bei der Erstellung des Koordinatensystems wird zusätzlich auf die Beschriftung der Achsen oder auch Objekte innerhalb des Koordinatensystems eingegangen. Desweiteren erfahren Sie etwas zur Einbindung von Hilfslinien in das Koordinatensystem. Für die Einfügung von Funktionstermen gehen wir zunächst auf zwei verschiedene Varianten der Einbindung ein und zeigen Möglichkeiten zur optisch besseren Darstellung. Für das Literaturverzeichnis möchten wir Ihnen die Erstellung und Möglichkeiten der optimalen Darstellung näherbringen. Zum besseren Verständnis sind immer wieder Beispiele zur möglichen Anwendung eingebracht, welche Sie zur Kontrolle des richtigen Verständnis verwenden können. Scriptauszüge sind in grauen Kästen dargestellt und zeigen die neben stehenden Abbildungen.

Für die korrekte Anwendung unseres Tutorials setzen wir die Grundkenntnisse des Tutorials I[2]+ II[3] sowie die Programme „Ghostviewer“[4] und „Ghostscript“¹[4] voraus.

Am Ende dieser Facharbeit finden Sie noch einige Übungsaufgaben, um das Gelernte direkt einmal anzuwenden.

¹Die Programme „GhostScript“ und „GhostViewer“ werden benötigt um ps. Dateien richtig umwandeln zu können. Unter <http://www.cs.wisc.edu/~ghost/> können beide Programme kostenlos heruntergeladen werden.

Inhaltsverzeichnis

1 Grundlagen zur Facharbeit	4
1.1 PSTricks zur Darstellung von Abbildungen	4
1.2 Vorteile von PSTricks	4
1.3 Nachteile von PSTricks	4
1.4 Erstellung eines Literaturverzeichnisses	4
2 PSTricks zur Erstellung von einfachen Koordinatensystemen	5
2.1 Ein einfaches Koordinatensystem wird erstellt	5
2.2 Einteilung der Achsenlängen	6
2.3 Beschriftung der Achsen eines Koordinatensystems	7
2.4 Die Integration von Hilfslinien in ein beliebiges Koordinatensystem	9
3 Funktionen in das Koordinatensystem einfügen	11
3.1 Linien und Polygone zeichnen	11
3.2 Die Erstellung einfacher Funktionen über die Umgekehrte Polnische Notation	14
3.3 Das Ergänzungspaket pst-infixplot zur Zeichnung mathematischer Funktionen	15
3.4 Beschriftung von Graphen innerhalb eines Koordinatensystems	16
4 Die Erstellung eines Literaturverzeichnisses	17
5 Übungsaufgaben	18
6 Nachwort	20
7 Anhang	21

1 Grundlagen zur Facharbeit

1.1 PSTricks zur Darstellung von Abbildungen

Das Hilfsprogramm PSTricks bietet eine Vielzahl von Möglichkeiten zur Darstellung von komplexen Abbildungen. Fertig bearbeitete Abbildungen können über das „TeXnicCenter“ durch das Fenster `LaTeX =>DVI` als .dvi Datei abgespeichert und bereits durch anschließendes Drücken der Taste „F5“ betrachtet werden.

Zum Ausdrucken dieser .dvi Dateien muss jedoch ein spezieller Drucker verwendet werden. Um diese Dateien auch mit einem handelsüblichen Drucker als .pdf Datei ausdrucken zu können, werden die Dateien zunächst über das Fenster `LaTeX =>PS` als PostScript Dateien abgespeichert. Über ein externes Programm² können diese Dateien in .pdf Dateien umgewandelt werden und mit Hilfe eines Programms zur Darstellung³ von .pdf Dateien auch ausgedruckt werden.

1.2 Vorteile von PSTricks

- PSTricks bietet eine Vielzahl von verschiedenen Möglichkeiten, um Abbildungen zu erstellen und diese nach seinen eigenen Vorstellungen zu bearbeiten.
- Das PSTricks ist für die Erstellungen optisch hochqualifizierter Darstellungen gut geeignet.
- Darstellungsfehler können schnell und leicht herausgefiltert und verbessert werden.

1.3 Nachteile von PSTricks

- „PostScript“ Dateien können nicht direkt als .pdf Dateien von L^AT_EX angezeigt werden und müssen zuerst über ein Hilfsprogramm umgewandelt werden.
- Für ein optisch gutes Ergebnis müssen oft komplexe Befehle verwendet werden.
- Manche Befehle des Programms PSTricks können sich untereinander funktionell stören.

1.4 Erstellung eines Literaturverzeichnisses

Das Programm PSTricks ermöglicht die Darstellung von qualifizierten Literaturverzeichnissen und bietet Möglichkeiten für benutzerfreundliche Einstellungen.

²Das Programm „GhostViewer“ bietet die Möglichkeit zum betrachten von .ps Dateien und wandelt auf Wunsch auch in .pdf Dateien um

³Hierzu eignet sich das Programm „Adobe Reader“

2 PSTricks zur Erstellung von einfachen Koordinatensystemen

Um auf die Befehle des Paketes PSTricks zugreifen zu können, muss zu Beginn jeder Bearbeitung das Paket `\pst-all` eingebunden werden, um die Verwendung der PSTricks-Befehle zu ermöglichen.

```
\usepackage{pst-all}
```

2.1 Ein einfaches Koordinatensystem wird erstellt

Um ein einfaches Koordinatensystem erstellen zu können, benötigt man zunächst den Befehl `\psaxes`. Durch diesen Befehl können Einstellungen für die zu zeichnenden Achsen vorgenommen werden.

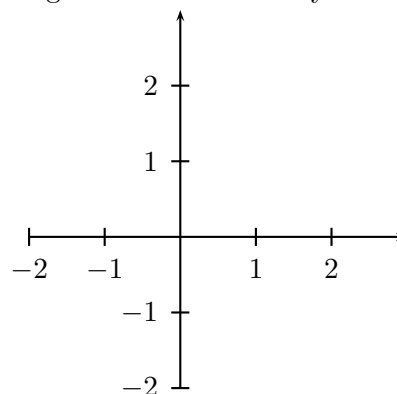
Allgemein kann dieser Befehl im Script wie folgt eingegeben werden:

```
\psaxes[<param>]{<arrows>}<x0>, <y0>(<x1>, <y1>)<x2>, <y2>
```

Das Script für ein einfaches Koordinatensystem:

```
\begin{document}
\psaxes{->}(0,0)(-2,-2)(3,3)
\end{document}
```

Als Ausgang ergibt sich folgendes Koordinatensystem:



Zu beachten ist jedoch, dass der Befehl `\psaxes` nicht die Längeneinheiten des Koordinatensystems festlegt, sondern die Achsen jeweils nur in Einserschritten unterteilt.

2.2 Einteilung der Achsenlängen

Bei der Betrachtung der .ps Datei kann es dazu kommen, dass sich Text und Abbildungen den selben Raum teilen und somit übereinander liegen. Zur Behebung dieses Problems bietet PSTricks über den Befehl `\begin{pspicture}` die Möglichkeit, Zeichnungen zu verschieben.

Allgemein lautet der Befehl :

```
\begin{pspicture}(<x0>, <y0>)(<x1>, <y1>)
\end{pspicture}
```

Über die beiden Variablen (`<x0>`, `<y0>`) wird die untere linke Ecke des Koordinatensystems festgelegt. Mit Hilfe der Variablen (`<x1>`, `<y1>`) wird die Verschiebung des Koordinatensystems von dem Punkt (`<x0>`, `<y0>`) aus in Höhe und Breite angegeben.

Man erhält durch den Befehl `\psaxes` ein Koordinatensystem, bei dem man die Längeneinheiten der Achsen noch nicht festlegen kann. Um dieses Problem zu beheben, wird der Befehl `\psset` benötigt.

Der allgemeine Befehl lautet :

```
\psset{param1=wert1, param2=wert2, ...}
```

Zur Erstellung eines Koordinatensystems mit einer x-Achse und einer y-Achse müssen die beiden Parameter `xunit=dim` und `yunit=dim` in die geschweifte Klammer eingefügt werden.

```
\psset{xunit=dim, yunit=dim}
```

Zu beachten ist, dass, sobald mehrere Parameter in eine Klammer eingefügt werden, diese durch ein Komma voneinander abgetrennt werden müssen.

Der Standardwert zur Einteilung der Längeneinheiten beträgt 1cm und kann nach Belieben verändert werden z.B. auf eine Längeneinheit der x-Achse von 1cm und der y-Achse von 0.5cm:

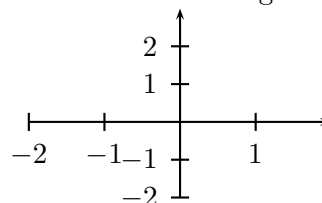
```
\psset{xunit=1cm, yunit=0.5cm}
```

Ein Koordinatensystem mit den oben genannten Längeneinheiten sähe beispielsweise so aus:

Im Script:

```
\psset{xunit=1cm, yunit=0.5cm}
\begin{document}
\psaxes{->}(0,0)(-2,-2)(2,3)
\end{document}
```

Als Abbildung:



Weiterhin ist es möglich noch weitere Parameter einzufügen, die z.B. Zeichenfarbe oder Linienbreite verändern. Dabei bleiben die Einstellungen des Befehls

`\psset{param1=wert1, param2=wert2, ...}` solange gültig, bis er erneut mit veränderten Parametern eingegeben wird.

2.3 Beschriftung der Achsen eines Koordinatensystems

Um das Koordinatensystem angemessen beschriften zu können, werden eine Vielzahl von verschiedenen Parametern zur Verfügung gestellt. Diese Parameter können direkt in die dafür vorgesehene Klammer im Befehl `\psaxes` eingefügt werden.

Die wichtigsten Parameter und dessen Auswirkungen sind in dieser Tabelle zusammengefasst:

Parameter	Auswirkung	Standart
<code>Ox=num Oy=num</code>	Beschriftung im Ursprung	0
<code>Dx=num Dy=num</code>	Zählerschritte für die Beschriftung der Achsen	1
<code>labels=all/x/y/none</code>	Definiert die Beschriftung der Achsen	all
<code>showorigin=true/false</code>	Beschriftung des Ursprungs	true

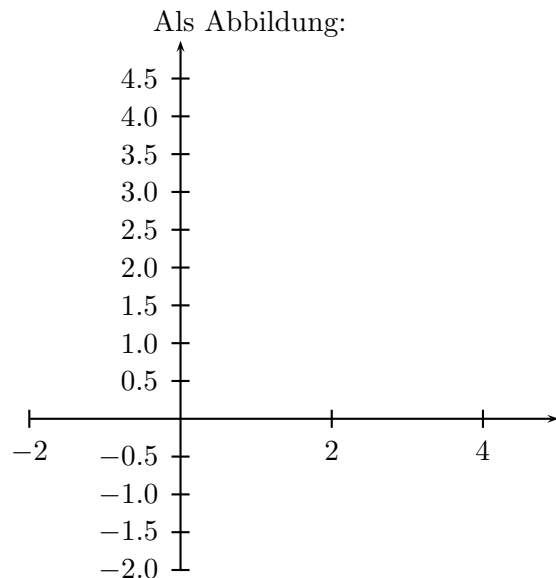
Zu beachten ist, sobald eine Achse auch negative Werte besitzt, wird der Achsenwert Null für diese Achse automatisch weggelassen.

Sollte keiner dieser Parameter aktiv eingegeben werden, so wird jedes Mal der Standartwert zur Darstellung durch \LaTeX verwendet.

Im folgendem Beispiel wird ein Koordinatensystem gezeichnet, indem die x-Achse in zweier Zählerschritten und die y-Achse in 0.5er Zählerschritte eingeteilt sein soll:

Im Script:

```
\psset{xunit=1cm, yunit=1cm}
\begin{pspicture}(0,0)(-4,-2)
\psaxes[showorigin=false, Dx=2,
Dy=0.5]{->}(0,0)(-2,-2)(5,5)
\end{pspicture}
```



Um eine Achse des Koordinatensystems zu beschriften, also einen Namen zu bestimmen, wird der Positionierungsbefehl `\uput` verwendet.

Allgemein wird der Befehl `\uput` so aufgerufen:

```
\uput{<labelsep>}[<refangle>]{<rotation>}<x>, <y>{<objekte>}
```

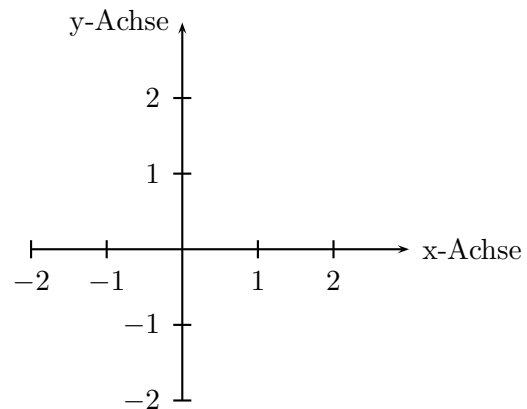
Dabei ist der Befehl `\uput` so aufgebaut, dass man über `{<labelsep>}` den Abstand zu der gewünschten Achse angibt, um zu verhindern, dass Beschriftungen zu dicht an der Achse stehen. Der Standardwert beträgt hierbei 5pt. Mit Hilfe des Parameters `{<refangle>}` kann man den Winkel einstellen, indem die gewünschte Beschriftung zur Achse stehen soll. Durch die Angabe des x- und y-Wertes wird die Positionierung der Beschriftung punktgenau festgelegt und über den letzten Parameter `{<objekte>}` die Beschriftung, also der Name, bestimmt.

Im folgenden Beispiel soll ein Koordinatensystem gezeichnet werden, welches an beiden Achsen beschriftet ist.

Im Script:

```
\begin{pspicture}(0,0)(-4,-2)
\uput[r](3,0){x-Achse}
\uput[l](0,3){y-Achse}
\end{pspicture}
```

Als Abbildung:



Für die Positionierung der Achsenbeschriftung wird noch ein zusätzlicher Parameter benötigt. Mit diesem Parameter wird der Winkel der Beschriftung an dem gewünschten Punkt(x,y) festgelegt.

Kürzel	steht für	Winkel
r	right	0
u	up	90
l	left	180
d	down	270
ur	up-right	45
ul	up-left	135
dl	down-left	225
dr	down-right	315

2.4 Die Integration von Hilfslinien in ein beliebiges Koordinatensystem

Zum genaueren Ablesen der Graphen bietet PSTricks eine Option zum Erstellen einfacher Hilfslinien in einem Koordinatensystem.

Der allgemeine Befehl dafür lautet:

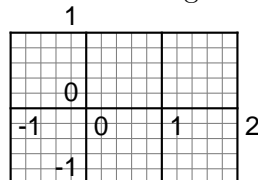
```
\psgrid[<param>] (<x0>, <y0>) (<x1>, <y1>) (<x2>, <y2>)
```

Über die beiden Variablen (x_0, y_0) wird innerhalb der Koordinatenbox ein Ursprung bestimmt, von dem aus, wie bei einem Koordinatensystem eine waagerechte Linie als x-Achse und die auf den Punkt senkrechte laufende Linie als y-Achse verwendet wird. Das zweite Variablenpaar (x_1, y_1) gibt die linke untere und das dritte Variablenpaar (x_2, y_2) die obere rechte Ecke der Koordinatenbox an.

Im Script:

```
\begin{pspicture}(-1,-1)(2,1)
\psgrid(0,0)(-1,-1)(2,1)
\end{pspicture}
```

Als Abbildung:



Mit den folgenden Parametern kann man diese Darstellung weiterhin gestalten.

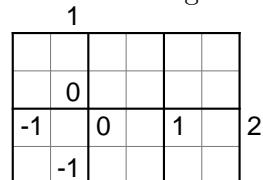
- **subgriddiv=int**

Über den Parameter subgriddiv lässt sich die Einteilung der Teilkästen pro Längeneinheit bestimmen. Der Standardwert beträgt hierbei 5.

Im Script:

```
\begin{pspicture}(-1,-1)(2,1)
\psgrid[subgriddiv=2]
(0,0)(-1,-1)(2,1)
\end{pspicture}
```

Als Abbildung:



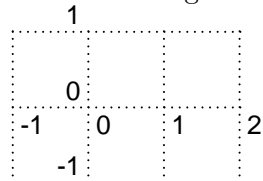
- **griddots=num**

Legt die Anzahl der Dots⁴ pro Längeneinheit fest. Der Standardwert beträgt 0, d.h. es wird eine Linie gezogen.

Im Script:

```
\begin {pspicture}(-1,-1)(2,1)
\psgrid[subgriddiv=1,griddots=10]
(0,0)(-1,-1)(2,1)
\end {pspicture}
```

Als Abbildung:



⁴Dots[engl.]: Punkte

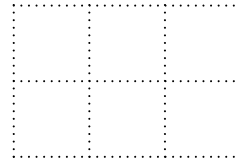
- **gridlabels=dim**

Durch diesen Parameter wird die Größe der Beschriftungszahlen der Koordinatenbox reguliert. Um zu verhindern, dass eine Doppelbeschriftung der Achsen bei Kombination von `\psaxes` und `\psgrid` auftritt, kann man diesen Parameter auf 0 setzen.

Im Script:

```
\begin {pspicture}(-1,-1)(2,1)
\psgrid[subgriddiv=1, griddots=10,
gridlabels=0] (0,0)(-1,-1)(2,1)
\end {pspicture}
```

Als Abbildung:



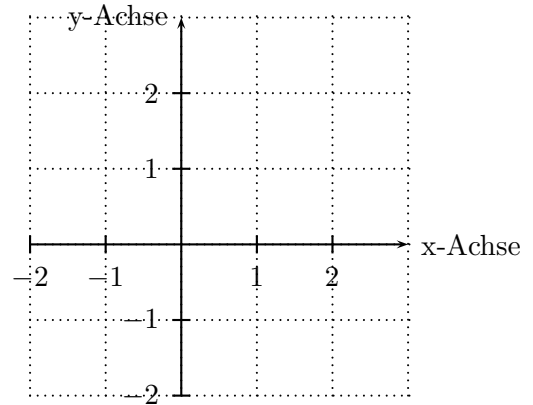
Um nun diese Koordinatenbox in ein zuvor erstelltes Koordinatensystem einzufügen ist noch zu beachten, dass die Grenzwerte der Koordinatenbox mit denen des Koordinatensystems übereinstimmen müssen.

Die Kombination davon sieht zum Beispiel so aus:

Im Script:

```
\psaxes{->}(0,0)(-2,-2)(3,3)
\uput[r](3,0){x-Achse}
\uput[l](0,3){y-Achse}
\psgrid[subgriddiv=1, griddots=10,
gridlabels=0](0,0)(-2,-2)(3,3)
```

Als Abbildung:



3 Funktionen in das Koordinatensystem einfügen

Über das Hilfsprogramm PSTricks stehen eine Vielzahl von Möglichkeiten zur Verfügung einfache Linien bis hin zu komplexeren Funktionsgraphen in ein Koordinatensystem einzufügen. Dabei werden die Funktionsgraphen automatisch an die Größenverhältnisse des Koordinatensystems angepasst.

3.1 Linien und Polygone zeichnen

Für die Erstellung einer Linie anhand von vorgegebenen Punkten bietet das Hilfsprogramm PSTricks den Befehl `\psline` an.

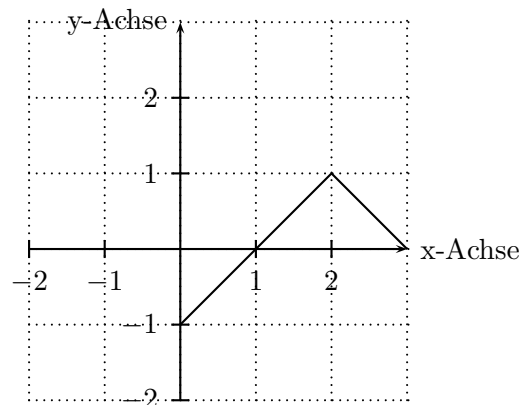
Der allgemeine Aufruf lautet:

```
\psline[<param>]{<arrows>}<x0>, <y0> <x1>, <y1> ... <xn>, <yn>}
```

Im Script:

```
\psaxes{->}(0,0)(-2,-2)(3,3)
\psline{-}(0,-1)(2,1)(3,0)
```

Als Abbildung:



Um eine solche Linie weiter auszustatten werden folgende Parameter zur Verfügung gestellt:

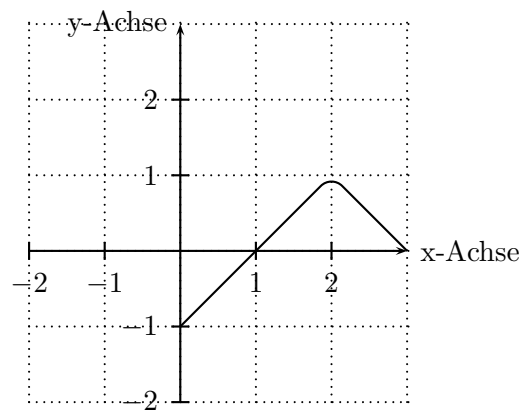
Parameter	Beschreibung	Standartwert
<code>linewidth=dim</code>	Legt die Dicke einer Linie fest	0.8pt
<code>linecolor=color</code>	Legt die Zeichenfarbe fest.	black
<code>linearc=dim</code>	Legt den Radius an den Eckpunkten fest	0

Im folgendem Beispiel wird durch den Parameter `linearc=dim` der Eckpunkt der Linie bei (2,1) abgerundet.

Im Script:

```
\psaxes{->}(0,0)(-2,-2)(3,3)
\psline[linearc=0.2]
{-}(0,-1)(2,1)(3,0)
```

Als Abbildung:



Mit dem Hilfsprogramm PSTricks lassen sich auch weitere Körper, wie das Polygon benutzerdefiniert zeichnen. Hierfür wird zunächst einmal das Paket `pst-poly` benötigt. Durch die Einbindung von zahlreichen Parametern, die \LaTeX zur Verfügung stellt, kann man das Polygon sehr vielseitig konstruieren. Bevor man jedoch ein Polygon erstellt, muss man mit dem Befehl `\rput` die Position des Körpers bestimmen. Allgemein wird dieser Befehl wie folgt abgerufen:

```
\rput[<param>](<x>,<y>)
```

Nachdem man die Position des Körpers bestimmt hat, wird mit Hilfe des Befehls `\PstPolygon` ein Polygon mit Standarteinstellungen erstellt. Dabei werden die Einstellungen für das `PstPolygon` direkt hinter den Befehl `\rput` in eine geschweifte Klammer gesetzt.

```
\rput[<param>](<x>,<y>){\PstPolygon[<param>]}
```

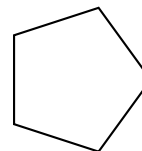
Gibt man keine Parameter für das Polygon an, so wird ein Polygon mit Standarteinstellungen erstellt.

Beispiel:

Im Script:

```
\rput(7,2){\PstPolygon}
```

Als Abbildung:



Über die nun folgenden Parameter lässt sich dieses Polygon noch nach Belieben gestalten.

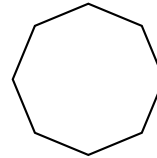
- **PolyNbSides=int**

Dieser Parameter legt die Anzahl der Ecken des Polygons fest, der Standardwert beträgt 5.

Im Script:

```
\rput(7,1){\PstPolygon[PolyNbSides=8]}
```

Als Abbildung:



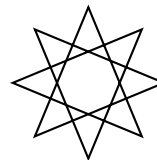
- **PolyOffset=int**

Durch Einbindung dieses Parameters wird die Anzahl der Ecken festgelegt, die beim Verbinden übersprungen werden.

Im Script:

```
\rput(7,1){\PstPolygon[PolyNbSides=8,
PolyOffset=3]}
```

Als Abbildung:



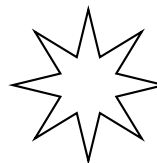
- **PolyIntermediatePoint=num**

Dieser Parameter legt den Abstand zum Mittelpunkt zweier Ecken fest.

Im Script:

```
\rput(7,1){\PstPolygon[PolyNbSides=8,
PolyIntermediatePoint=0.4]}
```

Als Abbildung:



Die von \LaTeX gezeichneten Körper kann man durch das Hinzufügen von zwei Parametern mit beliebiger Farbe ausfüllen. Mit dem Parameter `fillstyle=none` bestimmt man die Art und mit `fillcolor=color` die Farbe der Füllung.

Im Script:

```
\rput(7,1){\PstPolygon[fillcolor=yellow,
fillstyle=solid,
PolyNbSides=8, PolyIntermediatePoint=0.4]}
```

Als Abbildung:



3.2 Die Erstellung einfacher Funktionen über die Umgekehrte Polnische Notation

Die Programmiersprache PostScript bietet zwei verschiedene Wege zum Einbinden von mathematischen Funktionen. Mit dem Befehl `\psplot` können mathematische Funktionen dargestellt werden. Der allgemeine Aufruf dafür lautet:

```
\psplot [<param>] {<xmin>} {<xmax>} {<funktion>}
```

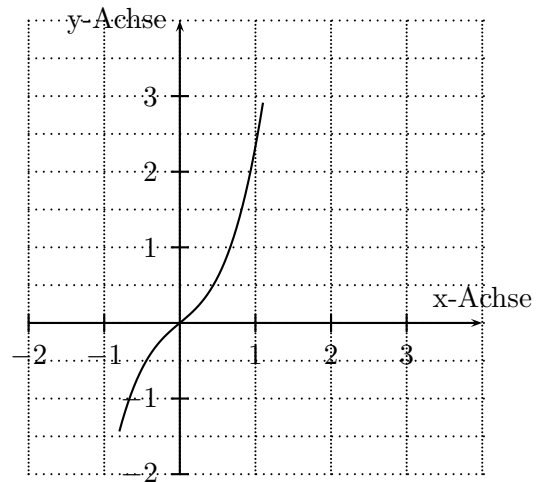
Dabei werden für `xmin` und `xmax` die Extremwerte festgelegt. Zur Eingabe der Funktion wird jedoch die „Umgekehrte Polnische Notation“ verwendet.

Bei der „Umgekehrten Polnischen Notation“ werden zuerst die Variablen angegeben und erst dann die Rechenoperation, die durchgeführt werden soll. So werden z.B. bei der Summe erst die beiden Summanden genannt und erst dann das Summenzeichen. Beispielsweise wird über die „Umgekehrte Polnische Notation“ die Funktion $3x^3 + 2x - x/3$ wie folgt eingegeben:

Im Script:

```
\psplot{-0.8}{1.1}{x 3 exp 3 mul x 2 mul add x 3 div sub}
```

Als Abbildung:



In der folgenden Tabelle sind die wichtigsten Rechenoperatoren, die für die „Umgekehrte Polnische Notation“ benötigt werden, zusammengefasst.

Funktion	Beschreibung	Beispiel
add	Addition	2 2 add
sub	Subtraktion	4 2 sub
mul	Multiplikation	1 8 mul
div	Division	9 3 div
exp	Potenz	2 3 exp
sqrt	Quadratwurzel	16 sqrt
sin	Sinus	45 sin
cos	Cosinus	60 cos
ln	Nat. Logarithmus	7 ln
log	Zehner-Logarithmus	4 log

3.3 Das Ergänzungspaket pst-infixplot zur Zeichnung mathematischer Funktionen

Die zweite Variante mathematische Funktionen von \LaTeX zeichnen zu lassen, bietet das Ergänzungspaket `pst-infixplot`. Hierbei werden die Funktionen direkt eingebunden, ohne vorher die „Umgekehrte Polnische Notation“ anwenden zu müssen. Um mit diesem Ergänzungspaket arbeiten zu können muss das folgende Paket zunächst eingebunden werden:

```
\usepackage{infix-RPN}
```

Der allgemeine Befehls lautet:

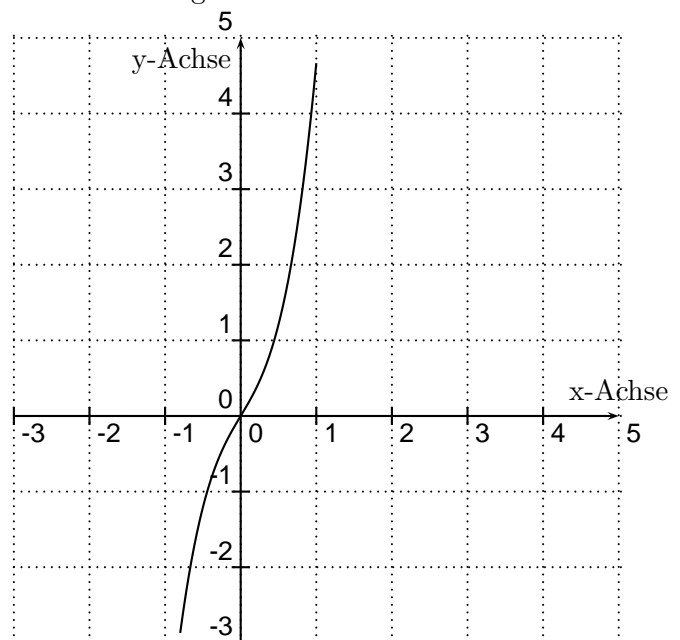
```
\infixtoRPN{<funktion>}
```

Um den Graphen der Funktion $3x^3 + 2x - x/3$ mit dem Befehl `\infixtoRPN` zu zeichnen, geben wir die Funktion so wie sie ist in die Klammer `{<funktion>}` ein. Der Befehl `\psplot` wird hierbei nur für die Bestimmung der Extremwerte und gegebenenfalls zur optischen Gestaltung (Farbe, Linienbreite, etc.) verwendet. Damit sich die `\psplot` Einstellungen auf den Graphen von `\infixtoRPN` übertragen, muss hinter den Befehl `\psplot` zusätzlich in geschweifte Klammern `{\RPN}` hinzugefügt werden.

Im Script:

```
\infixtoRPN{3x^3 + 2x - x/3}
\psplot{-0.8}{1} {\RPN}
```

Als Abbildung:



3.4 Beschriftung von Graphen innerhalb eines Koordinatensystems

Eine Möglichkeit zur Beschriftung eines Graphen im Koordinatensystem bietet der Befehl `\uput`. Es ist jedoch zu beachten, dass man zum Beginn und Ende von Zahlen und Rechenoperatoren ein „\$“ einbindet.

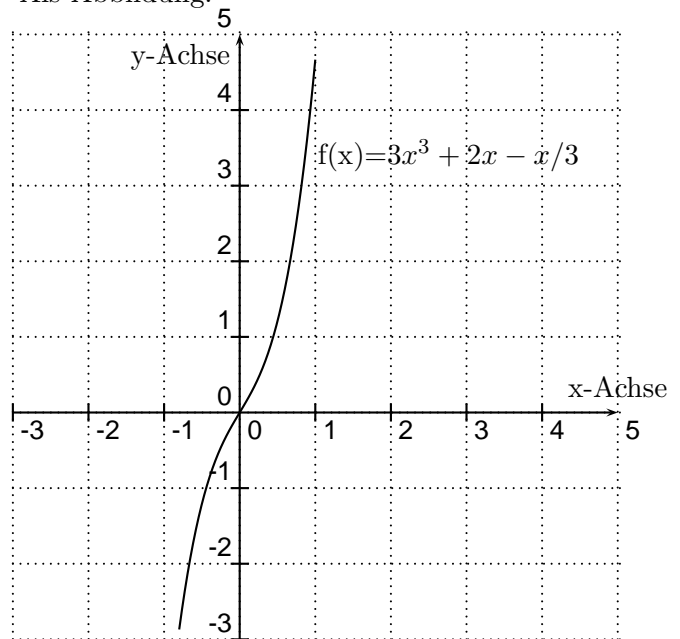
Beispiel:

Die Beschriftung wird mit Hilfe von `\uput` in das Koordinatensystem geschrieben.

Im Script:

```
\infixtoRPN{3x^3 + 2x - x/3}
\psplot{-0.8}{1} {\RPN}
\uput{1pt}[r](1,3.4)%
{f(x)=$3x^3 + 2x - x/3$}
```

Als Abbildung:

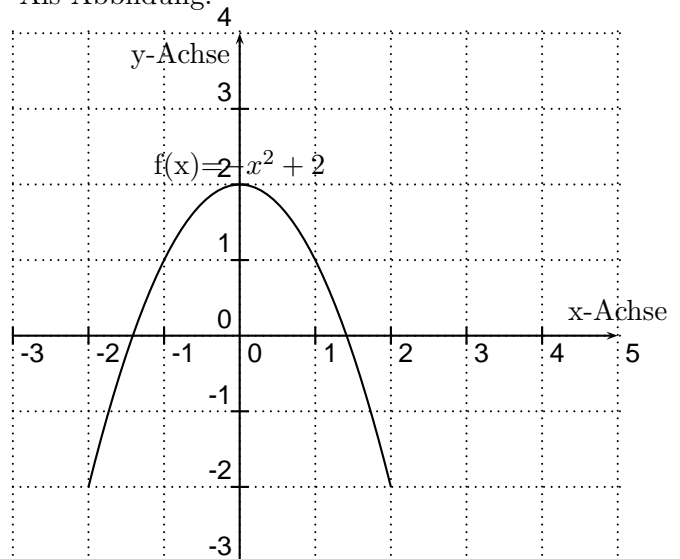


Ein anderes Beispiel anhand der Funktion „ $f(x) = -x^2 + 2$ “. Hier soll nun sich die Beschriftung oberhalb des Punktes (0,2) angebracht sein.

Im Script:

```
\infixtoRPN{(x^2) * (-1) + 2}
\psplot{-2}{2} {\RPN}
\uput{1pt}[u](0,2)%
{f(x)=$-x^2 + 2$}
```

Als Abbildung:



4 Die Erstellung eines Literaturverzeichnisses

Jede wissenschaftliche Ausarbeitung benötigt ein Literaturverzeichnis für die Referenzliteratur. \LaTeX bietet hierfür die Umgebung `thebibliography`.

Der allgemeine Aufruf dafür lautet:

```
\begin{thebibliography}{<mustermarke>}
\bibitem{<bezug>}<literaturangabe>
...
\end{thebibliography}
```

Das Literaturverzeichnis kann an beliebiger Stelle im Dokument platziert werden, üblicherweise wird es jedoch im Anhang eingefügt. Der Parameter `{<bezug>}` muss ein Name sein, der beim Zitieren mit dem Befehl `\cite{<bezugsmarke>}` im Text exakt übereinstimmt. Über die Angabe des Parameters `{<mustermarke>}` wird die maximale Anzahl der Literaturverweise angegeben.

Hier ein Beispiel:

... bei Lügnern ist das Lächeln links meist ausgeprägter als rechts (vgl. `\cite{1}`).
 ... Emotionen werden durch spontane Körperbewegungen ausgedrückt... (vgl. `\cite{2}`).

Im Script:

```
\begin{thebibliography}{99}
\bibitem{1} Allan \& Barbara Pease:
Die kalte Schulter und der warme Händedruck, Ulstein Buchverlage GmbH,
Berlin, 2004
\bibitem{2} Samy Molcho:
Körpersprache, Wilhelm Goldmann Verlag, München, 1998
\end{thebibliography}
```

Als Abbildung:

Literatur

- [1] Allan & Barbara Pease, Die kalte Schulter und der warme Händedruck, Ulstein Buchverlage GmbH Berlin, 2004
- [2] Samy Molcho, Körpersprache, Wilhelm Goldmann Verlag München, 1998

5 Übungsaufgaben

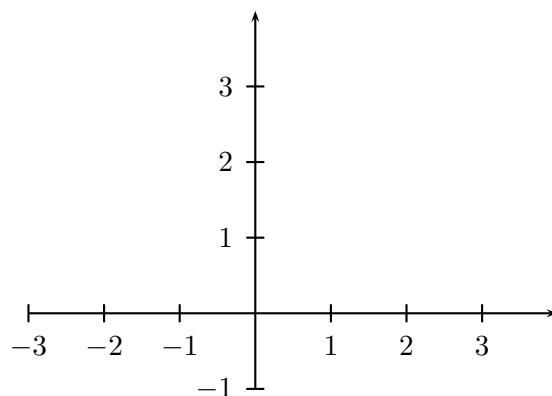
Diese Übungsaufgaben sollen das bisher gelernte verfestigen, daher ist es empfehlenswert zunächst die Aufgaben ohne Zuhilfenahme der Facharbeit zu bearbeiten. Falls trotzdem Schwierigkeiten auftreten sollten, kann diese immernoch zur Hilfe genommen werden. Die richtigen Scriptauszüge zu den Aufgaben befinden sich auf Seite 22.

1. Aufgabe

Zum der Übungen soll ein einfaches Koordinatensystem mit Standartinstellungen der Achsen erstellt werden:

1. Die x-Achse soll von -3 bis 4 bestimmt sein.
2. Die y-Achse soll von -1 bis 4 bestimmt sein.

Das Ergebnis:

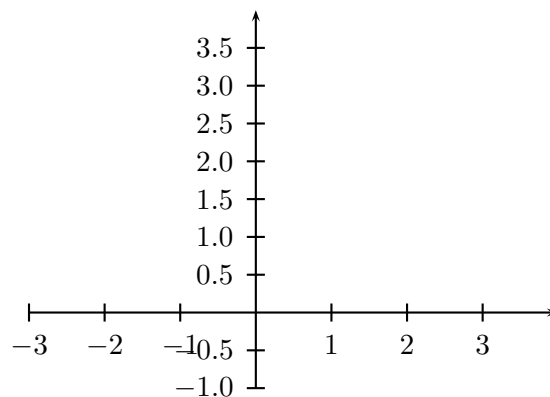


2. Aufgabe

Weiterhin soll das Koordinatensystem wie folgt verändert werden:

1. Das Koordinatensystem soll um 1cm nach Rechts verschoben werden.
2. Die y-Achse in 0.5er Schritten eingeteilt werden.

Hier das Ergebnis:

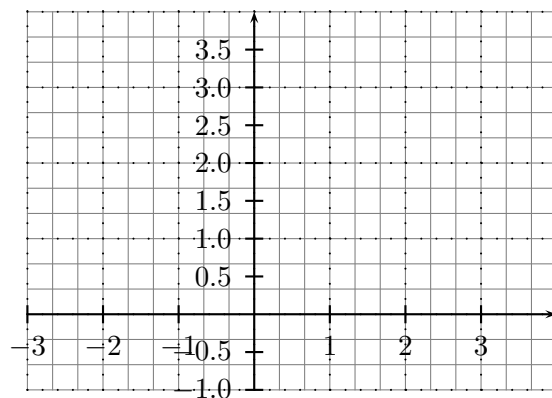


3.Aufgabe

Zu dem bereits erstellten Koordinatensystem, soll eine Koordinatenbox mit folgenden Werten hinzugefügt werden:

- Pro Längeneinheit der Achsen sollen drei Teilkästen eingebunden sein.
- Pro Längeneinheit soll die Anzahl der Dots auf 5 gesetzt werden.
- Die Beschriftungszahl der Koordinatenbox soll auf 0 gesetzt werden

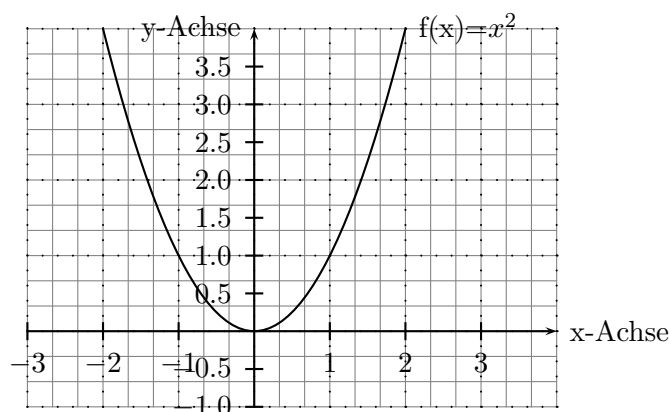
Daraus ergibt sich:



4.Aufgabe

- Die x-Achse soll rechts und die y-Achse links vom letzten positiven Achsenpunkt beschriftet werden.
- Der Graph von $f(x)=x^2$ soll mit Hilfe des Paketes pst-infixplot von $x_{min}=-2$ und $x_{max}=2$ eingebunden werden.
- Der Graph soll rechts vom Punkt (2,4) mit der Beschriftung „ $f(x)=x^2$ “ versehen werden.

Daraus ergibt sich folgendes Ergebnis:



6 Nachwort

Für die Erstellung dieses Tutorials über das Hilfsprogramm PSTricks haben wir uns viel Zeit genommen, um die Ausarbeitung möglichst präzise und leicht verständlich zu gestalten. Zu Beginn unserer Arbeitsphase mit dem Hilfsprogramm PSTricks hatten wir Schwierigkeiten mit der Umsetzung der Hilfsprogramme, die für PSTricks benötigt werden. Bei unserer weiteren Bearbeitung sind Schwierigkeiten mit der Umwandlung von .tex Dateien zu .pdf Dateien aufgetreten. Zu einigen der angegebenen Befehle mussten wir uns aus verschiedenen Quellen Information zusätzlich erarbeiten, um diese fachgerecht umsetzen zu können.

Uns ist aufgefallen, dass PSTricks mehrere umständliche Mittel zur Bearbeitung von Aufgaben anbietet, wobei es weitaus weniger umständliche Möglichkeiten zur gleichwertigen Bearbeitung durch PSTricks gibt. Beispielsweise kann anstatt der sehr komplexen „Umgekehrten Polnischen Notation“ auch das leichter zu handhabende Ergänzungspaket `pst-infixplot` verwendet werden, um Funktionsgraphen darzustellen. Trotz einiger Schwierigkeiten ist jedoch das Hilfsprogramm PSTricks sehr gut geeignet, um qualitativ hohe Abbildungen zu erstellen.

Die vielen Möglichkeiten diese Abbildungen seinen Vorstellungen anzupassen, sprechen für die Verwendung des Hilfsprogramms PSTricks, jedoch wird dafür auch ein entsprechend großes Hintergrundwissen benötigt.

Für die Erstellung eines Literaturverzeichnisses ist wesentlich weniger Hintergrundwissen notwendig und deshalb ist der Umgang leicht zu erlernen. Es werden nur wenige Befehle benötigt, um ein gutes Literaturverzeichnis zu erstellen, die uns auch in der Anwendung als nicht besonders schwer anwendbar aufgefallen sind.

7 Anhang

Literatur

- [1] Niedermair, Elke & Michael: \LaTeX Das Praxisbuch 2. vollständige aktualisierte und überarbeitete Neuauflage. Poing: Franzis 2005
- [2] Niklas Peschke & Jan Betting: Kleines \LaTeX Tutorial Teil I, Hannover, 2006
- [3] Wiebke Mattekant & Asja Lippert, Kleines \LaTeX Tutorial Teil II, Hannover, 2006
- [4] <http://forum.developers-guide.net/index.php>⁵

Aufteilung der Bearbeitung dieser Facharbeit

Diese Facharbeit wurde von Henrik Maack & Paul Dawidow zusammen verfasst. Henrik Maack hat die Seiten 1 bis 10 und 18 & 19 verfasst.
Paul Dawidow hat die Seiten 11 bis 17 und 20 bis 23 verfasst.

⁵Dies ist ein Online Forum. Wir haben Informationen aus der Rubrik „Sonstige Programmiersprachen“ (<http://forum.developers-guide.net/forumdisplay.php?f=29>) bezogen.

Lösungen der Übungsaufgaben

Lösung zu Aufgabe 1

```
\documentclass{scrartcl}
\usepackage{pst-all}
\begin{document}
\psaxes{->}(0,0)(-3,-1)(4,4)
\end{document}
```

Lösung zu Aufgabe 2

```
...
\begin{document}
\begin{pspicture}(0,0)(-1,0)
\psaxes[Dy=0.5]{->}(0,0)(-3,-1)(4,4)
\end{document}
```

Lösung zu Aufgabe 3

```
...
\begin{document}
\begin{pspicture}(0,0)(-1,0)
\psgrid[subgriddiv=3, griddots=5,
gridlabels=0](0,0)(-3,-1)(4,4)
\psaxes[Dy=0.5]{->}(0,0)(-3,-2)(4,3)
\end{document}
```

Lösung zu Aufgabe 4

```
...
\usepackage{infix-RPN}
\begin{document}
\begin{pspicture}(0,0)(-1,0)
\psgrid[subgriddiv=3, griddots=5,
gridlabels=0](0,0)(-3,-1)(4,4)
\psaxes[Dy=0.5]{->}(0,0)(-3,-2)(4,3)
\infixtoRPN{x^2}
\psplot{-2}{2}{\RPN}
\uput[r](2,4){f(x)=$x^2$}
\uput[r](4,0){x-Achse}
\uput[l](0,4){y-Achse}
\end{document}
```

Versicherung der selbständigen Erarbeitung

Hiermit versichern wir, dass diese Facharbeit von uns selbstständig erarbeitet wurde, nur die angegebenen Quellen und Hilfsmittel zur Bearbeitung verwendet worden und unserem Fachlehrer zugänglich sind.

Hannover, den 19. Dezember 2006

Henrik Maack

Paul Dawidow

Wir stimmen der Veröffentlichung unserer Arbeit durch Ausstellung in der Schülerbibliothek der Marie Curie Schule Ronnenberg zu.

Die Erklärung wird nur wirksam, wenn die Facharbeit mit mindestens 08 Punkten bewertet wird.

Henrik Maack

Paul Dawidow