

Das randbild-Package – Mini-Kurven auf dem Rand*

Paul Ebermann[†]

15. Mai 2007

Zusammenfassung

Dieses Paket ermöglicht es mit den beiden Umgebungen `randbild` und `randbildbasis`, bequem kleine Graphen von Kurven am Seitenrand zu setzen.

Inhaltsverzeichnis

1 Benutzerdoku	1
1.1 Befehle	1
1.2 Bildkoordinaten	2
2 Implementation	2
3 Liste der Änderungen	4
4 Index	4

1 Benutzerdoku

Dieses Paket wird wie üblich mit `\usepackage{randbild}` eingebunden. Optionen gibt es keine.

1.1 Befehle

`randbild` Dieses Package stellt die Umgebung `randbild` zur Verfügung. Damit kann man kleine veranschauende Bilder auf dem Rand der Seite unterbringen. Verwendung:

```
\begin{randbild}[\langle Titel \rangle]{\langle x_1, y_1 \rangle}{\langle x_2, y_2 \rangle}
  \langle inhalt \rangle
\end{randbild}
```

Dabei geben $\langle Titel \rangle$ eine Beschriftung für das Bild (kann weggelassen werden), $\langle x_1, y_1 \rangle$ die Koordinaten der linken unteren Ecke, $\langle x_2, y_2 \rangle$ die Koordinaten der

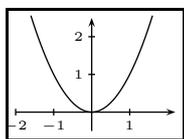
*Dieses Dokument gehört zu `randbild v0.2`, vom 2007/05/15.

[†]Paul-Ebermann@gmx.de

rechten oberen Ecke (jeweils in Bildkoordinaten) an. Man sollte dabei auf die begrenzte Breite des Seiten-Randes achten.

Das Bild erhält ein (beschriftetes) Achsenkreuz – weitere graphische Elemente (also das, was man eigentlich zeichnen will) kommen dorthin, wo *inhalt* steht. Dabei können alle pstricks-Makros, wie etwa `\psplot` oder `\psline` verwendet werden. Siehe dazu die pstricks-Dokumentation.

Beispiel:



Normalparabel

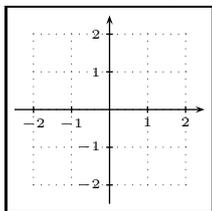
```
\begin{randbild}[Normalparabel]{-2,-.5}{2.2,2.5}
  \psplot{-1.6}{1.6}{ x x mul }
\end{randbild}
```

`randbildbasis`
`randbild*`

Die Umgebung `randbildbasis` funktioniert genauso wie `randbild`, nur wird kein Achsenkreuz gezeichnet. Ein alternativer Name dafür ist `randbild*` (weil ich jetzt schon mindestens zum fünften Mal diesen Namen verwendet habe, als ich `randbildbasis` haben wollte).

`\showgrid`

Mit dem Makro `\showgrid` kann man seinem Bild ein Koordinaten-Gitter hinzufügen. Es ist also innerhalb der Umgebungen zu verwenden.



Beispiel mit Gitter

1.2 Bildkoordinaten

Jedes Bild erhält sein eigenes Koordinatensystem. Die Einheiten sind jeweils halbe Zentimeter. Die Lage des Ursprungs wird implizit durch die Angabe der Koordinaten der Bild-Ecken gegeben.

2 Implementation

Mit `pst-plot` werden die Diagramme gezeichnet. `pst-plot` lädt auch gleich das Basis-Package `pstricks`.

```
1 %<*package>
2 \RequirePackage{pst-plot}
```

`\randbild@box` In `\randbild@box` speichern wir später die Box, welche die Grafik enthält. In `\randbild@titel` wird der eventuelle Titel des Randbildes gespeichert.

```
3 \newsavebox{\randbild@box}
4 \newcommand{\randbild@titel}{bla}
```

`randbild` Die wichtigste (und namensgebende) Umgebung dieses Paketes.

```
5 \newenvironment*{randbild}[3][\ ]{%
```

Mit `\@bsphack` (und dem `\@Esphack` am Ende) sorgen wir dafür, dass die Umgebung an der Stelle, wo sie verwendet wird, keine Spuren in der Ausgabe hinterlässt – auch keine doppelten Leerzeichen o.ä.

```
6 \@bsphack%
```

Wir rufen hier einfach die Umgebung `randbildbasis` mit unseren drei Parametern auf.

```
7 \begin{randbildbasis}[#1]{#2}{#3}%
```

Da drin malen wir jetzt das Koordinatensystem. (`\tiny` bezieht sich auf die Größe der Beschriftungen der Achsen.)

```

8      {\tiny%
9      \psaxes[ticks=1pt,labelsep=2pt]{->}(0,0)(#2)(#3){}%
10     }%
11 }{%

```

Am Ende dieser Umgebung beenden wir natürlich auch `randbildbasis`.

```

12 \end{randbildbasis}%
13 \@Esphack%
14 }

```

`randbildbasis` `randbildbasis` macht die eigentliche Arbeit.

```

15 \newenvironment*{randbildbasis}[3][\ ]{%

```

Da wir am Ende der Umgebung leider nicht mehr auf die Parameter zugreifen können, müssen wir uns den Titel hier merken.

```

16 \@bsphack%
17 \renewcommand{\randbild@titel}{#1}%

```

Wir setzen zunächst die Maßeinheit (`unit`) und die Linienbreite für unsere Zeichnung auf passende Werte.

```

18 \psset{linewidth=.5pt,unit=0.5cm}%

```

Da es `\fbox` und `\marginpar` leider nicht in Umgebungsform gibt, müssen wir zunächst (mit `lrbox`) eine Box erstellen und als `\randbild@box` speichern, um sie nachher mit einem Rahmen versehen und in den Rand stellen zu können.

```

19 \begin{lrbox}{\randbild@box}%

```

Hier beginnen wir eine `pspicture`-Umgebung passener Größe. Der Inhalt der `randbildbasis`-Umgebung wird dann Inhalt der `pspicture`-Umgebung, hier kann man also nach Herzenslust `pstricks`-Befehle nutzen.

```

20 \begin{pspicture}(#2)(#3)%

```

Ich habe noch nicht ganz verstanden, warum das folgende `\ignorespaces` notwendig ist, wo doch eigentlich schon `pspicture` dafür sorgt, dass Leerzeichen (und Zeilenenden) ignoriert werden – aber leider nicht in Zusammenarbeit mit `\multips`. Jedenfalls habe ich danach etwa zwei Stunden gesucht, bei einem Beispiel wie im Rand¹. Dieses `\ignorespaces` behebt das Problem.

```

21 \ignorespaces
22 }{%

```

Am Ende der `randbildbasis`-Umgebung beenden wir zunächst die `pspicture`-Umgebung und dann die `lrbox`-Umgebung (womit das alles nicht gezeichnet, sondern in der Box `\randbild@box` gespeichert wird).

```

23 \end{pspicture}%
24 \end{lrbox}%

```

Jetzt kommt die eigentliche Ausgabe. Wir beginnen einen „Rand-Absatz“ ...

```

25 \marginpar{%

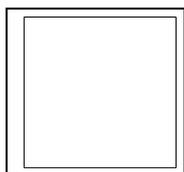
```

..., in welchem alles (horizontal) zentriert ist.

```

26 \centering%

```



Beispiel

¹Für die Darstellung des Beispiels hier (nach Behebung des Bugs) habe ich ein explizites Leerzeichen (`\`) verwendet, im Original war das nur ein ungeschützter Zeilenumbruch.

Dies betrifft zunächst das Bild (`\usebox{\randbild@box}`), welches am Rand rundum abgeschnitten wird (0.1 cm Abstand), und einen Rahmen bekommt.

```
27     \fbox{%
28         \clipbox[0.2]{%
29             \usebox{\randbild@box}%
30         }%
31     }%
```

Auf einer weiteren Zeile (mit wenig Abstand) ...

```
32     \[0.2\psunit]%
```

... folgt nun (in kleiner Schrift) der Titel des Bildes (der ja am Anfang als Parameter gesichert wurde).

```
33     {\small\randbild@titel}%
34 }%
35 \@Esphack%
36 }%
```

randbild* Jetzt definieren wir uns noch einen Alias für `randbildbasis`:

```
37 \expandafter\let\csname randbild*\endcsname\randbildbasis
38 \expandafter\let\csname endrandbild*\endcsname\endrandbildbasis
```

\showgrid Unser Koordinatengitter ist eine Spezialisierung von `psgrid`. Mit `\newpsobject` legen wir die Parameter fest.

```
39 \newpsobject{showgrid}{psgrid}%
40     {%
```

Wir wollen eine keine weitere Unterteilung unseres Gitters mit Gitterweite 1. Statt Linien malen wir je Einheit fünf Punkte.

```
41     subgriddiv=1,%
42     griddots=5,%
```

Die Dicke der (gepunkteten) Linien definieren wir hier als 0.4pt. Dagegen setzen wir die Größe der Beschriftung auf 0 (d.h. unsichtbar), da ja schon das Achsenkreuz eine Beschriftung darstellt.

```
43     gridwidth=0.4pt,%
44     gridlabels=0pt%
45 }
```

Ende.

```
46 \endinput
47 </package>
```

3 Liste der Änderungen

v0.0		v0.2
Allgemein: Anfang der Fassung als DTX	1	Allgemein: Verwendung von <code>\@bsphack</code> und <code>\@Esphack</code> (bei <code>randbild</code> und <code>randbildbasis</code>), um die Umgebungen an der Verwendungsstelle unbemerkbar zu machen. Außerdem wurden einige überflüssige Leerzeichen
v0.1		
Allgemein: erste veröffentlichte Version	1	
<code>randbildbasis</code> : Neu (als Abspaltung von <code>randbild</code>)	3	

