

# TU/ePDFScreen

## Presentations, Slides and Posters

Marko Boon

18th June 2002

# 1. Introduction

TU/ePDFScreen is a  $\text{\LaTeX}$  package that you can use to create presentations, without having to use Powerpoint. TU/ePDFScreen is based on PDFScreen, which uses modern techniques to create interactive PDF documents, suitable for full screen viewing. The TU/ePDFScreen package is also suitable for making posters or transparent slides (see the `aa` option). The package is written specifically for PDF $\text{\LaTeX}$ , so running the  $\text{\LaTeX}$  command usually will not give the desired result.

The main advantage of TU/ePDFScreen is that you can use normal  $\text{\LaTeX}$  commands. You do not need to learn a lot of new commands. The source code of this document can be found in the directory `C:\MiKTeX\localtexmf\examples\presentation\slides.tex` (where C: should be replaced by your MiKTeX installation drive).

## 2. Adobe Acrobat Shortcut Keys

It is quite useful to know (some of) these shortcut keys in Adobe Acrobat Reader:

|           |   |
|-----------|---|
| PAGE UP   | go back to the previous slide                           |
| PAGE DOWN | advance to the next slide                               |
| HOME      | go to the first slide                                   |
| END       | go to the last slide                                    |
| CTRL+L    | toggle between Full Screen mode and normal viewing mode |
| CTRL+N    | prompts for a page number and opens that page           |
| CTRL+P    | brings up the print dialog                              |
| F8, F9    | toggle visibility menu bar and tool bar                 |

### 3. Syntax of the $\text{\LaTeX}$ source

#### The header

The first couple of lines do not differ a lot from other  $\text{\LaTeX}$  documents:

```
\documentclass{report}  
\RequirePackage[T1]{fontenc}
```

Please note that these two lines are required. Do not use another document class than `report`. The package `fontenc` is required to enable the TU/e fonts. The next line will load the package `tuepdfscreen`. This package has a lot of options. The next pages will enumerate the most frequently used options.

## Options of the tuepdfscreen package

- screen** Makes the presentation suitable for PDF<sub>TEX</sub> and screen dimensions. Most users will **always** use this option, but you can also specify the **print** option which ignores slides and makes the presentation suitable for paper.
- panelleft** Shows a navigation panel on the left side of the screen. The **screen** option is required if you want to use this option.
- panelright** Same as **panelleft**, but shows the navigation buttons on the right side of the screen.
- handouts** Omits all pages that are placed within a **screen** environment. This is mostly used to print handouts of your slides, where you want to leave out some pages. There is a section on **printing handouts**.

- a4 Sets the screen size to A4 portrait paper. This makes your presentation no longer suitable for the screen. Use this option if you want sheets rather than a screen viewable presentation. Since PDF is fully scalable, you can use this option not only for transparent sheets, but also for A0 posters. We also provided the sources of a [poster example](#) (poster.tex).

Here is an example of the usage of the tuepdfscreen package:

```
\usepackage[screen,panelleft,paneltoc]{tuepdfscreen}
```

But usually you will probably use the following line:

```
\usepackage[screen]{tuepdfscreen}
```

## New commands in tuepdfscreen

The tuepdfscreen package defines some new commands:

```
\overlay{imagefile}
```

You can specify a background image using this command. If you do not specify a background image, a TU/e background image will be chosen. In the MiKTeX installation a couple of default background images are provided: (tuebg), tuebgk, tuebgkuk, tuebgkukbw, tuebgkbw, tuebgkukbw, tuebga4 and tuebgka4 (k = dept. of Mathematics and Computer Science, uk = English, bw = black/white, a4 = portrait). If you are a member of another department and want a background for your department, just send an e-mail to [marko@win.tue.nl](mailto:marko@win.tue.nl). In general any PowerPoint background can be converted to PDF, so you can also e-mail any PowerPoint file.

If you do not want a background image, you can specify: `\overlay{ }`.

## `\pagenumbering`

If you use this command, page numbers will be shown in the right bottom corner of your presentation. Be careful with this if you use the **handouts** option. If some slides will be omitted because they are placed in a **screen** environment, the page numbering will be different! Using the command `\addtocounter{page}{-1}` you can subtract 1 from the current page number. You should use this if you construct a page in several stages. Just look at the section about **images** to see how it can be done.

## `\bottombuttons`

Putting this command before the `\begin{document}` statement, will add small navigation buttons in the left bottom corner of the slides. It has the same functionality as the **navigation panel**, but does not consume as much space.



```
\screensize{height}{width}
```

This command changes the screen size. The default setting is 6.25 x 8 inch. If you decide to change these settings, make sure to check the background image, which will be out of proportion.

```
\margins{left}{right}{top}{bottom}
```

This changes the space between the text and the border of the slide. The default values are 0.5, 0.5, 1.0 and 1.0 inch. Please note:

- You need to call the `\screensize` command before the new margins will be applied!
- This only works before the `\begin{document}` statement.

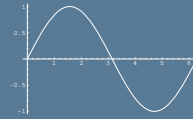
```
\urlid{url}
```

Sets the target of the "Home page" button in the **navigation panel**. The default value is [www.win.tue.nl](http://www.win.tue.nl).

```
\tuebox{text}
```

Creates a coloured box with TU/e gray background. Besides text, you can put any  $\text{\LaTeX}$  code in it:

You can even put an image in a tuebox:



## Other Settings

The file `localtexmf/tex/latex/pdfscreen/tuepdfscreen.cfg` contains all kind of settings, like margins and colours. You can create a copy of this file in your working directory and modify this copy.

## 4. Page Transitions

You can choose other page transitions than the one you saw until now. The command to do this, is:

```
\pagedissolve{<transition> <options>}
```

The options depend on the transition. However, one option that you can always use is `/D n`. This sets the number of seconds that the transition lasts. The default page transition is creating using this command:

```
\pagedissolve{R /D 0}
```

The letter R stands for Replace, and `/D 0` for a delay of 0 seconds. Before you use the command `\pagedissolve`, you should use a `\newpage` command before starting the **slide** or **slidetop** environment.

Page transitions do not work really well in Linux/Unix (tested in Acrobat Reader 4 and 5).

`\pagedissolve{Split /Dm /H /M /O}`

Two lines wipe over the screen and reveal the new page. These lines can move in a horizontal and vertical direction. Furthermore, they can move from the center towards the border (out), or the other way around (in).

## Options

`/Dm` (*dimension*) Determines whether the lines are horizontal or vertical. Possible values are `/H` (horizontal) and `/V` (vertical).

`/M` (*motion*) Determines whether the lines move inward or outward. Possible values are `/I` (in) or `/O` (out).

```
\pagedissolve{Blinds /Dm /V}
```

Multiple lines wipe over the screen and reveal the new page. These lines can be horizontal or vertical.

## Options

*/Dm (dimension)* Determines whether the lines are horizontal or vertical. Possible values are */H* (horizontal) and */V* (vertical).

`\pagedissolve{Box /M /O}`

A rectangle reveals the new page by moving from the center outward, or from the border inward.

## Options

`/M` (*motion*) Determines whether the rectangle moves inward or outward. Possible values are `/I` (in) or `/O` (out).

```
\pagedissolve{Wipe /Di 270}
```

One line moves over the screen and reveals the next page. The direction of this line can be set using degrees. 0 degrees stands for "from left to right". Adding multiples of 90 degrees will change the direction counterclockwise.

## Options

*/Di (direction)* Specifies the direction of the line. Possible values are:

- 0 (from left to right)
- 90 (from top to bottom)
- 180 (from right to left)
- 270 (from bottom to top)

`\pagedissolve{Dissolve}`

Dissolves the old page, so the new page will become visible.

## Options

No options available.



```
\pagedissolve{Glitter /Di 0}
```

This is a combination of Dissolve and Wipe. The old page dissolves, but the effect moves in the specified direction.

## Options

*/Di (direction)* Specifies the direction of the dissolve effect. Possible values are:

- o (from left to right)
- 90 (from top to bottom)
- 180 (from right to left)
- 270 (from bottom to top)

## 5. Images

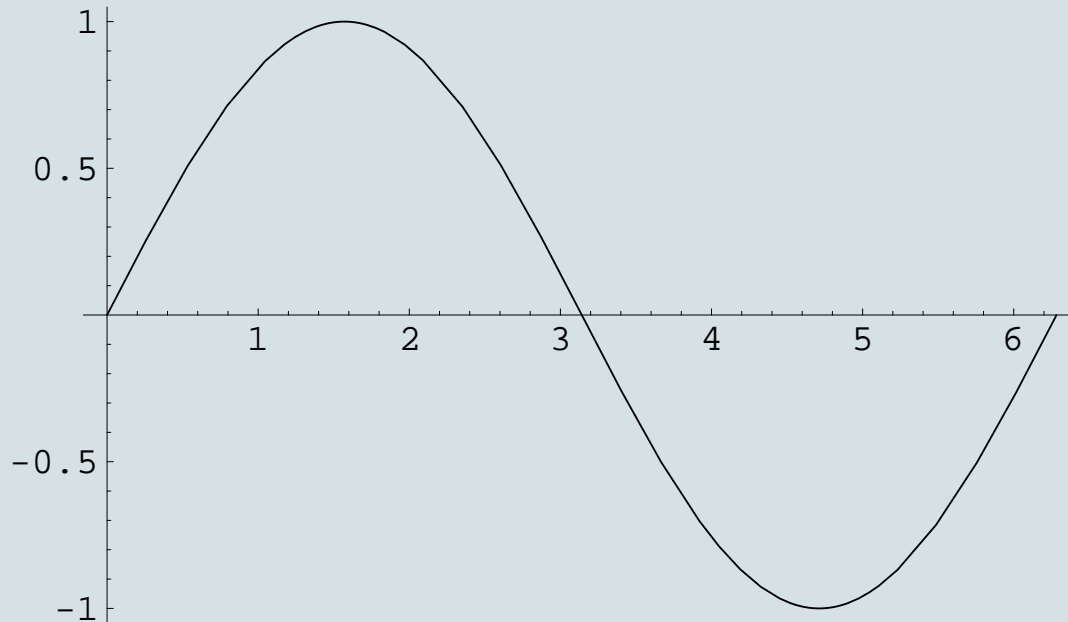
To include images in your presentation, you should use the command `\includegraphics`. This command is defined in the package `graphicx`. The usage is:

```
\includegraphics[width=n]{imagefile}
```

The width can be specified in units, like cm, mm or inch. But you can also specify any other  $\text{\LaTeX}$  length, e.g. `0.8\linewidth`. Imagefile is the file name of the image **without extension!** Usually you start by generating an Encapsulated PostScript (EPS) file. You can convert this file to PDF using `epstopdf` (TU/e  $\text{\LaTeX}$  users can drag it on a shortcut called EPS2PDF on their desktop). By leaving out the file extension,  $\text{\LaTeX}$  will choose the PDF version, and  $\text{\LaTeX}$  will choose the EPS version automatically.

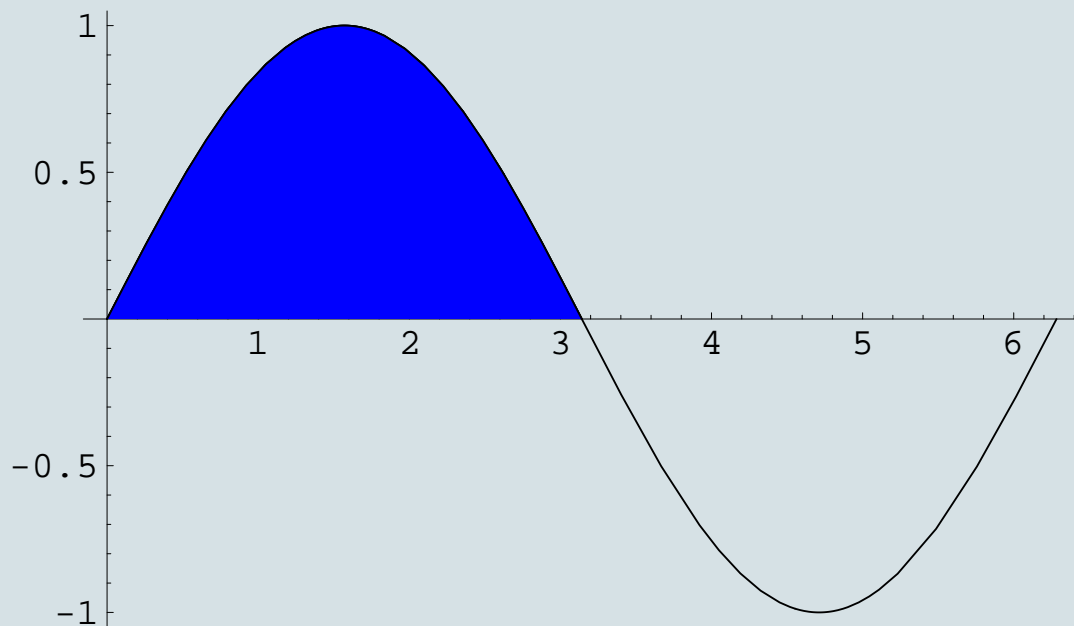
# Image example

$$\int_0^{2\pi} \sin x \, dx$$



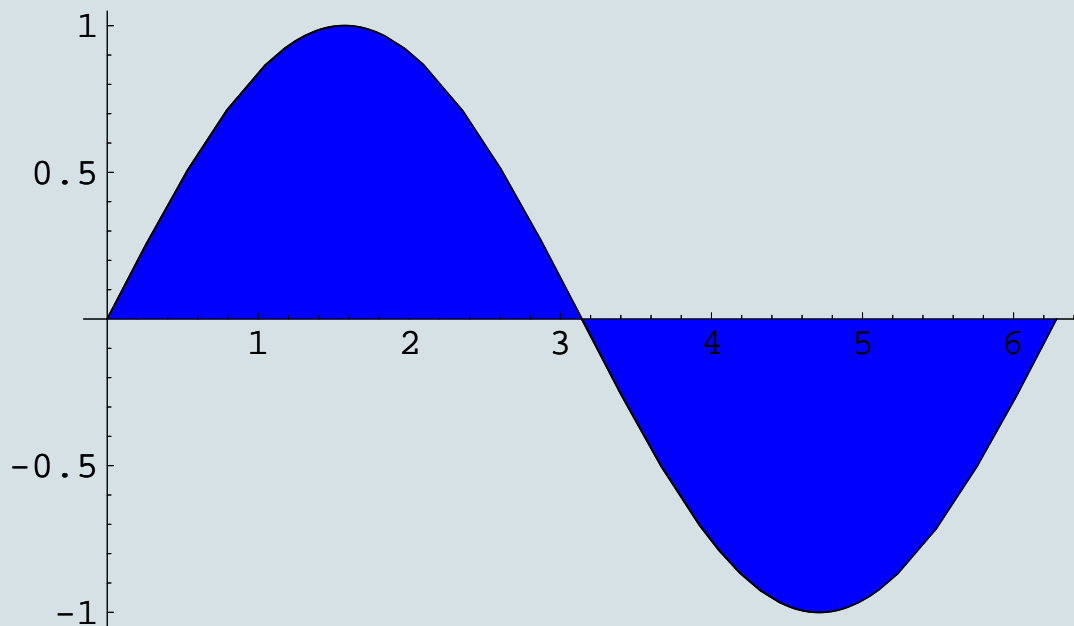
# Image example

$$\int_0^{2\pi} \sin x \, dx = 2$$



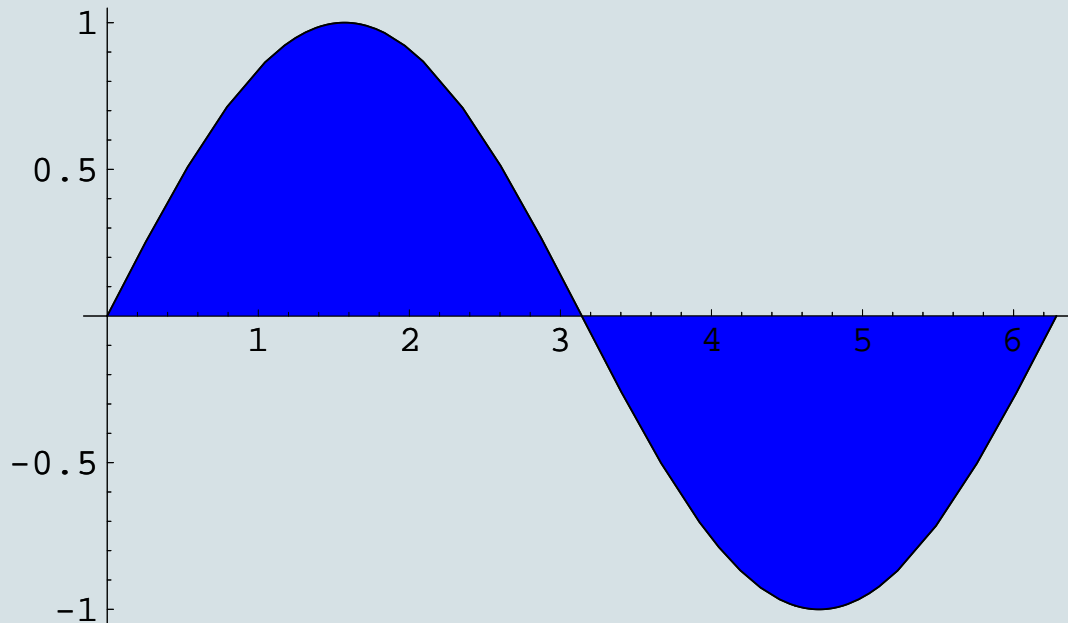
# Image example

$$\int_0^{2\pi} \sin x \, dx = 2 - 2$$



# Image example

$$\int_0^{2\pi} \sin x \, dx = 2 - 2 = 0$$



## 6. Captions and References

Because the `slide` and `slidetop` environments are implemented as mini-pages, the `figure` and `table` environments cannot be used. Most of the people do not regard this as a problem, but they regret the fact that the `caption` command can no longer be used. This means that you cannot refer to an image or a table. We solved this problem in `TU/ePDFScreen` by defining two new commands: `\tablecaption` and `\figcaption`. These commands should be used instead of the `\caption` command:

```
\includegraphics[width=0.5\linewidth]{plot1}  
\figcaption{a sine function}  
\label{fig:sine}
```

You can still use the command `\ref` to create a reference to this image.

On this page you will see an example of the usage of `\figcaption` and `\tablecaption`. Look at the source code for more details.

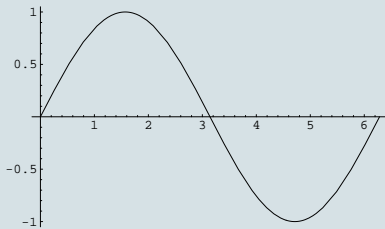


Figure 1: Een sine function

| A Table |       |
|---------|-------|
| Mick    | Keith |
| Charlie | Ron   |

Table 1: Some random names.

For the figure we used the `\figcaption` command, for the table we used the `\tablecaption` command. Creating a reference to figure 1 or table 1 can be done using `\ref`.

In the source code you might see the usage of a `tabularx` environment. We used this only to be able to put the figure and the table next to each other. It has nothing to do with the caption commands.



## 7. Printing Handouts

Since TU/ePDFScreen is written specifically for PDF<sub>La</sub>T<sub>E</sub>X, you have to use Adobe Acrobat (Reader) for both viewing and printing your slides. For printing you can use the keyboard shortcut CTRL+P. If you want to print the slides on A4 paper, you do not have to change any print settings. Make sure that you have at least Adobe Acrobat version 5, because in version 4 some fonts are printed incorrectly.

If have pages in your presentation that you don't want to print, you should put them in a **screen** environment:

```
\begin{screen} ... some slides ... \end{screen}
```

This means that the page numbers on the printed slides will be different from the page numbers of the presentation. To solve this problem, read the section on **pagenumbering**.

## Multiple slides on one page

Modern printer drivers (HP printers in Windows 2000 or XP) support printing multiple sheets on one page. Just click the Properties button in the Print dialog. If your printer driver does not support multiple sheets on one page, you can always generate pure PostScript. To generate PostScript from any application you can install the QMS Colorscript 1000 printer. The drivers are available on any Windows CD-ROM. Do not specify any port, but choose "Print to file (FILE:)". This will make a dialog pop up every time you print a file to this printer. Save this file on your local disk. Then use PrintFile, which comes with your TU/e MiKTeX distribution, to print this file. First select the file, then press **Settings** → **PostScript** → **Enable n-up printing**. Then, also in the Settings dialog, press the N-up button. This will bring up the dialog shown in figure 2.

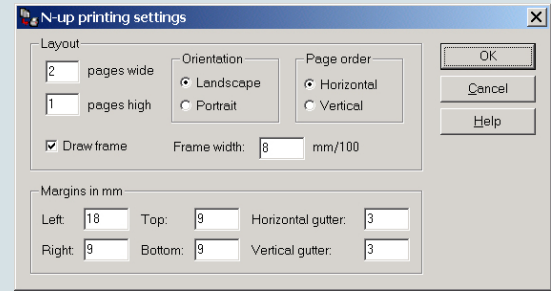


Figure 2: N-up in PrintFile

## 8. Revealing slides in multiple steps

- PowerPoint users might miss the animation feature that makes objects move into the slide. This is not possible in TU/ePDFScreen.

## 8. Revealing slides in multiple steps

- PowerPoint users might miss the animation feature that makes objects move into the slide. This is not possible in TU/ePDFScreen.
- However, as you can see, you can simulate this kind behaviour. There are two ways to do this.

## 8. Revealing slides in multiple steps

- PowerPoint users might miss the animation feature that makes objects move into the slide. This is not possible in TU/ePDFScreen.
- However, as you can see, you can simulate this kind behaviour. There are two ways to do this.
- The first way is just to create the complete page (use **slidetop** environment). When the page is finished, make as many copies of this page as necessary, and remove one more statement each time. Put all pages except the last, complete page in a **screen** environment to prevent them from being printed in **handouts** mode. Please note that page numbering and chapter numbering should manually be controlled to prevent these number from being increased. See **pagenumbering** for more information.

- the second way, which is used here, is easier to use. Just put a `\pause` command whenever you want a break. After running PDF $\TeX$  you still will not notice anything in the resulting PDF file. You have to run a postprocessor called **AddPause**. This program will bring up an Open File dialog so you can select your PDF file. AddPause will add the breaks and generate another PDF file without the break effects (you can use this for handouts).

Please note that AddPause uses PPower4, a Java program written by Klaus Guntermann. Therefore you need Java to run it. This means that you have to install it manually on systems without Java (like Windows XP).

You can always contact:

|  |  |
|--|--|
| Marko Boon   | Wil Kortsmit   |
| <a href="mailto:marko@win.tue.nl">marko@win.tue.nl</a> | <a href="mailto:rcwil@win.tue.nl">rcwil@win.tue.nl</a> |
| HG 9.09  | HG 9.83  |
| 2989   | 4587   |

- for questions concerning TU/ePDFScreen
- for questions regarding MiKTeX or  $\text{\LaTeX}$
- to get a free copy of the latest TU/e MiKTeX CD-ROM

