The **fancytooltips** package, the **fancy-preview** script[*][†]

Robert Mařík

`marik@mendelu.cz`

June 3, 2012

Contents

1 Introduction

The LaTeX package `fancytooltips` allows to insert tooltips into PDF documents – a popup windows appears if the mouse hovers or clicks particular area. You can use plain text as well as mathematics, pictures and animations in your tooltips. The PDF viewer must interpret Javascripts to make the tooltips work. This is currently true only for (free) Abobe Reader and (commercial) Adobe Acrobat. There are two ways how to produce the PDF file. The simplest way is to use `pdflatex`. Another option is `latex` + `dvips` + AdobeDistiller[1] + Adobe Acrobat[2].

Here you can test two tooltips: Einstein's formula and simple animation (numbers from 1 to 6). For more examples how the presentation may look like see the

---

[1]not free `ps2pdf`
[2]not free Adobe Reader.

examples subdirectory. We also provide a Perl script `fancy-preview` which allows to extract equations, theorems and related parts of text into separate PDF file and insert tooltips after each `\ref`, `\eqref` and `\cite` command automatically. For more information see Figure 1 and demos (scientific papers, presentations, tests) at `http://user.mendelu.cz/marik/fancy-preview`.

Related LaTeX packages are `cooltooltips`, `pdfcomment` (smaller size of the resulting PDF, but restricted to plain text), `ocgtools` (works with layers, only `pdflatex` is supported) and AcroTeX (works with layers, only `latex` + `dvips` + Adobe Distiller + Adobe Acrobat are supported). All these "similar" packages allow to write the text for the tooltips inside the main document. In contrast to this approach, `fancytooltips` package makes use of an external file. This allow to use graphics or mathematical expressions in the tooltips.

The package requires `eforms.sty` and `insdljs.sty` packages, which are part of AcroTeX bundle (`http://www.math.uakron.edu/~dpstory/webeq.html`).

2 Usage of `fancytooltips.sty`

2.1 Small technical background

- The pages from the external PDF file with tootlips are inserted as icons at hidden buttons into the resulting PDF.

- If you insert tooltip, the current page is covered by an invisible transparent button which spans across the whole page (the information about the pages with tooltips is stored in `aux` file and hence we have to run LaTeX *three times*). The button has an associated JavaScript action to close all tooltips (i.e. make itself and other related buttons hidden).

- Another button which reveals tooltip is created. This button is transparent, covers the text typeset by TeX and allows to activate a JavaScript. This JavaScript takes the picture required, puts this picture as icon to the button which covers the corresponding page and makes this button visible.

- Each page has an associated action which closes all tooltips when the page is opened.

2.2 The file with tooltips

The file with tooltips is an ordinary PDF file, one tooltip per page. Tooltips should be in the top right corner at the page, in a *box with nontransparent background* and the rest of the page should be transparent. If you consider to use `movetips` option which causes the tooltip appear near the mouse cursor instead of in the top right corner (see below), then every page should have the dimensions equal to the dimensions of the colored box with tooltip[3]. You can call the tooltips by their page num-

---

[3]Look at the files `tooltipy.tex` and `tooltipy.pdf` from `examples` subdirectory for a simple example how to meet this condition under pdfLaTeX. You may also use ordinary LaTeX class, set `\pagestyle{empty}` and crop the boundary.

bers, but for better comfort we also provide simple cross referencing mechanism.

\keytip   If the pdf file is created by LaTeX, you can define keywords to refer to the pages using `\keytip` command. Simply put `\usepackage[createtips]{fancytooltips}` into preamble and write `\keytip{⟨foo⟩}` in document. This writes information about keyword ⟨foo⟩ and the pagenumber into file `\jobname.tips`. See `http://user.mendelu.cz/marik/fancytooltips` for some tooltip templates.

2.3   The file with presentation

In the file with presentation, the user is responsible to

- input either `color` or `xcolor` package in the preamble

- LaTeX the file two times (we write some macros into `aux` file) or three times (if option `fg` is used).

You may use the following options to set the necessary information and change the default behavior of the package.

filename To input the tooltips from file ⟨foo.pdf⟩ call the package with `filename` option: `\usepackage[filename=foo]{fancytooltips}`. This option is required if compiled with `pdflatex`.

movetips By default, tooltip appears in the top right corner of the page (use View–PageLayout-Single Page in your Adobe Reader, please). If the option `movetips` is used, then tooltip appears close to the mouse pointer. More precisely, tooltip appears with left down corner at the mouse position, if there is enough place. If not, tooltip appears with right down corner at the mouse position. Finally, the tooltip is shifted down to fit the page, if necessary[4].

mouseover If you use `mouseover` option, then tooltip appears if you move the mouse pointer to the active area (no clicking is necessary).

inactive This option makes the package inactive.

active This option forces the package active even if `inactive` option is loaded.

blur The rest of the page is blurred, when showing tooltip. Use either `\usepackage[blur]{fancytooltips}` or `\usepackage[blur=number]{fancytooltips}`, where `number` is a number between 0 and 1. (Note that we use transparent package and hence this could have an influence on the colors of the document and could make your PDF less portable.) This option is allowed in pdfLaTeX mode only. If this options brakes colors only on the page which include tooltips, you may want to use option `fixcolor` to fix it.

---

[4]This option works in this way if every page of the file with tooltips has dimensions of the box with tooltip. See the `examples` subdirectory.

fixcolor See `blur` option.

debug Prints two alerts reporting success or problems when opening PDF file in Adobe Acrobat (Adobe Reader). Use this option to find possible source of problems. See also the Section 4 in this manual.

noextratext No mark is appended at the end of the link which opens tooltips (see help for `\tooltip`).

nosoap A single space is used to occupy the space for button produced by `\tooltip*` command instead of blue soap. As a result, the button produced by `\tooltip*` works as usual, but it is invisible. This is sometimes convenient for the user, since the mark does not disturb the text. However, the author has to instruct the user, that the hidden buttons follow links from cross references.

fg The button for displaying pictures is placed on background and the buttons which activate tooltips are placed immediately in the text by default. With `fg` option all these buttons are placed into foreground after the page is completed. We use `\pdfsavepos` command and keep track of the position for buttons in aux file and hence this option works in pdfLaTeX only and the file needs more compilations. This option does not do anything in dvi mode now (despite the fact that pdfLaTeX provides the feature also for dvi mode), but this could be changed if someone requests this feature. Use this option for example, if you use frame around hyperlinks or buttons and form fields in your PDF (like tests produced by AcroTeX).

preview Redefines `\ref`, `\eqref` and `\cite` commands to work with tooltips extracted by `fancy-preview`, see Section 3 and demo files in `example/fancy-preview` directory. The tooltip is inserted only if the target of the reference is on different PDF page as the tooltip. This option is suitable for presentations where the whole PDF page is visible. Replaces former `\FancyHook` command.

previewall Like `preview` but shows tooltip also if the target is on the same page. Suitable for enhanced versions of papers written on A4 page.

tooltipmark Allows to change the tooltipmark to some predefined styles, values are 1 , 2 , 3 and 4 . Note that you can change the mark to whatever different by redefining `\TooltipExtratext` command. Work only in pdflatex mode.

`\tooltip`   The user can put the tooltip into her or his presentation using the command `\tooltip{⟨stuff⟩}{⟨keyword-or-pagenumber⟩}` where ⟨stuff⟩ is the printed text in ⟨tooltipcolor⟩ color and ⟨keyword-or-pagenumber⟩ is either the pagenumber of the tooltip in the external file or the keyword defined by `\keytip` command and stored in `\TooltipFilename.tips` file, where `\TooltipFilename`
`\TooltipExtratext`   is set automatically from the `filename` option. The printed text ⟨stuff⟩ is

followed by `\TooltipExtratext` command. The default value is small blue soap, as you have seen in the second paragraph of this documentation. There is a `noextratext` option which defines `\TooltipExtratext` to be empty. If {⟨*keyword-or-pagenumber*⟩} is not recognized as valid keyword for tooltips, it is supposed to be pagenumber.

The text {⟨*stuff*⟩} is inserted in `\hbox` by `\tooltip`. With starred version of the `\tooltip` macro the text {⟨*stuff*⟩} is not inserted into the box and the active button dos not cover the text {⟨*stuff*⟩}, but covers the mark produced by `\TooltipExtratext`.

The user can put a series (animation) of tooltips into the presentation by using `\tooltipanim{`⟨*stuff*⟩`}{`⟨*start*⟩`}{`⟨*end*⟩`}` command, where ⟨*start*⟩ and ⟨*end*⟩ are keywords defined by `\keytip` command or page numbers. The delay between two frames is `\delayinterval` milliseconds. The default value is 200, you can change it by command `\def\delayinterval{100}`. There is also starred version which works similarly like `\tooltip*` command.

Extra text added to `\ref`, `\eqref` and `\cite` commands with previews, see Section 3. The default value is the same as for `\TooltipExtratext`.

Margin notes: noextratext option; `\tooltip*`; `\tooltipanim`; `\tooltipanim*`; `\delayinterval`; `\TooltipRefmark`

2.3.1   Changes for dvips users

Dvips users also have to prepare tooltips into PDF file, not eps as usual. But we have to insert these tooltips in Adobe Acrobat Pro program. If you use Acrobat Pro version 8.1 and later, install the file `aeb.js` from AcroTeX eDucation bundle as described in the documentation to AcroTeX.[5]

Since LaTeX is not capable to find the number of pages in external PDF file with tooltips, dvips users have to specify option `dvips` in `fancytooltips` package. You have to use also a `pages` option with the number of pages in the PDF file with tooltips, if you use dvips route. You have to call the package by something like this:
`\usepackage[dvips,filename=tooltipy,pages=27]{fancytooltips}`
You have to `latex` (two times) and `dvips` your file first. This produces `filename.ps` and `Tooltipsdljs.fdf` files. Distill the `filename.ps` file into `filename.pdf` and open this file by Adobe Acrobat Professional – this imports macros from `Tooltipsdljs.fdf` file. In Acrobat's JavaScript console (open by `Crtl+J`) run (using `Ctrl+Enter`) the command `ImportTooltips();` which is defined for the document. This command inserts invisible buttons on the first page, imports icons (the file with icons specified as ⟨*filename*⟩ parameter when loading fancytooltips must be in working directory). You should see a message "`importing pictures`" and the command returns 1 when finished. Then **save the file** (you can use the same name). If the command `ImportTooltips();` fails, you either have not the PDF file with tooltips in current directory, or the PDF file does not contain JavaScripts. In the latter case insert document level JavaScripts manually as described in the Section 4.

Margin notes: dvips; pages

---

[5]If you do not install `aeb.js` properly, you can still create your presentation, but you have to import the `fdf` file manually, see the section Troubleshooting and known problems.
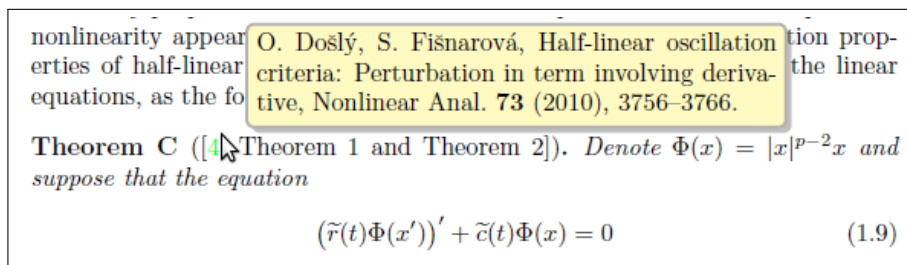
nonlinearity appear... tion properties of half-linear... the linear equations, as the fo...

O. Došlý, S. Fišnarová, Half-linear oscillation criteria: Perturbation in term involving derivative, Nonlinear Anal. **73** (2010), 3756–3766.

Theorem C ([4], Theorem 1 and Theorem 2]). *Denote* $\Phi(x) = |x|^{p-2}x$ *and suppose that the equation*

$$\left(\widetilde{r}(t)\Phi(x')\right)' + \widetilde{c}(t)\Phi(x) = 0 \tag{1.9}$$

Figure 1: `fancy-preview`

3 Tootlips at references: `fancy-preview`

3.1 Basic usage

There is a Perl script `fancy-preview` which can be used to extract text from bibliography items, numbered equations, numbered theorems, lemmas, etc, put this text into separate PDF file and add this text as tooltips to the corresponding `\ref`, `\eqref` and `\cite` commands. Reading the resulting file may look like on Figure 1.

The script `fancy-preview` has been tested with Texlive2011 on both Linux and MS Windows. To run this script you need working Perl installation (usually present on Linux workstations, on MS Windows you may need to install Perl from `http://www.activestate.com/activeperl`) and `Config::IniFiles` module[6])

To compile your document `file.tex` do the following

- Put `\usepackage{hyperref}` into the preamble of your document (if not already loaded).

- If you write references in `thebibliography` environment, put empty line after each `\bibitem` command (including the last item in `thebibliography`).

- Run `fancy-preview file`. After several compilations you should get the PDF file `file.pdf`.

The default work-flow is the following. The file is compiled with `pdflatex` to get correct numbers of equations and in the first pass of `preview.sty` we extract displayed equations (but the numbers are thrown away). After this we crop the PDF file by using `pdfcrop` program (an alternate program can be specified as optional parameter). Then we extract numbered environments (theorem, Theorem, lemma, corollary, definition, figure, table) using the second pass and crop the PDF file again. After this we merge all equations, theorems etc which are marked with `\label` command. The PDF file with extracted parts of the text is the used as source of toltips in final compilations.

---

[6]Package `libconfig-inifiles-perl` on Ubuntu Linux, `cpan Config::IniFiles` or `ppm install Config::Inifiles` on MS Windows. Alternatively you can run `ppm` without any parameters to invoke the GUI.

Many things can be customized. The following options are available.

pdfcrop You may specify an alternative batch file to crop boundary of PDF file. Default is `pdfcrop`. The command line for an alternative program to crop PDF file is supposed to be the following: `programname input.pdf output.pdf`. Using optimal program to PDF file may be much fasater and may produce significantly smaller files.

tooltips You may combine the tooltips extracted by `fancy-preview` with "ordinary hand made tooltips". Simply call `fancytooltips` in the main document by `\usepackage[inactive]{fancytooltips}` in your document and specify the file with tooltips in the command line of the `fancy-preview` or in the `ini` file. You may also compile your file by `pdflatex` and get "normal" PDF output (the compilation is way faster).

fancy_options Options passed `fancytooltips` in final compilations. Default is `previewall,nosoap`. Options `mouseover` and `movetips` are added automaticaly.

ini_file Specifices the `ini` file with configuration, see the next subsection.

3.2 Configuration from ini files

Other customization can be done via `ini` files. The script looks for customizations in the file specified by `ini_file` command line parameter. If this parameterer is not used, the script looks for customization in two default locations: `~/.fancy-preview.ini` and in `./fancy-preview.ini` (both files are used if both exist). You can use `~/.fancy-preview.ini` for customizations related to all your projects and `./fancy-preview.ini` for projects in the current directory. The options from the file `./fancy-preview.ini` override `~/.fancy-preview.ini` and the options from command line override options from `./fancy-preview.ini`. The format is described at `http://search.cpan.org/~shlomif/Config-IniFiles-2.75/lib/Config/IniFiles.pm`.

The parameters are divided into two sections, `[main]` and `[latex]`. In section `[main]` of `ini` file you can set parameters `pdfcrop`, `tooltips` and `fancy_options`.

In the section `[latex]` if the initialization file you can customize the compilation by LaTeX. Here you can set parameters `environments` and `snarfenvironments` to set the environments which will be extracted. The default values are `environments=Theorem,theorem,lemma,corollary,definition` and `snarfenvironments=figure`.

The material from `tex` file is extracted in three passes. These passes are denoted by `a`, `b` and `c`. If `\label{⟨foo⟩}` appears in the text which is marked for extraction, then the corresponding `\newlabel{⟨foo⟩}` command is written to the `aux` file and ⟨foo⟩ is supposed to be the name of the keyword corresponding to the PDF page with the text.

If a referenced material appears in more passes, then the priority is set in the variable `$latex{'pass_order'}` and can be customized in the ini file in `[latex]`

section as `pass_order` parameter. The default value is `pass_order=acb`, i.e. `c` overrides `b` and `a` overrides `c`.

As a typical example consider equation with label `\label{eq}` in numbered theorem with label `\label{th}`. The equation is extracted in pass `a` (displayed equation) and in pass `b` (the whole theorem). The corresponding `\newlabel{eq}` command appers in two `aux` files – from passes `a` and `b`. The first one corresponds to the PDF page with equation, the latter to the PDF page with whole theorem. Since `a` overrides `b`, then `\ref{eq}` and `\eqref{eq}` commands show the number of the equation followed by the tooltip with equation only. Further `\ref{th}` shows number of the theorem followed by the tooltip with the whole theorem. If you set `pass_order=ba`, then both `\ref{eq}` and `\ref{th}` are followed by the same tooltip.

The following options are available[7].

<dl>

a Defines commands for the first pass. It inserts `preview.sty` command which extracts displayed mathematics. Also resets `\tagform@` and `\@eqnnum` to skip printing of equation numbers.

a_extra Defines material which is appended to `a`

b Defines commands fot the second pass. In this LaTeX run are (by default) floating figures and theorem-like environments extracted. Inserts `preview.sty`. At the runtime, `\PreviewEnvironment[{[]}]{env}` and `\PreviewSnarfEnvironment[{[]}]{env}` for each `env` in comma separated list from `environments` and `snafenvironments` is added, respectively.

b_extra Defines material which is appended to `b`

environments See `b` option.

snarfenvironments See `b` option. Default value is `figure`.

c If empty (default value), then the third pass is skipped. Otherwise, you may activate `preview.sty` similarly like in `b` (for a template see the source code and the default setting of `$latex{'b'}`) and extract environments and commands according to your interest. A possible application is to extract `minipage` environments, if there are two or more figures inserted in `minipage` environments into one `figure` environment.

pass_order Sets priority, which pass is supposed to produce the output for a `\label` which is extracted more times than once, see the previous paragraphs for explanation and example.

preview_bibitem Redefines `\bibtem` command. The material between `\bibitem` and `\par` is wrapped to the width of `0.75\textwidth` and extracted.

preview_biblatex Similarly like `preview_bibitem` but works with `biblatex`.

</dl>

---

[7]These options are used as keywords in a hash variable `latex`, i.e. for default value of `param` parameter search the file `fancy-preview` for `$latex{'param'}`.

ini Inserted at the begin of each `pdflatex` run.

tooltips_envelope_preamble Used in preamble. Defines command `\tooltipwraper`. This command wraps the tooltips. Default is to use `tikz` to put everything into a yellow box with rounded corners and shading.

biblatex Creates temporary file `fancy-preview-biblatex-settings.tex`. This file contains definition which allow `biblatex` to work with citations and tooltips and we input this file in final compilations. This code comes from `tex.stackexchange.com`.

3.3 Tips and tricks

- The program `pdfcrop` from TEXlive may produces large PDF files. See the discussion at `http://tex.stackexchange.com`[8]. The smaller size can be obtained with the solution from the discussion below the question, which is based on `gs` and `pdftk`. The `python` script from the same discussion produces slightly larger file than `pdftk`, but still much smaller than `pdfcrop` and provides the fastest solution.

- Do not use floats in environments, which are extracted. Otherwise you get an error from LATEX. A workaround could be also to change temporarily definition of the floating environment (redefine `figure` environment, for example).

- If you are not interested in customization via `ini` files and do not want to install extra modules to your Perl installation, you may delete the about twenty lines from `fancy-preview` starting with `use Config::IniFiles;` up to the line `read_config("./fancy-preview.ini");`

4 Troubleshooting and known problems

The source code is in Mercurial repository at `http://bitbucket.org/robert.marik/fancytooltips/`. You can also report problems and issues in the forum at this site. The code on `bitbucket.org` is considered as development version and repository for older versions. The last stable version is always the version from CTAN.

- The package works with `eforms.sty` from version 2006/10/03 v1.0a. You can download this or newer version from `http://www.acrotex.net` site.

- If the graphics included by `\TooltipExtratext` and `\TooltipRefmark` has colors with **custom opacity**, Adobe Acrobat Pro sometimes renders the pictures bad. No problems of this type have been observed with free Adobe Reader.

---

[8]`http://tex.stackexchange.com/questions/42236/pdfcrop-generates-larger-file`

- For **dvips** users: In some cases the file `Tooltipsdljs.fdf` is **not imported automatically** (probably some setting in Adobe Acrobat or new versions of `eforms.sty` and `insdljs.sty`). In this case you do not see any message when using `debug` option. You have troubles of this type if you see in the Javascript console (Ctrl+J) error messages like "`aebTrustedFunctions is not defined 3:Page:Open CloseTooltips is not defined 1:Page:Open`". In this case you have to import the file `Tooltipsdljs.fdf` **manually from "Form" menu** in Adobe Acrobat Pro. Then go to the JavaScript console and run `ImportTooltips();` command.

Follow the points below if you want to find the source of your problems.

- For dvips users it is a good idea to check that AcroTEX is properly installed. Do the demo files from AcroTEX work for you?

- Try to use `debug` option, prepare the PDF file and open it in Adobe Acrobat or Adobe Reader.

  - You should see two messages when opening the file. If not, the Javascript do not work in your document (are not inserted properly).

  - Both messages should report success for pdflatex users. For dvips users one of the message should report error and if you run `ImportTooltips();` command in Javascript console, you should see a message in Javascript console which confirms that the pictures from external PDF file have been imported. If you save the PDF file and open again, both messages should report success.