

Colour slides with L^AT_EX and seminar.sty ¹

Michel Goossens

(CERN)

Sebastian Rahtz

(ArchaeoInformatica)

1 Slides and L^AT_EX

Many L^AT_EX users want to take advantage of T_EX's high-quality typesetting when they produce overhead slides for a presentation. This facility was originally provided by a separate package, SLI_TE_X, but that had a number of disadvantages:

- it was limited to a set of specially-scaled Computer Modern fonts and it was not easy to adapt to other fonts;
- the user was required to have two separate files, one for control information and the other for the actual slides;
- the control of colour and overlays was limited and crude;
- There was only one 'style' for slides, and writing a different layout (to, say, put a logo on each slide) was not documented.

L^AT_EX users now have a variety of fonts, and vast numbers of styles, to choose from, but SLI_TE_X has lagged behind. When L^AT_EX_{2 ϵ} was released at the end of 1993, this included a simple L^AT_EX package (already available in the New Font Selection Scheme, version 2) to emulate SLI_TE_X without the overhead of a separate macro package. However, there is a much better L^AT_EX package which has been available for some time now—**seminar.sty**; if used in conjunction with a POSTSCRIPT printer, and a set of useful macros called PStricks,² this offers almost every imaginable facility, including:

- ☞ Fancy frames, headers and footers;
- ☞ Landscape and portrait slides in the same document;
- ☞ Coloured text and tables;
- ☞ Interleaving of annotations and slides;
- ☞ Slide 'chapters' and list of slides;
- ☞ Overlays.

seminar is a normal L^AT_EX package which can be used with almost all other L^AT_EX packages (such as those to change font, include graphics etc). Its main job is to produce transparencies, but it can also make accompanying

notes from the same file. It is compatible with *AMS-L^AT_EX* and L^AT_EX_{2 ϵ} .

2 Using the seminar style

Usage is simple; begin your document in the normal way³ with

```
\documentclass{seminar}
```

or

```
\documentstyle[... options ...]{seminar}
```

The slide environments are

```
\begin{slide}
```

```
...
```

```
\end{slide}
```

```
\begin{slide*}
```

```
...
```

```
\end{slide*}
```

Where `slide` is for landscape slides and `slide*` is for portrait slides. By default, the document is typeset in *landscape* mode, but if you include the `portrait` style option, the document is typeset in portrait mode. *Typesetting* the document in landscape mode is different from *printing* it in landscape mode; you have to worry about the orientation of the page when printing, but with `dvips` this is simple, and taken care of in the local control file described below.

So the default output⁴ from this input:

```
\documentclass{seminar}
\usepackage{times}
\begin{document}
\begin{slide}
My talk is about:
\begin{description}
\item[Cats] Nice furry creatures
which should have a place in
every good home;
\item[Dogs] Nasty barking things
which bite you;
```

¹ Reprint from the Annals of the UK T_EX Users' Group: **Baskerville**, Volume 4.1, Januari 1994.

² The `seminar` package and PStricks are the work of Timothy van Zandt (tvz@princeton.edu).

³ We are assuming L^AT_EX_{2 ϵ} here, just to remind you to upgrade.

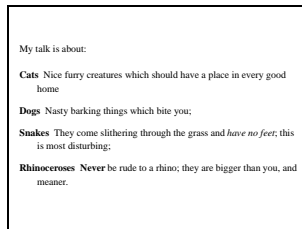
⁴ We have added a style option `output 'times'` so that the output will reduce properly to thumbnails for this article.

```

\item[Snakes] They come slithering
  through the grass and \emph{have
    no feet}; this is most
  disturbing;
\item[Rhinoceroses]
  \textbf{Never} be rude to a
  rhino; they are bigger than you,
  and meaner.
\end{description}
\end{slide}
\end{document}

```

will look like:



Most slides will be no more complicated than this, using standard \LaTeX environments like `itemize`, `enumerate` and `tabular`.

3 Frame styles

A variety of slide framing styles are available, set with the `\framestyle` command; the following are some of the predefined ones (some assume you have a POSTSCRIPT printer), using the `\slideframe` command:

none

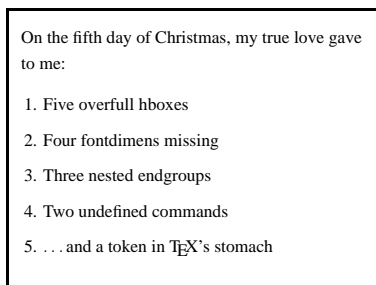
```

On the fifth day of Christmas, my true love gave
to me:

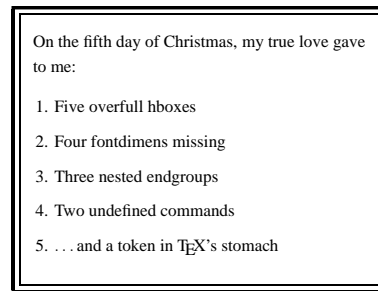
1. Five overfull hboxes
2. Four fontdimens missing
3. Three nested endgroups
4. Two undefined commands
5. ... and a token in  $\text{\TeX}$ 's stomach

```

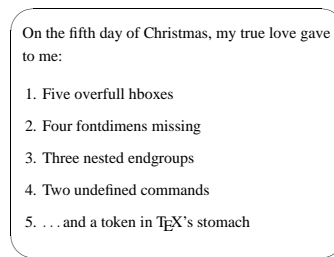
shadow



double

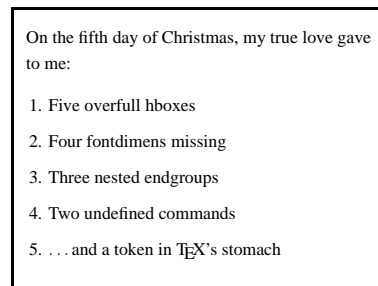


oval

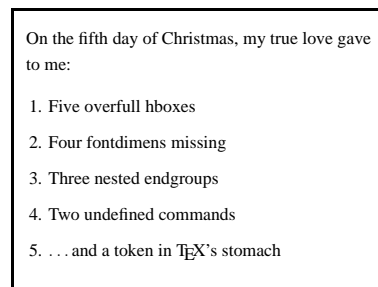


Similarly, a variety of page styles (the headers and footers) are available with the `\pagestyle` command, such as:

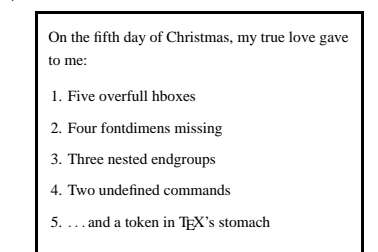
empty



plain



align



Both slide frames and page styles can be customized; for instance, the examples in this paper (eg Figure 1) are suitable for use at the European Particle Physics Laboratory (CERN).

4 Interleaving notes, and selecting subsets

It is easy to intersperse your slides with notes to yourself; these can be simply placed between the `slide` environments or enclosed in a specific `note` environment. You can use any L^AT_EX commands in these notes, and include your whole article here if desired. When you want to print the slides, a variety of style options can be used:

- slidesonly** Only the slides are printed;
- notesonly** Only the notes are printed;
- notes** The slides are interleaved with the notes;
- article** The document notes are typeset like a normal L^AT_EX paper, and the slides are placed as figures (reduced to half size).

The `\slideplacement` command can be used to affect how slides are placed in the `\article` format; the possible parameters are:

- float** (default) Slides are floated
- float*** Slides are floated, but if two column format is chosen they will span both columns
- here** Slides are placed where they occur in the notes

Further detailed control of the interaction between slides and notes is given in the *User's Manual*.

Selected slides can be included or excluded with the `\onlyslides` or `\noteslides` commands which a parameter of a comma-separated list of slides; this can be numbers, ranges (e.g., 5–10) or L^AT_EX `\ref` commands referring to `\label` commands in the slides.

5 Control over slide size, fonts and magnification

There are a great number of parameters by which the user can change any of the following either on a slide-by-slide basis, or for the whole document:

- Slide height and width;
- Top, bottom, left and right margins;
- Text justification (it is ragged right by default);
- Page breaking by varying tolerance of over-running material;
- Inter-line spacing;
- Point size, and choice of fonts.

How to change the default settings is explained in detail in the *User's Guide*.

Because `seminar` works by magnifying pages, sophisticated users should read the manual to see how to deal with setting and changing T_EX dimensions. Most users need not worry about this—in commands like `\epsfig` you should always express your ‘width’ and ‘height’ requests in fractions of the line size anyway.

6 Fonts

As the size of the text on your slides is larger than that for normal text (articles, manuals, etc.), often you will find that

some fonts (especially for mathematics) are not available at the size needed. If you use `dvips`, it tries to generate these fonts but does not always find the necessary Metafont input files to successfully complete the generation of the new font bitmaps. It makes sense, therefore, to use scaleable POSTSCRIPT fonts; the complete Computer Modern family is available in POSTSCRIPT Type1 format, either in the commercial set from Blue Sky or Y&Y, or in (slightly less polished) public domain Paradissa set by Basil Malyshev.

7 The local control file

The `seminar` package always starts by trying to find a file called `seminar.con` on the `TEXINPUTS` path; this gives the user or site an opportunity to conveniently overwrite the defaults. The `seminar.con` file can contain any L^AT_EX commands, including inputting style files. Figure 1 was typeset using a `seminar.con` set up for CERN; the contents of this are given below, with explanation of what is being done:

First, we set up landscape macros for the `dvips` driver.

```
\newcommand{\printlandscape}{%
  \special{papersize=297mm,210mm}}
```

We will assume POSTSCRIPT printers, and gain nice POSTSCRIPT effects like rounded box corners; these will need some extra style files:

```
\input semcolor.sty
\input fancybox.sty
```

Since this setup is designed to be used in Europe, where A4 paper is standard, we need to set up some sizes and re-invoke the code to set internal parameters:

```
\def\paperwidth{210mm}
\def\paperheight{297mm}
\input sem-page.sty
\slidewidth 205mm
\slideheight 150mm
\def\slideleftmargin{.8in}
\def\sliderightmargin{.8in}
```

For slide ‘sections’, list of contents, etc we use another style file:

```
\input slidesec.sty
```

This gives us access to various internal commands, some of which are used below. We can also produce a list of slides, in two layouts:

<code>\listofslides</code>	<code>\Slidecontents</code>
<p>List of Slides</p> <p>Introduction</p> <ul style="list-style-type: none"> 3 Slides and \LaTeX 4 Introducing 'seminar' 5 Normal slide with coloured background and text <p>Frame styles</p> <ul style="list-style-type: none"> 6 Frame styles <p>Text colours and colour tables</p> <ul style="list-style-type: none"> 8 Text colors 9 Colour tests 10 A multi-page coloured table 11 Z schemas built up with overlays 	<p>Introduction</p> <ul style="list-style-type: none"> ✓ Slides and \LaTeX 3 ✓ Introducing 'seminar' 4 ✓ Normal slide with coloured background and text 5 <p>Frame styles</p> <ul style="list-style-type: none"> ✓ Frame styles 6 <p>Text colours and colour tables</p> <ul style="list-style-type: none"> ⇒ Text colors 8 • Colour tests 9 • A multi-page coloured table 10 • Z schemas built up with overlays 11

For slide headings, there is a predefined `\slideheading` command; we will amend this so that it is typeset with a 'shadow'. A `slidechapter` command is also defined:

```
\newcount\@inchap
\@inchap0
\def\l@chapslide#1{%
  \slide@undottedcline{#1}{}%
}
\def\slidechapter#1{%
  \begin{slide}
  \immediate\addtocontents{los}{%
  \protect\l@chapslide{\ignorespaces%
  \protect\bfseries
  \protect\large #1}}%
  \gdef\thechapterheading{#1}%
  \end{slide}
  \@inchap0
}
```

This allows the user to break the slides into groups; the slide chapter title will be given in the bottom right corner with this CERN style.

```
\def\@empty{}
\renewcommand{\makeslideheading}[1]{%
  \gdef\theslideheading{#1}%
  \def\@tempa{#1}%
  \ifx\@tempa\@empty\else
    \begin{Sbox}
      \begin{Bcenter}
        \large\textbf{#1}%
      \end{Bcenter}
    \end{Sbox}
  \centerline{\shadowbox{\TheSbox}}
  \vspace{1ex minus 1ex}
  \fi
}
```

Now the CERN page and frame styles; the plain 'cern' style just places registration '+' marks, and the date:

```
\newpagestyle{cern}%
{ {\color{Black}\small
  \textbf{+} \hfil \today
  \hfil \textbf{+}} }%
{ {\color{Black}\small
  \textbf{+} \hfil \thepage
  \hfil \textbf{+}} }%
```

Whereas the 'cernsections' style has section headings and a logo:

```
\newpagestyle{cernsections}%
{ {\color{Black}\small
  \raisebox{- .5cm}{0cm}[0cm]{%
  \epsfig{figure=cernlogo.eps,%
  height=.8cm}}
  \hfil\textbf{\theslideheading}
  \hfil\textbf{\thepage}} }%
{ {\small\color{Black}\today
  \hfil\thechapterheading
  /\inchap
  } }%
```

For the slide frames, we define a frame with the word 'CERN' set in a coloured box on the lower left; this is done using the `PStricks` macros (which are automatically included by the 'semcolor' option above). The colour commands are also defined using `PStricks` (the code is not listed here); the names are used in the 'colour' figure.

```
\newslideframe{cern}{ {\SlideFront
  \boxput(-0.7,-1.11){%
  \psframebox[linecolor=black,%
  fillcolor=ForestGreen,%
  fillstyle=solid]{%
  \hbox{%
  {\normalsize
  \sfamily
  \color{Black}CERN}%
  }}{#1}%
  \color{Black}
  } }
```

Finally we make sure that each slide starts with the current foreground colour.

```
\def\everyslide{\SlideFront}
\def\theslideheading{}
```

The user sets up slides with the command `\SlideColours`, with two parameters, which are colour names for foreground and background. A synonym is defined for black on white. We have to be a bit careful defining the frame border, because by default it is coloured using the POSTSCRIPT 'setgray' operator, and that might not work with colour separation, so we define an explicit blue frame (for variety).

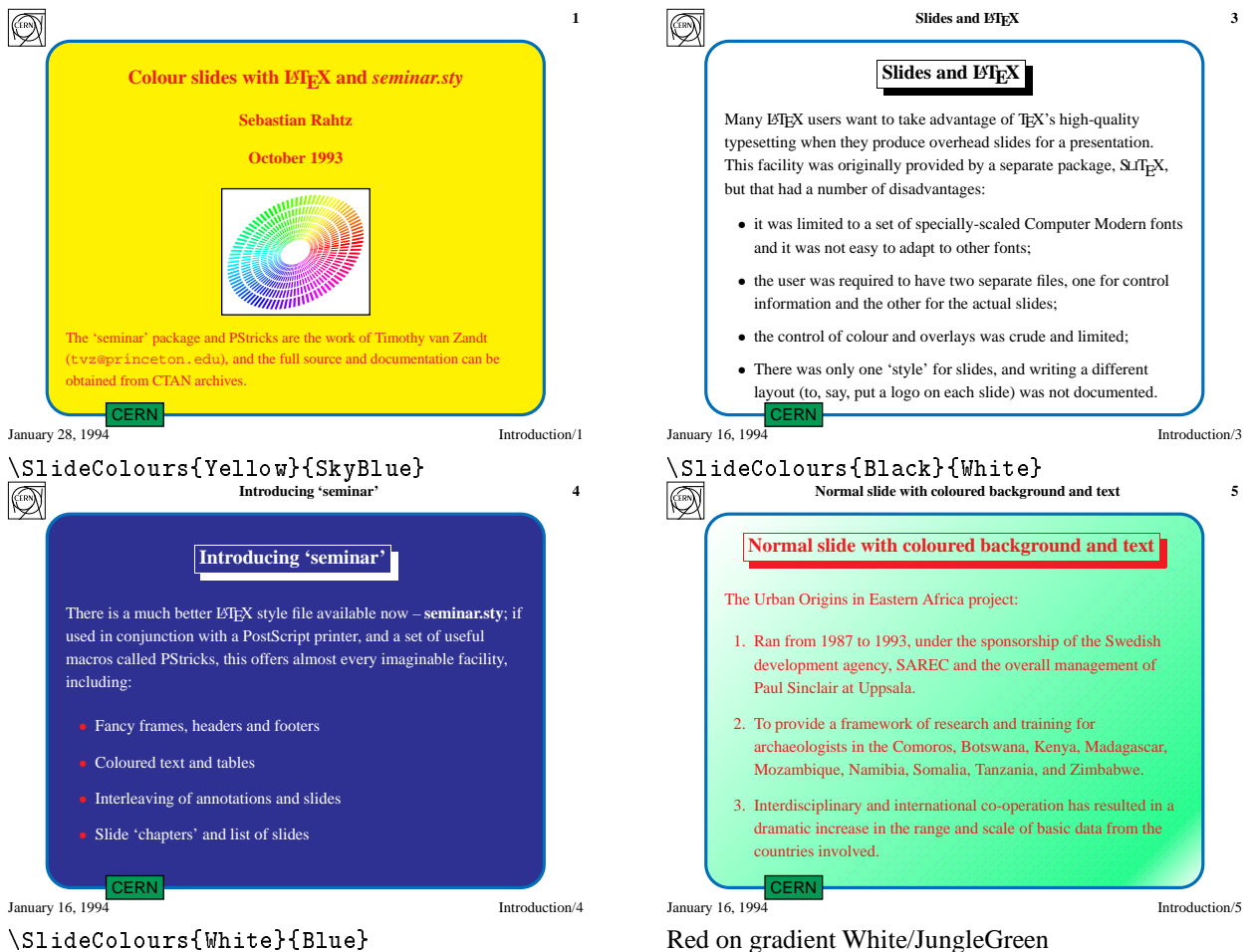


Figure 1: *Colour in slide background and foreground: (simulated in black and white)*

```
\newslideframe{blueframe} [%
\psset{linecolor=NavyBlue,%
linewidth=\slideframewidth,%
framesep=\slideframesep,%
cornersize=absolute,%
linearc=.5cm%
}]{\psframebox{#1}}
\def\SlideColours#1#2{%
\gdef\SlideFront{\color{#1}}%
\slideframe{\Framedefault}%
\slideframe*[\psset{fillcolor=#2,%
fillstyle=solid}]{blueframe}%
}
\def\blackandwhite{%
\SlideColours{Black}{White}%
}
```

The slide defaults will be for a detailed layout and CERN logo, with yellow writing on a blue background:

```
\pagestyle{cernsections}
\slideframe{cern}
\def\Framedefault{cern}
\SlideColours{Yellow}{RoyalBlue}
```

The result of this setup is show in Figure 1; Figure 2 shows how we might combine colour and seminar's overlay com-

mands to highlight the relationship between parts of a program schema.

If you do not like the CERN layout we describe above (or whatever your local system has provided), you can 'roll your own' to various degrees of sophistication. In the simplest case you can just write a file in your local directory:

```
% A do-nothing file seminar.con
\endinput
```

Of course it might be more sensible to build on our work by redefining the commands so that they do what you want. For instance, if you had the seminar.con described above, and wanted to change the default set-up to black on white and use a plain pagestyle and frame, you could add some code in front of the command `\begin{document}`:

```
\documentstyle{seminar}
\pagestyle{plain}
\slideframe{plain}
\renewcommand{\SlideFront}{%
{\color{Black}}
}
\begin{document}
\begin{slide}
...
```

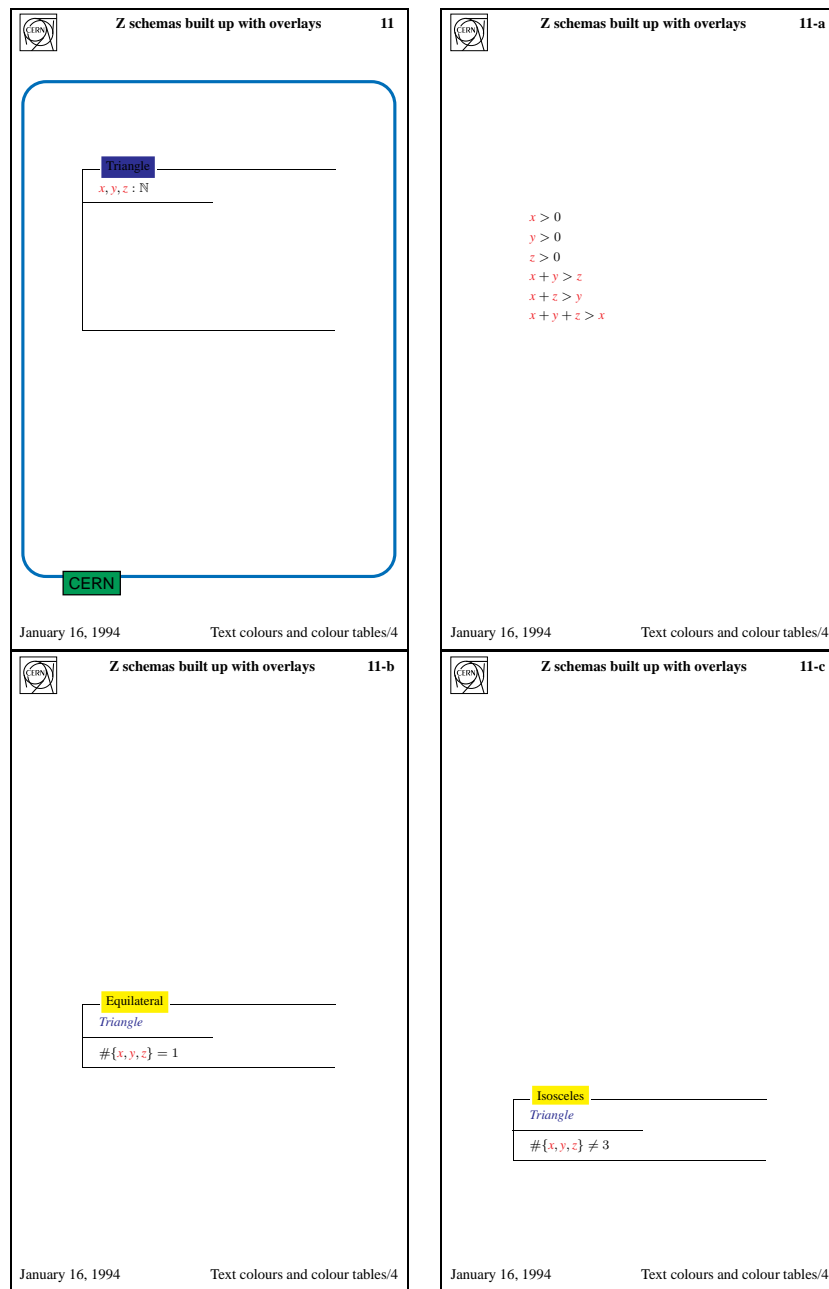


Figure 2: Slide overlays

If you still would like colour backgrounds for your slides by keeping the ‘SlideColours’ interface, but with a plain border and no CERN logos etc., you can use the following set-up:

```
\documentstyle{seminar}
\pagestyle{plain}
\renewcommand{\SlideColours}[2]{%
\renewcommand{\SlideFront}{\color{#1}}%
\slideframe%
[\psset{fillcolor=#2,fillstyle=solid}]%
```

```
{scplain}%
}
\begin{document}
\SlideColours{Red}{Yellow}
\begin{slide}
...
\end{slide}
\end{document}
```

The complete syntax of the many other commands which you have at your disposal is described in the *seminar* user guide; you will probably also want to read the *PStricks* manual!