

# Typesetting Figures under $\text{\LaTeX}$

## Author

Andrew Lincoln Burrow  
albcorp@gmail.com

## Abstract

The `AlbFloatTools` package provides a single `alb-float-tools`  $\text{\LaTeX}$  package to provide greater control over the layout of material in floats. It is tailored to the typesetting of mathematical figures. While the standard `figure` environment already supplies basic support, the package allows figure floats to extend into the document margins, and provides a mechanism to layout subfigures. The package is supported by an emacs lisp file customising  $\text{\AUCTeX}$  and  $\text{\RefTeX}$  which automates the markup process of figure floats, graphic inclusions, and `psfrag` rewriting, and also constructs labels in the file namespace. These extensions to  $\text{\AUCTeX}$  override the builtin support for the `graphicx` and `psfrag` packages.

## Copyright

Copyright © 2001–2006, 2013 Andrew Lincoln Burrow.

This program may be distributed and/or modified under the conditions of the  $\text{\LaTeX}$  Project Public License, either version 1.3 of this license or (at your option) any later version.

The latest version of this license is in

<http://www.latex-project.org/lppl.txt>

and version 1.3 or later is part of all distributions of  $\text{\LaTeX}$  version 2005/12/01 or later.

This work has the LPPL maintenance status ‘author-maintained’.

This work consists of the files

`alb-algorithms.sty`, `alb-avm.sty`, `alb-corp.cls`,  
`alb-float-tools.sty`, `alb-graph-theory.sty`,  
`alb-journal.cls`, `alb-order-theory.sty`,  
`alb-proofs.sty`, `alb-theorems.sty`, `alb-thesis.cls`,  
`alb-algorithms.tex`, `alb-avm.tex`,  
`alb-corp-layout.tex`, `alb-float-tools.tex`,  
`alb-graph-theory.tex`, `alb-journal-layout.tex`,  
`alb-order-theory.tex`, `alb-proofs.tex`,  
`alb-theorems.tex`, and `alb-thesis-layout.tex`.

## Version Information

Revision

Date

## 1 Introduction

The `alb-float-tools` L<sup>A</sup>T<sub>E</sub>X package is designed to extend the standard figures and the support provided by AUC<sub>T</sub>E<sub>X</sub> for both figures and graphics. The new L<sup>A</sup>T<sub>E</sub>X features allow figures to utilise the horizontal margin par allocations, and group subfigures. The AUC<sub>T</sub>E<sub>X</sub> extensions prompting for these new features, and override `graphicx.el` and `psfrag.el` to improve the prompting for graphics inclusion and label rewriting.

## 2 Using the Commands and Environments

The environments of `alb-float-tools` place certain syntactic restrictions on their use. This section concerns these restrictions and the typographic meaning of the markup. Section 3 describes how the AUC<sub>T</sub>E<sub>X</sub> customisation eases the burden of entering syntactically correct L<sup>A</sup>T<sub>E</sub>X code.

### 2.1 Wide Figures

The `alb-float-tools` L<sup>A</sup>T<sub>E</sub>X package provides the `albInflate` environment for use within a float environment such as `figure`. The environment allows float material to extend into the margin. It is designed to work in any combination of `oneside`, `twoside`, and `reverse` margin options, and also tests for case where the figure occurs within a column.

`\begin{albInflate}{label}` Inflate the column to include the marginpar allocation, where the marginpar location is calculated from the page reference of *label*. This environment should only be used inside a float with a label, and the label of the float and the environment must match!

For example,

```
\begin{figure}
  \begin{albInflate}{fig:example-wide-figure}
    \centering
    \includegraphics[scale=1.44]{poset-mapto-lattice.pdf}
  \end{albInflate}
  \caption[Example wide figure]{%
    An example of a wide figure.}
  \label{fig:example-wide-figure}
\end{figure}
```

produces Figure 1.

The definition is easy in `oneside` mode, but requires cunning in `twoside` mode. We must guess the final position of the float to decide which text edge to cross. The solution is to look at the page number cross reference. If the float was set on an odd page last run, then format for an odd page in this run. L<sup>A</sup>T<sub>E</sub>X will still converge because only vertical size affects page layout. Furthermore, it requires exactly the same number of L<sup>A</sup>T<sub>E</sub>X runs to converge, as the cross reference itself requires.

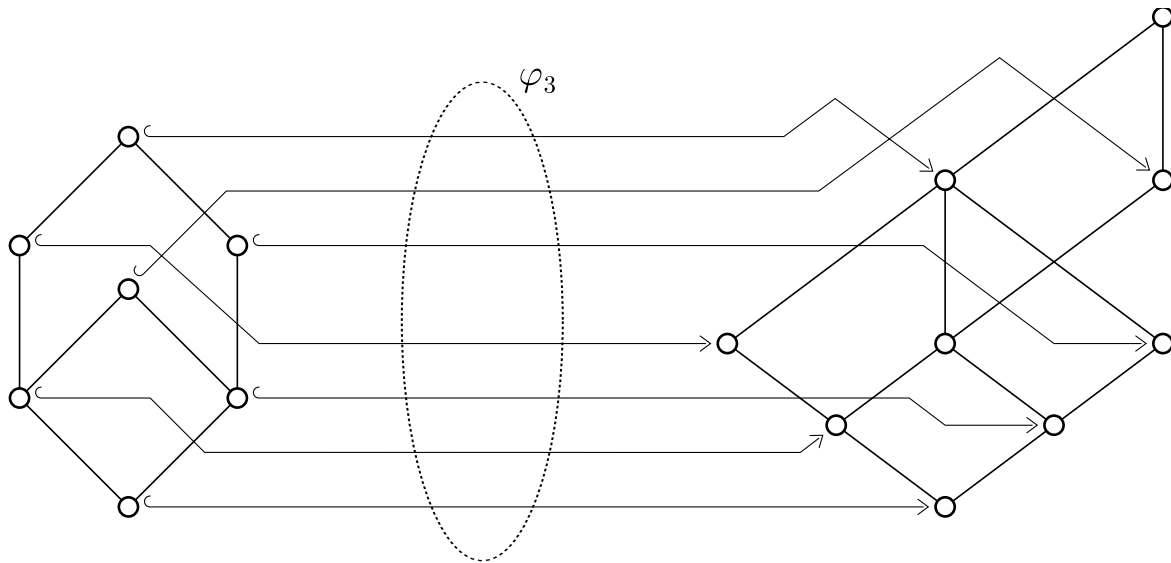


Figure 1: An example of a wide figure.

## 2.2 Collections of Subfigures

The `alb-float-tools` package provides the `albShelve` environment for use within a float environment such as `figure`. The environment collects and labels subfigures. The elements are distributed across the available width, and flow onto additional lines where required. An optional argument controls whether the elements are top-aligned or bottom-aligned.

`\begin{albShelve}[align]` Collect smaller figures within a larger figure. This is a list environment in which the `\item[label]` command marks-up subfigures with the optional tag *label*. However, the list does not use a counter so that it does not enumerate or support the `\label` command. The list displays the subfigures across the column with each line of subfigures evenly spread across the space. If *align* is `t` then the elements are top-aligned, and if *align* is `b` then the elements are bottom-aligned.

For example,

```
\begin{figure}
  \begin{albShelve}[b]
    \item[(a)] \includegraphics{down-closure-source.pdf}
    \item[(b)] \includegraphics{down-closure-of-a.pdf}
    \item[(c)] \includegraphics{down-closure-of-b.pdf}
    \item[(d)] \includegraphics{down-closure-of-c.pdf}
  \end{albShelve}
  \caption[Example subfigure collection]{%
    An example of a subfigure collection.}
  \label{fig:example-subf-coll}
\end{figure}
```

produces Figure 2.

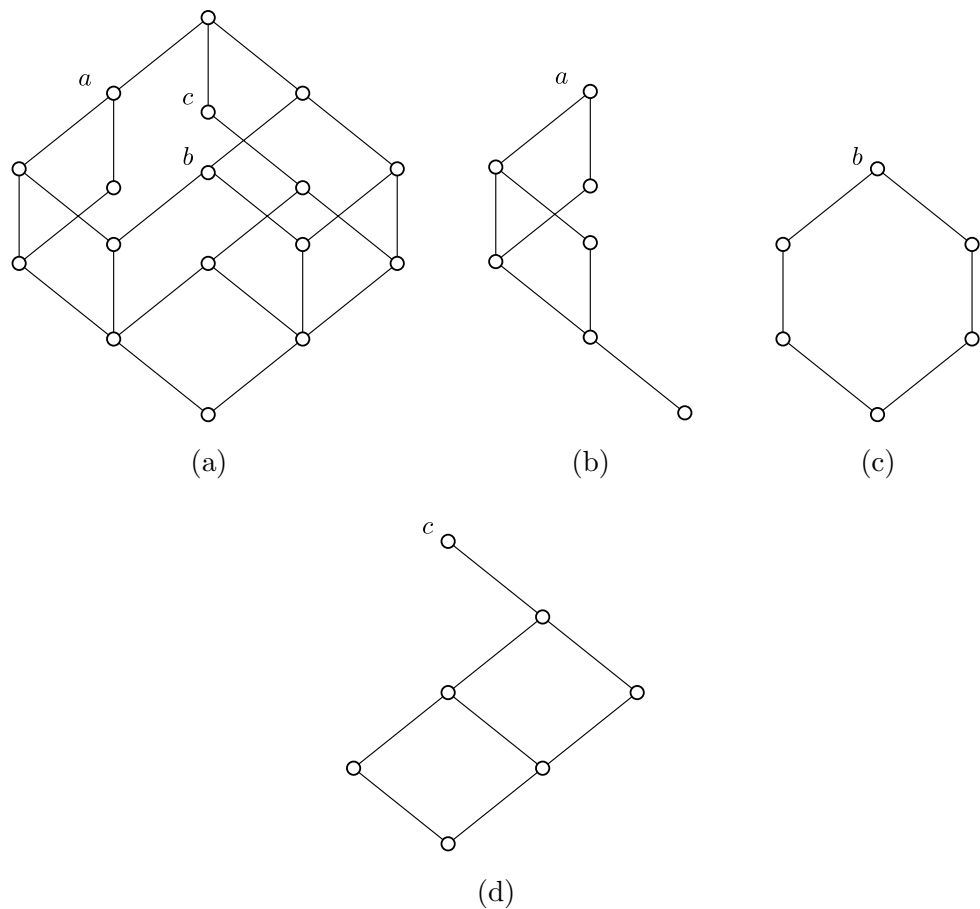


Figure 2: An example of a subfigure collection.

### 3 AUCT $\text{\E}$ X Customisations

Under AUCT $\text{\E}$ X the file `alb-float-tools.el` is automatically loaded whenever the `alb-float-tools` package is used. The customisation adds support for the symbols and environments from `alb-float-tools`. In addition, the AUCT $\text{\E}$ X supplied customisations for `graphicx` and `psfrag` are overridden. This results in the following improved prompting.

**figure** Prompts and constructs the float environment. The prompts follow the following sequence.

**(optional) Float to** Complete to one of the  $\text{\LaTeX}$  locations `h`, `t`, `b`, or `p`.

**(optional) Abbreviated caption:** The abbreviated caption is used in the list of figures, and to generate a label for the figure. It is best to have a well planned one line description ready before inserting a figure.

**Caption:** If you provide an abbreviated caption, then this starts as the abbreviated caption. You can expand it now, or later because the label will be derived from the abbreviated caption.

**Label:** If you provide an abbreviated caption, then this is already generated for you. Otherwise, you will need to provide a label. It is required for the `inflate` option.

**albInflate** Constructs the `inflate` environment inside a `float` environment, and searches within the enclosing environment for a label to use to compute the margin crossing.

**albShelve** Constructs the `shelve` environment inside a `float` environment, and prompts for the alignment of the elements.

**includegraphics** Constructs the `includegraphics` command, and prompts for options, and the local graphics files.

**psfrags** Constructs the `psfrags` environment and a list of `psfrag` command tag replacements. While the user inputs non-empty tags,  $\text{AUCT}_{\text{E}}\text{X}$  prompts for the components of a `psfrag` command.

## 4 Makefile Targets

The inclusion of graphic files affects document compilation, since graphic files must often be converted into an appropriate format. This in turn affects the choice of markup. The makefile from the `AlbLaTeXDocumentTemplate` package, attempts to automate the graphic file transformations for  $\text{PDF}_{\text{L}}\text{A}_{\text{T}}\text{E}_{\text{X}}$  compilation. In this case, the `includegraphics` command accepts PDF, JPEG, and PNG graphics. The makefile attempts to construct these formats from other source formats such as SVG. Therefore, each filename should include the extension to assist in the search for possible transformations.