

The asyfig packages

Will Robertson*

vo.1b 2009/06/30

Abstract

This suite of packages provides an alternate method of including stand-alone Asymptote figures within L^AT_EX documents via the `\asyfig` command.

Contents

I	USER DOCUMENTATION	1	II	IMPLEMENTATION	4
1	Introduction	1	4	The asyfig package	4
2	Getting started	2	5	The asyalign package	7
3	Package information	3	6	The asyprocess package	10

Part I

User documentation

1 Introduction

Asymptote (or `asy`) is a vector graphics programming language inspired by MetaPost but based around an extended C-like language and full support for 3D bezier curves. Asymptote uses an auxiliary L^AT_EX process to typeset its labels, and figures can be either generated as stand-alone graphics or in an ‘inline’ form in which labels get placed by the main typesetting process as the figure is inserted into a document.

Support for `asy` in a L^AT_EX document is provided by the `asymptote` package, which defines the `\begin{asy}` environment in which `asy` figures may be directly typed. In this case, the source file contains the complete specification

*wspr81@gmail.com

for the text and graphics in the document. However, for large documents it can be quite inconvenient to maintain `asy` graphics that are inline with the document source, because the whole document requires two compilations before any changes in the graphic can be visualised.

This package, `asyfig`, provides an alternative, whereby all `asy` figures are defined *separately* from the source in their own individual `.asy` files. `asyfig` uses Asymptote's inline mode so that labels in the graphics are produced by the main typesetting run; this ensures consistent font and size selection of text within the graphics. In addition, each individual `.asy` graphic can be very quickly processed individually to facilitate easy maintenance and editing of the graphics.

This package works with Asymptote v1.78 (and later, probably).

2 *Getting started*

Load the `asyfig` package like any other. I'll discuss the workflow of the package with an illustrative example.

An asy graphic First we need an example Asymptote graphic. This package is distributed with one such, `frf.asy`:

```
unitsize(10mm);
draw( (0,0){right}..{up}(3,2){down}..
      {down}(4,-2){up}..{right}(7,0) );
draw( "Resonance" , align=E, (3,2) );
draw( "Anti-resonance" , align=W, (4,-2) );
```

Material within `tex preamble` is *not* used in the final typesetting of the labels; it is purely for the 'proof' graphic that is produced before the graphic is integrated within the main document.

Inserting the graphic After processing (see the next step), this graphic can be included in the document with the `\asyfig{<graphic name>}` command. In the case of the example, this would be `\asyfig{frf}`. It does *not* take any option arguments like a regular graphic to affect the scaling or rotation of the graphic (cf. `\includegraphics`); you are expected to produce the figure in the correct size and orientation within Asymptote.

If all of your `.asy` files take a common path prefix (such as `./figures/asy/`), this can be defined with the `\asypath{<path>}` command. For example, instead of writing `\asyfig{asy/frf}` one can write `\asypath{asy/}` somewhere in the document (usually the preamble) and then later `\asyfig{frf}`.

Processing the graphic But before the graphic can be placed into the document it must be processed. If the `asy` graphic has not been processed, or if the `asy`

file is *newer* than its processed graphic, then this package will attempt to do the processing automatically. To turn off this automatic processing, load the package with the `[process=none]` package option. Alternatively, to re-process *all* asy graphics, use `[process=none]` instead.

The primary feature of this package is that figures may be processed independently of the main document in order to be able to rapidly iterate changes to the graphic. This processing is performed by the `asyprocess` package in an auxiliary \LaTeX execution. Here is a basic shell script that I use to do this:

```
#!/bin/sh
pdflatex -shell-escape -interaction=batchmode -jobname=$1-comp
"\RequirePackage{asyprocess}\ProcessAsy
\documentclass{article}\begin{document}\ShowAsy\end{document}"
```

Simply change `pdflatex` to `latex` to have EPS graphics produced by Asymptote. Note that it is *mandatory* to use the `-comp` suffix for the jobname.

By saving the script above into (say) `asyprocess` and making it executable, an individual asy graphic can be processed by running (following from the running example) `'asyprocess frf'`.

3 *Package information*

The most recent publicly released version of `asyfig` is available from CTAN:

<http://tug.ctan.org/pkg/asyfig/>

Historical and developmental versions are available at GitHub:

<http://github.com/wspr/asyfig/>

While general feedback at wspr81@gmail.com is welcomed, specific bugs should be reported through the issue tracker at GitHub: <http://github.com/wspr/asyfig/issues>.

This package is freely modifiable and distributable under the terms and conditions of the \LaTeX Project Public Licence, version 1.3c or greater (your choice). The latest version of this license is available at: <http://www.latex-project.org/lppl.txt>. This work is maintained by WILL ROBERTSON.

Part II

Implementation

4 *The asyfig package*

LaTeX2e file 'asyfig.sty' generated by the 'filecontents' environment from source 'asyfig' on 2009/06/30.

```
1 \ProvidesPackage{asyfig}[2009/06/30_v0.1b
2   Commands_for_using_asymptote_figures]
```

This package is the main user interface for inserting external asy figures into the document.

```
3 \RequirePackage{%
4   asyalign,color,ifmtarg,ifpdf,ifplatform,import,
5   graphicx,pdftexcmds,suffix,xkeyval}
```

Better conditionals than \newif provides:

```
\@True 6 \def\@True{11}
\@False 7 \def\@False{01}
\asy@if 8 \def\asy@if#1{\if#1\relax\expandafter\@firstoftwo\else%
          \expandafter\@secondoftwo\fi}

9 \let\asy@always\@False
10 \let\asy@never\@False
11 \let\asy@process\@False
```

Package options:

```
process 12 \define@choicekey*{asyfig.sty}{process}[\@tempa\@tempb]{%
          all,none,auto}{%
13   \ifcase\@tempb\relax
14     \let\asy@always\@True
15   \or
16     \let\asy@never\@True
17   \or
18     \fi
19 }
20 \ExecuteOptions{process=auto}
21 \ProcessOptionsX
```

4.1 Auxiliary macros

```
\asy@splitpath 22 \def\asy@splitpath#1/#2/{%
```

Recursive macro that is used like

```
\asy@splitpath abc/def/ghi.asy/\@nil/
```

It defines `\asy@filename` \rightarrow `ghi.asy` and `\asy@path` \rightarrow `abc/def/`

```
23 \ifx\@nil#2\relax
```

If input is *<anything>*/`\@nil/` then we've reached the end:

```
\asy@filename 24 \def\asy@filename{#1}%
```

```
25 \else
```

Otherwise we're in the middle of the slash-separated list; build up `\asy@path`, and iterate:

```
26 \edef\asy@path{\asy@path#1/}%
```

```
27 \def\@tempa{\asy@splitpath#2/}%
```

```
28 \expandafter\@tempa
```

```
29 \fi
```

```
30 }
```

```
\asy@path 31 \newcommand\asy@path[1]{\def\asy@pathprefix{#1}}
```

```
\asy@pathprefix 32 \asy@path{}
```

```
\asy@asyfile 33 \def\asy@asyfile{\asy@pathprefix\asy@path\asy@filename.asy}
```

```
\asy@texfile 34 \def\asy@texfile{\asy@pathprefix\asy@path\asy@filename%  
 \string_.tex}
```

```
\asy@cmdsep 35 \def\asy@cmdsep{\ifwindows\string&\else;\fi}
```

4.2 The main macro

```
\asyfig 36 \newcommand\asyfig[1]{%
```

```
37 \let\asy@path\@empty
```

```
38 \asy@splitpath_#1/\@nil/%
```

```
39 \IfFileExists{\asy@asyfile}{%
```

```
40 \asy@If\asy@process}{%
```

```
41 \asy@If\asy@always{%
```

```
42 \let\asy@process\@True
```

```
43 }{%
```

```
44 \IfFileExists{\asy@texfile}{%
```

```
45 \asy@If\asy@never{ }{%
```

compare file dates to see if we want to reprocess:

```
46 \ifnum\pdf@strcmp{\pdf@filemoddate{\asy@texfile}}
47 {\pdf@filemoddate{\asy@asyfile}}<
\z@
48 \let\asy@process\@True
49 \fi
50 }%
51 }{% if the .tex file doesn't exist, either give an error or process the
.asy file:
52 \asy@If\asy@never{%
53 \PackageError{asyfig}{%
54 ^^J\space\space\space\space
55 "\asy@pathprefix\asy@path\asy@filename.asy"
requires_processing%
56 }{%
57 The_generated_file_that_is_required_to_insert_the_
asy_graphic,
58 ^^J\space\space\space\space
59 "\asy@pathprefix\asy@path\asy@filename%
\string_.tex"^^J%
60 does_not_exist.
61 Please_process_the_asy_figure_manually_or_
de-activate_the^^J%
62 [process=none]_package_option.
63 }%
64 }{%
65 \let\asy@process\@True
66 }
67 }%
68 }}%
69 \asy@If\asy@process{%
70 \edef\@tempa{\asy@pathprefix\asy@path}%
71 \pdf@system{%
72 echo"^^J=====ASY_PROCESS=====^^J"
73 \asy@cmdsep
74 \ifx\@tempa\@empty\else
75 cd\@tempa
76 \asy@cmdsep
77 \fi
```

```

78     \ifpdf\pdf\fi\latex
79     -shell-escape
80     -interaction=batchmode
81     -jobname=\asy@filename-comp
82     \unexpanded{%
83       "\RequirePackage{asyprocess}\ProcessAsy
84       \documentclass{article}
85       \begin{document}\ShowAsy
86       \end{document}"
87     }%
88     \asy@cmdsep
89     echo "\^^J====\ASY\END\PROCESS\^^J"
90   }%
91 }{}%
92 \import{\asy@pathprefix\asy@path}{\asy@filename%
93   \string_.tex}%
94 }{%
95   \PackageWarning{asyfig}{%
96     ^^J\space\space
97     "\asy@pathprefix\asy@path\asy@filename.asy" \_not\_
98     found.^^J%
99     This\_warning\_occurred%
100   }%
101 }%
102 \let\asy@process\@False
103 }

```

The starred version of `\asyfig` processes the graphic always:

```

\asyfig* 102 \WithSuffix\newcommand\asyfig*[1]{%
103   \begingroup
104     \let\asy@process\@True
105     \csname\NoSuffixName\asyfig\endcsname{#1}%
106   \endgroup
107 }

```

<eof>

5 *The asyalign package*

LaTeX2e file 'asyalign.sty' generated by the 'filecontents' environment from source 'asyfig' on 2009/06/30.

```
1 \ProvidesPackage{asyalign}
```

This package provides code for placing Asymptote labels inline in L^AT_EX documents. It is adapted from code that is usually included within Aymptote's `<filename>_pre` file, which provides a L^AT_EX preamble for asy processing; this preamble is skipped with the asyfig package since all figures inherit the preamble from that of the main document.

```
2 \RequirePackage{ifpdf}
```

```
\ASYbox 3 \newbox\ASYbox
```

```
\ASYdimen 4 \newdimen\ASYdimen
```

```
\ASYbase 5 \long\def\ASYbase#1#2{%
```

```
6 \leavevmode
```

```
7 \setbox\ASYbox\hbox{#1}%
```

```
8 \ASYdimen=\ht\ASYbox
```

```
9 \setbox\ASYbox\hbox{#2}%
```

```
10 \lower\ASYdimen\box\ASYbox
```

```
11 }
```

```
12 \ifpdf
```

```
\ASYaligned 13 \long\def\ASYaligned(#1,#2)(#3,#4)#5#6#7{%
```

```
14 \leavevmode
```

```
15 \setbox\ASYbox\hbox{#7}%
```

```
16 \setbox\ASYbox\hbox{%
```

```
17 \ASYdimen\ht\ASYbox
```

```
18 \advance\ASYdimen\dp\ASYbox
```

```
19 \kern#3\wd\ASYbox
```

```
20 \raise#4\ASYdimen
```

```
21 \box\ASYbox
```

```
22 }%
```

```
23 \put(#1,#2){%
```

```
24 #5\wd\ASYbox_0pt\dp\ASYbox_0pt\ht\ASYbox_0pt\box%
```

```
\ASYbox#6%
```

```
25 }%
```

```
26 }
```

```
\ASYalignT 27 \long\def\ASYalignT(#1,#2)(#3,#4)#5#6{%
```

```
28 \ASYaligned(#1,#2)(#3,#4){%
```

```
29 \special{pdf:q_#5_0_0_0_cm}%
```



```

30     }{%
31     \special{pdf:Q}%
32     }{#6}%
33 }

\ASYalign 34 \long\def\ASYalign(#1,#2)(#3,#4)#5{%
          \ASYaligned(#1,#2)(#3,#4){}{-}{#5}}

35 \let\ASYraw\@firstofone

36 \else

\ASYaligned 37 \long\def\ASYaligned(#1,#2)(#3,#4)#5#6#7{%
38     \leavevmode
39     \setbox\ASYbox\hbox{#7}%
40     \setbox\ASYbox\hbox{%
41         \ASYdimen\ht\ASYbox%
42         \advance\ASYdimen\dp\ASYbox
43         \kern#3\wd\ASYbox
44         \raise#4\ASYdimen
45         \box\ASYbox
46     }%
47     \put(#1,#2){#5\wd\ASYbox\Opt\dp\ASYbox\Opt\ht\ASYbox\Opt%
          \box\ASYbox#6}%
48 }

\ASYalignT 49 \long\def\ASYalignT(#1,#2)(#3,#4)#5#6{%
50     \ASYaligned(#1,#2)(#3,#4){%
51         \special{%
52             ps:gsave\currentpoint\currentpoint\translate
53             [#5\0\0]\concat\neg\exch\neg\exch\translate%
54         }%
55     }{%
56         \special{ps:currentpoint\grestore\moveto}%
57     }{#6}%
58 }

\ASYalign 59 \long\def\ASYalign(#1,#2)(#3,#4)#5{%
          \ASYaligned(#1,#2)(#3,#4){}{-}{#5}}

\ASYraw 60 \def\ASYraw#1{%
61     currentpoint\currentpoint\translate\matrix\currentmatrix
62     100\12\div\100\12\div\scale

```

```

63     #1
64     setmatrix_neg_exch_neg_exch_translate%
65   }

66 \fi

⟨eof⟩

```

6 *The asyprocess package*

LaTeX2e file 'asyprocess.sty' generated by the 'filecontents' environment from source 'asyfig' on 2009/06/30.

```

1 \ProvidesPackage{asyprocess}
2 \nofiles

3 \RequirePackage{%
4     ifmtarg,ifpdf,catchfile,ifplatform,color,graphicx}
5 \RequirePackage[active,tightpage]{preview}

\@par@macro 5 \def\@par@macro{\par}

\asy@status 6 \def\asy@status{asyprocess-statusfile.txt}

7 \edef\@tempa{\detokenize{-comp}}
\asy@strip@comp 8 \@temptokena{\def\asy@strip@comp#1}
9 \expandafter\the\expandafter\@temptokena\@tempa#2\@nil{%
10     \@ifmtarg{#2}{%
11         \errorstopmode
12         \PackageError{asyprocess}{%
13             The\string\jobname\space of\space this\space compilation\space must\space end\space
14             with\space '-comp'%
15             }{%
16                 You\space must\space set\space the\space \cmd\jobname\space with\space the\space equivalent\space
17                 of^^J\space\space
18                 pdflatex\space-jobname=XYZ-comp\space...%
19             }
20         }{}%
21     \edef\asy@compname{#1}}
22 \expandafter\expandafter\expandafter
23 \asy@strip@comp\expandafter\jobname\@tempa\@nil

24 \newcommand\ProcessAsy{%

```

```

\ProcessAsy 23 \immediate\write18{%
24   asy_-wait_-inlinetex_-tex_\ifpdf_pdf\fi_latex
25   \asy@compname\space_2>\asy@status}%
26 \CatchFileDef{\@tempb}{\asy@status}{}%
27 \immediate\write18{\ifwindows_del_\else_rm_\fi_\asy@status}
28 \ifx\@tempb\@par@macro
29   \expandafter\@gobble
30 \else
31   \g@addto@macro\@tempb{^^J^^J%
32   -----_ASY_ERROR_-----^^J%
33   -----}%
34   \expandafter\@firstofone
35 \fi{%
36   \errorstopmode
37   \typeout{%
38     -----^^J%
39     -----_ASY_ERROR_-----^^J}
40   \typeout{\expandafter\strip@prefix\meaning\@tempb}
41   \batchmode
42   \end{document}}}}

\ShowAsy 43 \newcommand\ShowAsy{%
44   \begin{preview}
45   \input{\asy@compname_}
46   \end{preview}}

47 \AtBeginDocument{\InputIfFileExists{\asy@compname_.pre}{-}{-}}

<eof>

```