The asyfig packages

Will Robertson*

v0.1c 2010/03/20

Abstract

This suite of packages provides an alternate method of including stand-alone Aymptote figures within LaTeX documents via the `\asyfig` command.

Contents

I User documentation 1
1 Introduction 1
2 Do you need this package? 2
3 Getting started 2
4 Package information 3

II Implementation 5
5 The asyfig package 5
6 The asyalign package 8
7 The asyprocess package 11

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 LaTeX 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 LaTeX document is provided by the asymptote package,

*wspr81@gmail.com

which defines the `\begin{asy}` environment in which `asy` figures may be directly typed. In this case, the source file contains the complete specification 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 sometimes lags behind the current release of Asymptote simply because I don't use Asymptote very often. The current release of this package is designed to work with Asymptote v1.91 and later.

2 *Do you need this package?*

After I wrote and used this package for quite some time, I realised that what it is intended to do can be done with the standard asymptote package. If you have an Aymptote graphic called `myfig.asy`, you can include it in your document as follows:

```
\begin{asy}
include "myfig";
\end{asy}
```

There's actually not much point using this package if this works for you. But I'll keep supporting this package for now while I continue to use it.

3 *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 `texpreamble` 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=all]` 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`'.

4 *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.

Part II
Implementation

5 *The asyfig package*

LaTeX2e file 'asyfig.sty' generated by the 'filecontents' environment from source 'asyfig' on 2010/03/20.

```
\ProvidesPackage{asyfig}[2010/03/20 v0.1c
  Commands for using asymptote figures]
```

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

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

Better conditionals than `\newif` provides:

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

\let\asy@always\@False
\let\asy@never\@False
\let\asy@process\@False
```

Package options:

```
process \define@choicekey*{asyfig.sty}{process}[\@tempa\@tempb]{%
          all,none,auto}{%
  \ifcase\@tempb\relax
    \let\asy@always\@True
  \or
    \let\asy@never\@True
  \or
  \fi
}
\ExecuteOptions{process=auto}

\ProcessOptionsX
```

5.1 Auxiliary macros

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

Recursive macro that is used like
`\asy@splitpath abc/def/ghi.asy/\@nil/`
It defines `\asy@filename` → `ghi.asy` and `\asy@path` → `abc/def/`

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

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

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

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

```
    \edef\asy@path{\asy@path#1/}%
    \def\@tempa{\asy@splitpath#2/}%
    \expandafter\@tempa
  \fi
}
```

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

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

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

5.2 The main macro

\asyfig
```
\newcommand\asyfig[1]{%
  \let\asy@path\@empty
  \asy@splitpath␣#1/\@nil/%
  \IfFileExists{\asy@asyfile}{%
    \asy@If\asy@process{}{%
    \asy@If\asy@always{%
      \let\asy@process\@True
    }{%
      \IfFileExists{\asy@texfile}{%
```

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

compare file dates to see if we want to reprocess:

```
        \ifnum\pdf@strcmp{\pdf@filemoddate{\asy@texfile}}
                         {\pdf@filemoddate{\asy@asyfile}}<
                               \z@
          \let\asy@process\@True
        \fi
      }%
    }{% if the .tex file doesn't exist, either give an error or process the
        .asy file:
      \asy@If\asy@never{%
        \PackageError{asyfig}{%
          ^^J\space\space\space\space
          "\asy@pathprefix\asy@path\asy@filename.asy"
               requires processing%
        }{%
          The generated file that is required to insert the
               asy graphic,
          ^^J\space\space\space\space
          "\asy@pathprefix\asy@path\asy@filename%
               \string_.tex"^^J%
          does not exist.
          Please process the asy figure manually or
               de-activate the^^J%
          [process=none] package option.
        }%
      }{%
        \let\asy@process\@True
      }
    }%
  }}%
  \asy@If\asy@process{%
    \edef\@tempa{\asy@pathprefix\asy@path}%
    \pdf@system{%
      echo "^^J====== ASY PROCESS =====^^J"
      \asy@cmdsep
      \ifx\@tempa\@empty\else
        cd \@tempa
        \asy@cmdsep
      \fi
```

```
          \ifpdf␣pdf\fi␣latex
            -shell-escape
            -interaction=batchmode
            -jobname=\asy@filename-comp
          \unexpanded{%
            "\RequirePackage{asyprocess}\ProcessAsy
             \documentclass{article}
             \begin{document}\ShowAsy
             \end{document}"
          }%
          \asy@cmdsep
          echo␣"^^J====␣ASY␣END␣PROCESS␣===^^J"
        }%
      }{}%
      \import{\asy@pathprefix\asy@path}{\asy@filename%
           \string_.tex}%
    }{%
      \PackageWarning{asyfig}{%
        ^^J\space\space
        "\asy@pathprefix\asy@path\asy@filename.asy"␣not␣
             found.^^J%
        This␣warning␣occurred%
      }%
    }%
    \let\asy@process\@False
}
```

The starred version of \asyfig processes the graphic always:

\asyfig*

```
\WithSuffix\newcommand\asyfig*[1]{%
  \begingroup
    \let\asy@process\@True
    \csname\NoSuffixName\asyfig\endcsname{#1}%
  \endgroup
}
```

⟨*eof*⟩

6 The asyalign package

LaTeX2e file 'asyalign.sty' generated by the 'filecontents' environment from source 'asyfig' on 2010/03/20.

```
\ProvidesPackage{asyalign}
```

This package provides code for placing Asymptote labels inline in LaTeX documents. It is adapted from code that is usually included within Asymptote's ⟨*filename*⟩`_.pre` file, which provides a LaTeX preamble for `asy` processing; this preamble is skipped with the asyfig package since all figures inherit the preamble from that of the main document.

```
\RequirePackage{ifpdf}

\ASYbox
\newbox\ASYbox
\ASYdimen
\newdimen\ASYdimen

\ASYbase
\long\def\ASYbase#1#2{%
  \leavevmode
  \setbox\ASYbox\hbox{#1}%
  \ASYdimen=\ht\ASYbox
  \setbox\ASYbox\hbox{#2}%
  \lower\ASYdimen\box\ASYbox
}

\ifpdf

\ASYaligned
  \long\def\ASYaligned(#1,#2)(#3,#4)#5#6#7{%
    \leavevmode
    \setbox\ASYbox\hbox{#7}%
    \setbox\ASYbox\hbox{%
      \ASYdimen\ht\ASYbox
      \advance\ASYdimen\dp\ASYbox
      \kern#3\wd\ASYbox
      \raise#4\ASYdimen
      \box\ASYbox
    }%
    \put(#1,#2){%
      #5\wd\ASYbox␣0pt\dp\ASYbox␣0pt\ht\ASYbox␣0pt\box%
            \ASYbox#6%
    }%
  }

\ASYalignT
  \long\def\ASYalignT(#1,#2)(#3,#4)#5#6{%
    \ASYaligned(#1,#2)(#3,#4){%
      \special{pdf:q␣#5␣0␣0␣cm}%
```

```
    }{%
      \special{pdf:Q}%
    }{#6}%
  }

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

  \let\ASYraw\@firstofone

\else

\ASYaligned  \long\def\ASYaligned(#1,#2)(#3,#4)#5#6#7{%
    \leavevmode
    \setbox\ASYbox\hbox{#7}%
    \setbox\ASYbox\hbox{%
      \ASYdimen\ht\ASYbox%
      \advance\ASYdimen\dp\ASYbox
      \kern#3\wd\ASYbox
      \raise#4\ASYdimen
      \box\ASYbox
    }%
    \put(#1,#2){#5\wd\ASYbox␣0pt\dp\ASYbox␣0pt\ht\ASYbox␣0pt%
          \box\ASYbox#6}%
  }

\ASYalignT  \long\def\ASYalignT(#1,#2)(#3,#4)#5#6{%
    \ASYaligned(#1,#2)(#3,#4){%
      \special{%
        ps:gsave␣currentpoint␣currentpoint␣translate
        [#5␣0␣0]␣concat␣neg␣exch␣neg␣exch␣translate%
      }%
    }{%
      \special{ps:currentpoint␣grestore␣moveto}%
    }{#6}%
  }

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

  \ASYraw  \def\ASYraw#1{%
    currentpoint␣currentpoint␣translate␣matrix␣currentmatrix
    100␣12␣div␣-100␣12␣div␣scale
```

```
      #1
      setmatrix␣neg␣exch␣neg␣exch␣translate%
    }

\fi
```

⟨*eof*⟩

7 *The asyprocess package*

LaTeX2e file 'asyprocess.sty' generated by the 'filecontents' environment from source 'asyfig' on 2010/03/20.

```
\ProvidesPackage{asyprocess}
\nofiles

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

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

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

\edef\@tempa{\detokenize{-comp}}
\asy@strip@comp  \@temptokena{\def\asy@strip@comp#1}
\expandafter\the\expandafter\@temptokena\@tempa#2\@nil{%
  \@ifmtarg{#2}{%
    \errorstopmode
    \PackageError{asyprocess}{%
      The␣\string\jobname\space␣of␣this␣compilation␣must␣end␣
            with␣'-comp'%
    }{%
      You␣must␣set␣the␣\cmd\jobname\␣with␣the␣equivalent␣
            of^^J\space\space
      pdflatex␣-jobname=XYZ-comp␣...%
    }
  }{}%
  \edef\asy@compname{#1}}
\expandafter\expandafter\expandafter
  \asy@strip@comp\expandafter\jobname\@tempa\@nil

\newcommand\ProcessAsy{%
```

\ProcessAsy
```
  \immediate\write18{%
    asy␣-wait␣-inlinetex␣-noprc␣-render␣0␣-tex␣\ifpdf␣pdf\fi␣
        latex
      \asy@compname\space␣2>␣\asy@status}%
  \CatchFileDef{\@tempb}{\asy@status}{}%
  \immediate\write18{\ifwindows␣del␣\else␣rm␣\fi␣\asy@status}
  \ifx\@tempb\@par@macro
    \expandafter\@gobble
  \else
    \g@addto@macro\@tempb{^^J^^J%
      ------------␣ASY␣ERROR␣------------^^J%
      ----------------------------------}%
    \expandafter\@firstofone
  \fi{%
     \errorstopmode
     \typeout{%
       ----------------------------------^^J%
       ------------␣ASY␣ERROR␣------------^^J}
     \typeout{\expandafter\strip@prefix\meaning\@tempb}
     \batchmode
     \end{document}}}
```

\ShowAsy
```
\newcommand\ShowAsy{%
  \begin{preview}
    \input{\asy@compname_}
  \end{preview}}

\AtBeginDocument{\InputIfFileExists{\asy@compname_.pre}{}{}}
```

⟨*eof*⟩