The asymptote package

John Bowman, Tom Prince, and Will Robertson

2011/05/15 v1.24

Abstract

This package provides integration of inline and external Asymptote graphics within a LaTeX document.

Contents

1 Introduction

This is the documentation for the LaTeX package `asymptote` which accompanies the Asymptote drawing package. For further details on Asymptote, please see its documentation in `asymptote.pdf`.

2 User syntax

2.1 Package loading and options

The package may take two options at load time: `inline` or `attach`. These options can also be set at any time with the `\asysetup{`⟨*options*⟩`}` command, or specified individually in the optional argument to each `asy` environment or `asyinclude` command.

The `inline` option uses Asymptote's 'inline' mode whereby included graphics have their labels typeset in the environment of the document they are contained within. Otherwise the Asymptote graphics are self-contained and their formatting is independent of the document.

The `attach` option allows generated graphics to be embedded within the PDF using the `attachfile2` package; please load that package separately if you wish to use it. The `attach` option takes precedence over the `inline` option.

This package produces quite a number of output files, which by default are created in the same directory as the LaTeX document that is being compiled. To keep things more tidy, you can specify an output directory for these files by defining the `\asydir` command. For example, if you wish to store the figure files in the subdirectory `asytmp/`, the you would write `\renewcommand\asydir{asytmp}`.

Alternatively (and tentatively), you may write `dir=asytmp` in either the `asy` environment options or the options to `\asysetup`.

2.2 Commands for inserting Asymptote graphics

The main environment defined by the package is the `asy` environment, in which verbatim Asymptote code is placed that will be compiled for generating a graphic in the document. For example,

```
\begin{figure}
\begin{asy}[ <options> ]
<ASYMPTOTE CODE>
\end{asy}
\caption{...}\label{...}
```

If you have Asymptote code in a separate file, you can include it with the `\asyinclude[`⟨*options*⟩`]{`⟨*filename*⟩`}` command.

For Asymptote code that should be included in *every* graphic, define it using the `asydef` environment.

2.3 Graphics options

Both the `asy` environment and the `\asyinclude` command take optional parameters for controlling aspects of the graphics creation. In addition to locally setting `inline` and `attach`, the following options may also be used:

height Height of the figure

width Width of the figure

keepAspect Maintain aspect ratio [default true]

viewportheight Viewport height for 3D figures

viewportwidth Viewport width for 3D figures

These may also be set globally using the `\asysetup` command.

3 Processing the document

After running LaTeX on the document, it is necessary to process the Asymptote graphics so they can be included in the next compilation. The simplest procedure is a recipe such as

```
pdflatex mydoc
asy mydoc-*.asy
pdflatex mydoc
```

This technique will recompile each graphic every time, however. To only recompile graphics that have changed, use the `latexmk` tool. Asymptote is distributed with a `latexmkrc` configuration file; place this file in a place where `latexmk` will find it and your document may be compiled, including the `asy` compilations, with `latexmk mydoc` or `latexmk --pdf mydoc`.

4 Implementation

```
\def\Asymptote{{\tt Asymptote}}
\InputIfFileExists{\jobname.pre}{}{}
```

4.1 Allocations

Allocations

```
\newbox\ASYbox
\newcounter{asy}
\newwrite\AsyStream
\newwrite\AsyPreStream
\newif\ifASYinline
\newif\ifASYattach
\newif\ifASYkeepAspect
\ASYkeepAspecttrue
```

4.2 Packages

```
\RequirePackage{keyval}
\RequirePackage{ifthen}
\RequirePackage{color,graphicx}
```

Emulating packages We cannot assume that Asymptote users have recent TEX distributions. (E.g., Fedora until recently still shipped teTeX.) So load ifpdf and ifxetex if they exist; otherwise, emulate them.

In due course, delete this code and just load the packages.

```
\IfFileExists{ifpdf.sty}{
  \RequirePackage{ifpdf}
}{
  \expandafter\newif\csname ifpdf\endcsname
  \ifx\pdfoutput\@undefined\else
    \ifcase\pdfoutput\else
      \pdftrue
    \fi
  \fi
}
\IfFileExists{ifxetex.sty}{
  \RequirePackage{ifxetex}
}{
  \expandafter\newif\csname ifxetex\endcsname
  \ifx\XeTeXversion\@undefined\else
    \xetextrue
  \fi
}
```

`\CatchFileDef` Used for `\asyinclude`.

```
\IfFileExists{catchfile.sty}{
```

```
  \RequirePackage{catchfile}
}{
  \newcommand\CatchFileDef[3]{%
    \begingroup
    \everyeof{%
      \ENDCATCHFILEMARKER
      \noexpand
    }%
    \long\def\@tempa##1\ENDCATCHFILEMARKER{%
      \endgroup
      \def#1{##1}%
    }%
    #3%
    \expandafter\@tempa\@@input #2\relax
  }
}
```

Ensuring attachfile2 is loaded if [attach] is requested

```
\newif\if@asy@attachfile@loaded
\AtBeginDocument{%
  \@ifpackageloaded{attachfile2}{\@asy@attachfile@loadedtrue}{}%
  \let\asy@check@attachfile\asy@check@attachfile@loaded
}
\newcommand\asy@check@attachfile@loaded{%
  \if@asy@attachfile@loaded\else
    \PackageError{asymptote}{You must load the attachfile2 package}{^^J%
      You have requested the [attach] option for some or all of your^^J%
      Asymptote graphics, which requires the attachfile2 package.^^J%
      Please load it in the document preamble.^^J%
    }%
  \fi
}
\newcommand\asy@check@attachfile{%
  \AtBeginDocument{\asy@check@attachfile@loaded}%
  \let\asy@check@attachfile\@empty
}
```

Macros

```
\def\csarg#1#2{\expandafter#1\csname#2\endcsname}
```

4.3 Package options

```
\DeclareOption{inline}{%
  \ASYinlinetrue
}
\DeclareOption{attach}{%
  \asy@check@attachfile
```

```
  \ASYattachtrue
}
\ProcessOptions*
\def\asydir{}
\def\ASYprefix{}
```

4.4 Testing for PDF output

Note this is not quite the same as `\ifpdf`, since we still want PDF output when using XeTeX.

```
\newif\ifASYPDF
\ifxetex
  \ASYPDFtrue
\else
  \ifpdf
    \ASYPDFtrue
  \fi
\fi
\ifASYPDF
  \def\AsyExtension{pdf}
\else
  \def\AsyExtension{eps}
\fi
```

4.5 Bug squashing

```
\def\unquoteJobname#1"#2"#3\relax{%
  \def\rawJobname{#1}%
  \ifx\rawJobname\empty
    \def\rawJobname{#2}%
  \fi
}
\expandafter\unquoteJobname\jobname""\relax
```

Work around jobname bug in MiKTeX 2.5 and 2.6: Turn stars in file names (resulting from spaces, etc.) into minus signs

```
\def\fixstar#1*#2\relax{%
  \def\argtwo{#2}%
  \ifx\argtwo\empty
    \gdef\Jobname{#1}%
  \else
    \fixstar#1-#2\relax
  \fi
}
\expandafter\fixstar\rawJobname*\relax
```

Work around bug in dvips.def: allow spaces in file names.

```
\def\Ginclude@eps#1{%
  \message{<#1>}%
  \bgroup
  \def\@tempa{!}%
```

```
  \dimen@\Gin@req@width
  \dimen@ii.1bp\relax
  \divide\dimen@\dimen@ii
  \@tempdima\Gin@req@height
  \divide\@tempdima\dimen@ii
    \special{PSfile=#1\space
      llx=\Gin@llx\space
      lly=\Gin@lly\space
      urx=\Gin@urx\space
      ury=\Gin@ury\space
      \ifx\Gin@scalex\@tempa\else rwi=\number\dimen@\space\fi
      \ifx\Gin@scaley\@tempa\else rhi=\number\@tempdima\space\fi
      \ifGin@clip clip\fi}%
  \egroup
}
```

4.6 Input/Output

```
\immediate\openout\AsyPreStream=\jobname.pre\relax
\AtEndDocument{\immediate\closeout\AsyPreStream}

\def\WriteAsyLine#1{%
  \immediate\write\AsyStream{\detokenize{#1}}%
}

\def\globalASYdefs{}
\def\WriteGlobalAsyLine#1{%
  \expandafter\g@addto@macro
  \expandafter\globalASYdefs
  \expandafter{\detokenize{#1^^J}}%
}
```

4.7 Commands for verbatim processing environments

```
\def\ProcessAsymptote#1{%
  \begingroup
  \def\CurrentAsymptote{#1}%
  \let\do\@makeother \dospecials
  \@makeother\^^L% and whatever other special cases
  \catcode`\ =10
  \endlinechar`\^^M \catcode`\^^M=12 \xAsymptote
}
```

Need lots of comment chars here because ⟨*line end*⟩ is no longer a space character.

```
\begingroup
  \catcode`\^^M=12 \endlinechar=-1\relax%
  \gdef\xAsymptote{%
    \expandafter\ProcessAsymptoteLine%
  }
  \gdef\ProcessAsymptoteLine#1^^M{%
    \def\@tempa{#1}%
    {%
      \escapechar=-1\relax%
```

```
      \xdef\@tempb{\string\\end\string\{\CurrentAsymptote\string\}}%
    }%
    \ifx\@tempa\@tempb%
      \edef\next{\endgroup\noexpand\end{\CurrentAsymptote}}%
    \else%
      \ThisAsymptote{#1}%
      \let\next\ProcessAsymptoteLine%
    \fi%
    \next%
  }
\endgroup
```

4.8 User interface

```
\newcommand\asy[1][]{%
  \stepcounter{asy}%
  \setkeys{ASYkeys}{#1}%
```

Disable the "inline" option if "attach" is enabled:

```
  \ifASYattach
    \ASYinlinefalse
  \fi
  \ifx\asydir\empty\else
    \def\ASYprefix{\asydir/}%
  \fi
  \immediate\write\AsyPreStream{%
    \noexpand\InputIfFileExists{%
      \ASYprefix\noexpand\jobname-\the\c@asy.pre}{}{}%
  }
  \asy@write@graphic@header
  \let\ThisAsymptote\WriteAsyLine
  \ProcessAsymptote{asy}%
}
\def\endasy{%
  \asy@finalise@stream
  \asy@input@graphic
}
\def\asy@write@graphic@header{%
  \immediate\openout\AsyStream=\ASYprefix\jobname-\the\c@asy.asy\relax
  \gdef\AsyFile{\ASYprefix\Jobname-\the\c@asy}%
  \immediate\write\AsyStream{%
    if(!settings.multipleView) settings.batchView=false;^^J%
    \ifxetex
      settings.tex="xelatex";^^J%
    \else\ifASYPDF
        settings.tex="pdflatex";^^J%
    \fi\fi
    \ifASYinline
      settings.inlinetex=true;^^J%
      deletepreamble();^^J%
```

```
    \fi
    defaultfilename="\Jobname-\the\c@asy";^^J%
    if(settings.render < 0) settings.render=4;^^J%
    settings.outformat="";^^J%
    \ifASYattach
      settings.inlineimage=false;^^J%
      settings.embed=false;^^J%
      settings.toolbar=true;^^J%
    \else
      settings.inlineimage=true;^^J%
      settings.embed=true;^^J%
      settings.toolbar=false;^^J%
      viewportmargin=(2,2);^^J%
    \fi
    \globalASYdefs
  }%
}
\def\asy@expand@keepAspect{%
  \ifASYkeepAspect keepAspect=true%
  \else keepAspect=false%
  \fi%
}
\def\asy@finalise@stream{%
```

Setting `size()`. Only inserted if one of the dimensions is set explicitly (i.e., if both height and width are not empty).

```
  \ifx\ASYwidth\@empty
    \ifx\ASYheight\@empty
      % write nothing!
    \else
      \immediate\write\AsyStream{size(0,\ASYheight,\asy@expand@keepAspect);}%
    \fi
  \else
    \ifx\ASYheight\@empty
      \immediate\write\AsyStream{size(\ASYwidth,0,\asy@expand@keepAspect);}%
    \else
      \immediate\write\AsyStream{size(\ASYwidth,\ASYheight,\asy@expand@keepAspect);}%
    \fi
  \fi
```

Setting `viewportsize=()`. Same logic as for `size()`. Note that `\ASYviewportwidth` varies according to `\ASYattach`, by default, so it needs to be fully expanded before checking if it equals zero.

```
  \edef\ASYviewportwidth{\ASYviewportwidth}% locally "freeze" viewportwidth
  \ifx\ASYviewportwidth\@empty
    \ifx\ASYviewportheight\@empty
      % write nothing!
    \else
      \immediate\write\AsyStream{viewportsize=(0,\ASYviewportheight);}%
    \fi
```

```
  \else
    \ifx\ASYviewportheight\@empty
      \immediate\write\AsyStream{viewportsize=(\ASYviewportwidth,0);}%
    \else
      \immediate\write\AsyStream{%
        viewportsize=(\ASYviewportwidth,\ASYviewportheight);}%
    \fi
  \fi
  \immediate\closeout\AsyStream
}
\def\asy@input@graphic{%
  \ifASYinline
    \IfFileExists{"\AsyFile.tex"}{%
      \catcode`:=12\relax
      \@@input"\AsyFile.tex"\relax
    }{%
      \PackageWarning{asymptote}{file `\AsyFile.tex' not found}%
    }%
  \else
    \IfFileExists{"\AsyFile.\AsyExtension"}{%
      \ifASYattach
        \ifASYPDF
          \IfFileExists{"\AsyFile+0.pdf"}{%
            \setbox\ASYbox=\hbox{\includegraphics[hiresbb]{"\AsyFile+0".pdf}}%
          }{%
            \setbox\ASYbox=\hbox{\includegraphics[hiresbb]{"\AsyFile".pdf}}%
          }%
        \else
          \setbox\ASYbox=\hbox{\includegraphics[hiresbb]{"\AsyFile.eps"}}%
        \fi
        \textattachfile{\AsyFile.\AsyExtension}{\phantom{\copy\ASYbox}}%
        \vskip-\ht\ASYbox
        \indent
        \box\ASYbox
      \else
        \ifASYPDF
          \includegraphics[hiresbb]{"\AsyFile".pdf}%
        \else
          \includegraphics[hiresbb]{"\AsyFile.eps"}%
        \fi
      \fi
    }{%
```

3D PRC figures require inline mode.

```
      \IfFileExists{"\AsyFile.tex"}{%
        \catcode`:=12
        \@@input"\AsyFile.tex"\relax
      }{%
        \PackageWarning{asymptote}{%
          file `\AsyFile.\AsyExtension' not found%
```

```
        }%
      }%
    }%
  \fi
}
\def\asydef{%
  \let\ThisAsymptote\WriteGlobalAsyLine
  \ProcessAsymptote{asydef}%
}
\newcommand\asyinclude[2][]{%
  \begingroup
  \stepcounter{asy}%
  \setkeys{ASYkeys}{#1}%
  \ifASYattach
    \ASYinlinefalse
  \fi
  \ifx\asydir\empty\else
    \def\ASYprefix{\asydir/}%
  \fi
  \immediate\write\AsyPreStream{%
    \noexpand\InputIfFileExists{%
      \ASYprefix\noexpand\jobname-\the\c@asy.pre}{}{}%
  }
  \asy@write@graphic@header
  \IfFileExists{#2.asy}{%
    \CatchFileDef\@tempa{#2.asy}{%
      \let\do\@makeother
      \dospecials
      \endlinechar=10\relax
    }
  }{
    \IfFileExists{#2}{%
      \CatchFileDef\@tempa{#2}{%
        \let\do\@makeother
        \dospecials
        \endlinechar=10\relax
      }
    }{
      \PackageWarning{asymptote}{file #2 not found}%
    }
  }
  \immediate\write\AsyStream{\unexpanded\expandafter{\@tempa}}%
  \asy@finalise@stream
  \asy@input@graphic
  \endgroup
}
\newcommand{\ASYanimategraphics}[5][]{%
  \IfFileExists{_#3.pdf}{%
    \animategraphics[{#1}]{#2}{_#3}{#4}{#5}%
```

```
  }{}
}
```

4.9 Keys for graphics processing

```
\newcommand\asysetup[1]{\setkeys{ASYkeys}{#1}}
\define@key{ASYkeys}{dir}{%
  \def\asydir{#1}%
}
\def\ASYwidth{}
\define@key{ASYkeys}{width}{%
  \edef\ASYwidth{\the\dimexpr#1\relax}%
}
\def\ASYheight{}
\define@key{ASYkeys}{height}{%
  \edef\ASYheight{\the\dimexpr#1\relax}%
}
\define@key{ASYkeys}{keepAspect}[true]{%
  \ifthenelse{\equal{#1}{true}}
    {\ASYkeepAspecttrue}
    {\ASYkeepAspectfalse}%
}
\def\ASYviewportwidth{\ASYdefaultviewportwidth}
\def\ASYdefaultviewportwidth{%
  \ifASYattach \@empty \else \the\linewidth \fi
}
\define@key{ASYkeys}{viewportwidth}{%
  \edef\ASYviewportwidth{\the\dimexpr#1\relax}%
}
\def\ASYviewportheight{}
\define@key{ASYkeys}{viewportheight}{%
  \edef\ASYviewportheight{\the\dimexpr#1\relax}%
}
\define@key{ASYkeys}{inline}[true]{%
  \ifthenelse{\equal{#1}{true}}
    {\ASYinlinetrue}
    {\ASYinlinefalse}%
}
\define@key{ASYkeys}{attach}[true]{%
  \ifthenelse{\equal{#1}{true}}
    {\ASYattachtrue}
    {\ASYattachfalse}%
}
```