

The `asymptote` package

John Bowman, Tom Prince, and Will Robertson

2026/08/22 v1.39

Abstract

This package provides integration of inline and external Asymptote graphics within a $\text{\LaTeX}$ document.

Contents

1 Introduction

This is the documentation for the $\text{\LaTeX}$ 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 $\text{\LaTeX}$ 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/`, then 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:

width Width of the figure

height Height of the figure

keepAspect Maintain aspect ratio [default true]

viewportwidth Viewport width for 3D figures

viewportheight Viewport height for 3D figures

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

3 Processing the document

After running $\text{\LaTeX}$ 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

```
1 \def\Asymptote{\tt Asymptote}
2 \InputIfFileExists{\jobname.pre}{\typeout{No file \jobname.pre.}}
```

4.1 Allocations

Allocations

```
3 \newbox\ASYbox
4 \newdimen\ASYdimen
5 \newcounter{asy}

6 \newwrite\AsyStream
7 \newwrite\AsyPreStream

8 \newif\ifASYinline
9 \newif\ifASYattach
10 \newif\ifASYkeepAspect
11 \ASYkeepAspecttrue
```

4.2 Packages

```
12 \RequirePackage{keyval}
13 \RequirePackage{ifthen}
14 \RequirePackage{graphicx}
```

Emulating packages We cannot assume that Asymptote users have recent $\text{\TeX}$ 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.

```
15 \IfFileExists{ifpdf.sty}{
16   \RequirePackage{ifpdf}
17 }{
18   \expandafter\newif\csname ifpdf\endcsname
19   \ifx\pdfoutput\@undefined\else
20     \ifcase\pdfoutput\else
21       \pdftrue
22     \fi
23   \fi
24 }

25 \IfFileExists{ifxetex.sty}{
26   \RequirePackage{ifxetex}
27 }{
28   \expandafter\newif\csname ifxetex\endcsname
29   \ifx\XeTeXversion\@undefined\else
30     \xetextrue
31   \fi
32 }
```

`\CatchFileDef` Used for `\asyinclude`. Note that the fallback definition is not as robust as the one provided by `catchfile`.

```
33 \IfFileExists{catchfile.sty}{
34   \RequirePackage{catchfile}
35 }
```

```

36 \newcommand\CatchFileDef[3]{%
37   \begingroup
38   \everyeof{%
39     \ENDCATCHFILEMARKER
40     \noexpand
41   }%
42   \long\def\@tempa####1\ENDCATCHFILEMARKER{%
43     \endgroup
44     \def##1{####1}%
45   }%
46   ##3%
47   \expandafter\@tempa\@@input ##2\relax
48 }
49 }

```

Ensuring attachfile2 is loaded if [attach] is requested

```

50 \newif\if@asy@attachfile@loaded

51 \AtBeginDocument{%
52   \ifpackageloaded{attachfile2}{\@asy@attachfile@loadedtrue}{}%
53   \let\asy@check@attachfile\asy@check@attachfile@loaded
54 }

55 \newcommand\asy@check@attachfile@loaded{%
56   \if@asy@attachfile@loaded\else
57     \PackageError{asyptote}{You must load the attachfile2 package}{^^J%
58       You have requested the [attach] option for some or all of your^^J%
59       Asymptote graphics, which requires the attachfile2 package.^^J%
60       Please load it in the document preamble.^^J%
61   }%
62   \fi
63 }

64 \newcommand\asy@check@attachfile{%
65   \AtBeginDocument{\asy@check@attachfile@loaded}%
66   \let\asy@check@attachfile\@empty
67 }

```

Macros

```

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

```

4.3 Package options

```

69 \DeclareOption{inline}{%
70   \ASYinlinetrue
71 }
72 \DeclareOption{attach}{%
73   \asy@check@attachfile
74   \ASYattachtrue
75 }
76 \ProcessOptions*

77 \def\asylatexdir{}
78 \def\asydir{}
79 \def\ASYasydir{}

```

```
80 \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.

```
81 \newif\ifASYPDF
82 \ifxetex
83   \ASYPDFtrue
84 \else
85   \ifpdf
86     \ASYPDFtrue
87   \fi
88 \fi
89 \ifASYPDF
90   \def\AsyExtension{pdf}
91 \else
92   \def\AsyExtension{eps}
93 \fi
```

4.5 Bug squashing

```
94 \def\unquoteJobname#1"#2"#3\relax{%
95   \def\rawJobname{#1}%
96   \ifx\rawJobname\empty
97     \def\rawJobname{#2}%
98   \fi
99 }
100 \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

```
101 \def\fixstar#1*#2\relax{%
102   \def\argtwo{#2}%
103   \ifx\argtwo\empty
104     \gdef\Jobname{#1}%
105   \else
106     \fixstar#1-#2\relax
107   \fi
108 }
109 \expandafter\fixstar\rawJobname*\relax
```

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

```
110 \def\Gininclude@eps#1{%
111   \message{<#1>}%
112   \bgroup
113   \def\@tempa{!}%
114   \dimen@\Gin@req@width
115   \dimen@ii.1bp\relax
116   \divide\dimen@\dimen@ii
117   \@tempdima\Gin@req@height
118   \divide\@tempdima\dimen@ii
119   \special{PSfile=#1\space
120     llx=\Gin@llx\space
121     lly=\Gin@lly\space
```

```

122     urx=\Gin@urx\space
123     ury=\Gin@ury\space
124     \ifx\Gin@scalex\@tempa\else rwi=\number\dimen@\space\fi
125     \ifx\Gin@scaley\@tempa\else rhi=\number\@tempdima\space\fi
126     \ifGin@clip clip\fi}%
127 \egroup
128 }

```

4.6 Input/Output

```

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

131 \def\WriteAsyLine#1{%
132   \immediate\write\AsyStream{\detokenize{#1}}}%
133 }

134 \def\globalASYdefs{}
135 \def\WriteGlobalAsyLine#1{%
136   \expandafter\g@addto@macro
137   \expandafter\globalASYdefs
138   \expandafter{\detokenize{#1^^J}}}%
139 }

```

4.7 Commands for verbatim processing environments

```

140 \def\ProcessAsymptote#1{%
141   \begingroup
142   \def\CurrentAsymptote{#1}%
143   \let\do\@makeother \dospecials
144   \@makeother^^L% and whatever other special cases
145   \catcode'\ =10
146   \endlinechar'\^^M \catcode'\^^M=12 \xAsymptote
147 }

```

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

```

148 \begingroup
149   \catcode'\^^M=12 \endlinechar=-1\relax%
150   \gdef\xAsymptote{%
151     \expandafter\ProcessAsymptoteLine%
152   }
153   \gdef\ProcessAsymptoteLine#1^^M{%
154     \def\@tempa{#1}%
155     {%
156       \escapechar=-1\relax%
157       \xdef\@tempb{\string\end\string\{\CurrentAsymptote\string\}}%
158     }%
159     \ifx\@tempa\@tempb%
160       \edef\next{\endgroup\noexpand\end{\CurrentAsymptote}}%
161     \else%
162       \ThisAsymptote{#1}%
163       \let\next\ProcessAsymptoteLine%
164     \fi%
165     \next%
166   }
167 \endgroup
168 \def\asy@init{%

```

```

169 \def\ASYlatexdir{}%
170 \ifx\asylatexdir\empty\else
171   \def\ASYlatexdir{\asylatexdir/}%
172 \fi
173 \ifx\asydir\empty\else
174   \def\ASYasydir{\asydir/}%
175 \fi
176 \def\ASYprefix{\ASYlatexdir\ASYasydir}%
177 }

```

4.8 User interface

```

178 \newcommand\asy[1][]{%
179   \stepcounter{asy}%
180   \setkeys{ASYkeys}{#1}%

```

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

```

181   \ifASYattach
182     \ASYinlinefalse
183   \fi
184   \asy@init
185   \immediate\write\AsyPreStream{%
186     \noexpand\InputIfFileExists{%
187       \ASYprefix\noexpand\jobname-\the\c@asy.pre}{-}{-}%
188   }%
189   \asy@write@graphic@header
190   \let\ThisAsymptote\WriteAsyLine
191   \ProcessAsymptote{asy}%
192 }

193 \def\endasy{%
194   \asy@finalise@stream
195   \asy@input@graphic
196 }

197 \def\asy@write@graphic@header{%
198   \immediate\openout\AsyStream=\ASYasydir\jobname-\the\c@asy.asy\relax
199   \gdef\AsyFile{\ASYprefix\Jobname-\the\c@asy}%
200   \immediate\write\AsyStream{%
201     if(!settings.multipleView) settings.batchView=false;^^J%
202     \ifxetex
203       settings.tex="xelatex";^^J%
204     \else\ifASYPDF
205       settings.tex="pdflatex";^^J%
206     \fi\fi
207     \ifASYinline
208       settings.inlinetex=true;^^J%
209       deletepreamble();^^J%
210     \fi
211     defaultfilename="\Jobname-\the\c@asy";^^J%
212     settings.outformat="";^^J%
213     \ifASYattach
214       settings.inlineimage=false;^^J%
215       settings.embed=false;^^J%
216       settings.toolbar=true;^^J%
217     \else

```

```

218     settings.inlineimage=true;^^J%
219     settings.embed=true;^^J%
220     settings.toolbar=false;^^J%
221     viewportmargin=(2,2);^^J%
222 \fi
223 \globalASYdefs
224 }%
225 }
226 \def\asy@expand@keepAspect{%
227 \ifASYkeepAspect keepAspect=true%
228 \else keepAspect=false%
229 \fi%
230 }
231 \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).
232 \ifx\ASYwidth\@empty
233 \ifx\ASYheight\@empty
234 % write nothing!
235 \else
236 \immediate\write\AsyStream{size(0,\ASYheight,\asy@expand@keepAspect);} %
237 \fi
238 \else
239 \ifx\ASYheight\@empty
240 \immediate\write\AsyStream{size(\ASYwidth,0,\asy@expand@keepAspect);} %
241 \else
242 \immediate\write\AsyStream{size(\ASYwidth,\ASYheight,\asy@expand@keepAspect);} %
243 \fi
244 \fi
    Setting viewportsize(). Same logic as for size().
245 \ifx\ASYviewportwidth\@empty
246 \ifx\ASYviewportheight\@empty
247 % write nothing!
248 \else
249 \immediate\write\AsyStream{viewportsize=(0,\ASYviewportheight);} %
250 \fi
251 \else
252 \ifx\ASYviewportheight\@empty
253 \immediate\write\AsyStream{viewportsize=(\ASYviewportwidth,0);} %
254 \else
255 \immediate\write\AsyStream{%
256 viewportsize=(\ASYviewportwidth,\ASYviewportheight);} %
257 \fi
258 \fi
259 \immediate\closeout\AsyStream
260 }
261 \def\asy@input@graphic{%
262 \ifASYinline
263 \IfFileExists{"\AsyFile.tex"}{%
264 \catcode'\:=12\relax
265 \@input"\AsyFile.tex"\relax
266 }{%
267 \PackageWarning{asymptote}{file '\AsyFile.tex' not found}%

```

```

268 }%
269 \else
270 \IfFileExists{"\AsyFile.\AsyExtension"}{%
271 \ifASYattach
272 \ifASYPDF
273 \IfFileExists{"\AsyFile+0.pdf"}{%
274 \setbox\ASYbox=\hbox{\includegraphics[hiresbb]{\AsyFile+0.pdf}}%
275 }{%
276 \setbox\ASYbox=\hbox{\includegraphics[hiresbb]{\AsyFile.pdf}}%
277 }%
278 \else
279 \setbox\ASYbox=\hbox{\includegraphics[hiresbb]{\AsyFile.eps}}%
280 \fi
281 \textattachfile{\AsyFile.\AsyExtension}{\phantom{\copy\ASYbox}}%
282 \vskip-\ht\ASYbox
283 \indent
284 \box\ASYbox
285 \else
286 \ifASYPDF
287 \includegraphics[hiresbb]{\AsyFile.pdf}%
288 \else
289 \includegraphics[hiresbb]{\AsyFile.eps}%
290 \fi
291 \fi
292 }{%

3D PRC figures require inline mode.
293 \IfFileExists{"\AsyFile.tex"}{%
294 \catcode'\:=12
295 \@input"\AsyFile.tex"\relax
296 }{%
297 \PackageWarning{asytote}{%
298 file '\AsyFile.\AsyExtension' not found%
299 }%
300 }%
301 }%
302 \fi
303 }

304 \def\asydef{%
305 \let\ThisAsymptote\WriteGlobalAsyLine
306 \ProcessAsymptote{asydef}%
307 }

308 \newcommand\asyinclude[2][]{%
309 \begingroup
310 \stepcounter{asy}%
311 \setkeys{ASYkeys}{#1}%
312 \ifASYattach
313 \ASYinlinefalse
314 \fi
315 \asy@init
316 \immediate\write\AsyPreStream{%
317 \noexpand\input\IfFileExists{%
318 \ASYprefix\noexpand\jobname-\the\c@asy.pre}{-}{-}%
319 }%

```

```

320 \asy@write@graphic@header
321 \IfFileExists{#2.asy}{%
322   \CatchFileDef\@tempa{#2.asy}{%
323     \let\do\@makeoother
324     \dospecials
325     \endlinechar=10\relax
326   }%
327 }{%
328   \IfFileExists{#2}{%
329     \CatchFileDef\@tempa{#2}{%
330       \let\do\@makeoother
331       \dospecials
332       \endlinechar=10\relax
333     }%
334   }{%
335     \PackageWarning{asymptote}{file #2 not found}%
336     \def\@tempa{}%
337   }%
338 }%
339 \immediate\write\AsyStream{\unexpanded\expandafter{\@tempa}}%
340 \asy@finalise@stream
341 \asy@input@graphic
342 \endgroup
343 }

344 \newcommand{\ASYanimategraphics}[5][{}]{%
345   \IfFileExists{_#3.pdf}{%
346     \animategraphics[{#1}]{#2}{_#3}{#4}{#5}%
347   }{}%
348 }

```

4.9 Keys for graphics processing

```

349 \newcommand\asysetup[1]{\setkeys{ASYkeys}{#1}}

350 \define@key{ASYkeys}{dir}{%
351   \def\asydir{#1}%
352 }
353 \def\ASYwidth{}
354 \define@key{ASYkeys}{width}{%
355   \edef\ASYwidth{\the\dimexpr#1\relax}%
356 }
357 \def\ASYheight{}
358 \define@key{ASYkeys}{height}{%
359   \edef\ASYheight{\the\dimexpr#1\relax}%
360 }
361 \define@key{ASYkeys}{keepAspect}[true]{%
362   \ifthenelse{equal{#1}{true}}
363     {\ASYkeepAspecttrue}
364     {\ASYkeepAspectfalse}%
365 }
366 \def\ASYviewportwidth{}
367 \define@key{ASYkeys}{viewportwidth}{%
368   \edef\ASYviewportwidth{\the\dimexpr#1\relax}%
369 }
370 \def\ASYviewportheight{}

```

```

371 \define@key{ASYkeys}{viewportheight}{%
372   \edef\ASYviewportheight{\the\dimexpr#1\relax}%
373 }

374 \define@key{ASYkeys}{inline}[true]{%
375   \ifthenelse{\equal{#1}{true}}
376     {\ASYinlinetrue}
377     {\ASYinlinefalse}%
378 }

379 \define@key{ASYkeys}{attach}[true]{%
380   \ifthenelse{\equal{#1}{true}}
381     {\ASYattachtrue}
382     {\ASYattachfalse}%
383 }

```