

The mgl \TeX package*

Diego Sejas Viscarra
diego.mathematician@gmail.com

October 15, 2014

Abstract

1 Introduction

MathGL is a fast and efficient library by Alexey Balakin for the creation of high-quality publication-ready scientific graphics. It implements more than 50 different types of graphics for 1d, 2d and 3d large sets of data. It supports exporting images to bitmap formats (PNG, JPEG, BMP, etc.), or vector formats (EPS, \TeX , SVG, etc.), or 3d image formats (STL, OBJ, XYZ, etc.), and even its own 3d format, MGLD. MathGL also defines its own vector font specification format, and supports UTF-16 encoding with \TeX -like symbol parsing. It supports various kinds of transparency and lighting, textual formula evaluation, arbitrary curvilinear coordinate systems, loading of subroutines from .dll or .so libraries, and many other useful features.

MathGL has interfaces for a wide variety of programming languages, such as C/C++, Fortran, Python, Octave, Pascal, Forth, and many others, but it also defines its own scripting language, called *MGL*, which can be used to generate graphics independently of any programming language. The mgl \TeX package adds support to embed MGL code inside \LaTeX documents, which is automatically extracted and executed, and the resulting images are included in the document.

Besides the obvious advantage of having available all the useful features of MathGL, mgl \TeX facilitates the maintenance of your document, since both code for text and code for graphics are contained in a single file.

2 Usage

The simplest way to load mgl \TeX to a \LaTeX document is to write the command

```
\usepackage{mgltex}
```

*This document corresponds to mgl \TeX v.1.0, dated 2014/09/16.

in the preamble. Alternatively, one can pass a number of options to the package by means of the syntax

$$\backslash\text{usepackage}[\langle\text{options list}\rangle]\{\text{mgl}\text{tex}\},$$

where $\langle\text{options list}\rangle$ is a comma-separated list that can contains one or more of the following options:

- **draft**: The generated images won't be included in the document. This option is useful when fast compilation of the document is needed.
- **final**: This overrides the **draft** option.
- **png**, **jpg**, **jpeg**: To export images to the corresponding bitmap format.
- **eps**, **epsz**: To export to uncompressed/compressed EPS format as primitives.
- **bps**, **bpsz**: To export to uncompressed/compressed EPS format as bitmap.
- **pdf**: To export to 3D PDF format.

It must be noted that the options that specify the format to save the images are exclusive, in the sense that if one specifies more than one format, only the last one will be used.

There are two ways to compile a document with $\text{mgl}\text{T}\text{E}\text{X}$: The first way is to run

$$\text{latex} \text{ --shell-escape } \langle\text{document}\rangle$$

twice, since the first run will extract the MGL code, execute it and include some of the resulting graphics, while the second run will include the remaining graphics; the second way is to run $\text{latex} \langle\text{document}\rangle$ to extract the MGL code, then execute the generated scripts with the program mglconv (which comes with MathGL), and execute $\text{latex} \langle\text{document}\rangle$ once more to include the graphics.

2.1 Environments for MGL code embedding

mgl The main environment defined by $\text{mgl}\text{T}\text{E}\text{X}$ is mgl . It extracts its contents to a general script, called $\langle\text{document}\rangle.\text{mgl}$, where $\langle\text{document}\rangle$ stands for the name of the $\text{L}\text{A}\text{T}\text{E}\text{X}$ file being compiled; this script is compiled, and the corresponding image is included. Its syntax is:

$$\begin{array}{c} \backslash\text{begin}\{\text{mgl}\}[\langle\text{key-val list}\rangle] \\ \langle\text{MGL code}\rangle \\ \backslash\text{end}\{\text{mgl}\} \end{array}$$

Here, $\langle\text{key-val list}\rangle$ accepts the same optional arguments as the $\backslash\text{includegraphics}$ command from the **graphicx** package, plus an additional one, **imgext**, which can be used to specify the extension to save the graphic. The $\langle\text{MGL code}\rangle$ doesn't

need to contain any specific instruction to create the image, since `mg|TeX` takes care of that.

mg|laddon This environment adds its contents to the general script `<document>.mgl`, but it doesn't produce any image. It doesn't require any kind of arguments.

```
\begin{mg|laddon}
  <MGL code>
\end{mg|laddon}
```

mg|lcode This is the same as the `mg|l` environment, but the corresponding code is written *verbatim* to a separate script, whose name is specified as mandatory argument. It accepts the same optional arguments as `mg|l`.

```
\begin{mg|lcode}[<key-val list>]{<script_name>}
  <MGL code>
\end{mg|lcode}
```

mg|lscript The code within `mg|lscript` is written *verbatim* to a script whose name is specified as mandatory argument, but no image is produced. It is useful for creation of MGL scripts which can be later post-processed by another package, like `listings`.

```
\begin{mg|lscript}{<script_name>}
  <MGL code>
\end{mg|lscript}
```

mg|lfunc This is used to define MGL functions within the general script `<document>.mgl`. It takes one mandatory argument, which is the name of the function, plus one optional argument, which specifies the number of arguments of the function. The environment needs to contain only the body of the function, since the lines “`func <function_name> <number of arguments>`” and “`return`” are appended automatically at the beginning and the end, respectively. The resulting code is written at the end of the general script, after the `stop` command, which is also written automatically.

```
\begin{mg|lfunc}[<number of arguments>]{<function_name>}
  <MGL function body>
\end{mg|lfunc}
```

2.2 Fast creation of graphics

`mg|TeX` defines a convenient way to work with many graphics that have exactly the same settings (for example, same angles of rotation, same type of grid, etc.):

instead of writing repetitive code every time it's needed, it can be stored in memory with the `mglplotsettings` environment, and then can be used when needed with the `\mglplot` command.

`mglplotsettings` This environment stores its contents in memory for later use. It accepts one mandatory argument, which is a keyword (name) to be associated to the corresponding block of code, so different blocks of code can be stored with different names.

```
\begin{mglplotsettings}{\keyword}
    \langle MGL code \rangle
\end{mglplotsettings}
```

`\mglplot` This command is used for fast generation of graphics with default settings, and can be used in parallel with the `mglplotsettings` environment. It accepts one mandatory argument which consists of MGL instructions, separated by the symbol “:”, which can span through various text lines. It also accepts the same optional arguments as the `mgl` environment, plus an additional one, called `settings`, which can be used to specify a keyword used in a `mglplotsettings` environment. If the `settings` option is specified, the code in the mandatory argument will be appended to the block of code of the corresponding `mglplotsettings` environment.

```
\mglplot[\langle key-val list \rangle]{\langle MGL code \rangle}
```

2.3 Verbatim-like environments

`mglblock` It writes its contents *verbatim* to a file, whose name is given as mandatory argument, and then it also typesets its contents on the \LaTeX document, numbering each line of code.

```
\begin{mglblock}{\script_name}
    \langle MGL code \rangle
\end{mglblock}
```

`mglverbatim` It typesets its contents to the \LaTeX document, numbering each line of code.

```
\begin{mglverbatim}
    \langle MGL code \rangle
\end{mglverbatim}
```

2.4 Working with external scripts

In case of having MGL scripts in their own files, $\text{mgl}\text{\TeX}$ can work with them without needing to transcript them to the \LaTeX document.

`\mglgraphics` This command takes one mandatory argument, which is the name of an exter-

nal MGL script, which will be automatically executed, and the resulting image will be included. The same optional arguments as the `mgl` environment are accepted.

$$\backslash\mathrm{mglgraphics}[\langle\textit{key-val list}\rangle](\langle\textit{script_name}\rangle)$$

`\mglinclude` This command takes one mandatory argument, which is the name of an external MGL script, which will be automatically transcript *verbatim* on the $\mathrm{\LaTeX}$ document, and each line of code will be numerated.

$$\backslash\mathrm{mglinclude}\{\langle\textit{script_name}\rangle\}$$

2.5 Additional commands

`\mgldir` This command can be used to specify where `mglTeX` should create the MGL scripts and corresponding images. This is useful, for example, to avoid a lot of scripts and images from polluting the current directory.

$$\backslash\mathrm{mgldir}\{\langle\textit{directory}\rangle\}$$

This command must be used in the preamble of the document, since the first MGL script is created at the moment of the `\begin{document}` command; trying to use it somewhere else will issue an error. On the other hand, it is the responsibility of the user to create the $\langle\textit{directory}\rangle$, since `mglTeX` won't do it automatically.

`\mglTeX` This command just pretty-prints the name of the package.

$$\backslash\mathrm{mglTeX}$$

2.6 Behavior of `mglTeX`

As a convenient feature, the environments `mglcode`, `mglscript` and `mglblock` will automatically check if they are being used to create different scripts with the same name, in which case `mglTeX` will issue a warning; however, if one of these environments overwrite an external script (not embedded in the document), it won't be noticed. Likewise, the user will be warned if the environment `mglfunc` is being used to create different MGL functions with the same name.

When `mglTeX` is unable to find a graphic that is supposed to include, instead of producing an error, it will warn the user about it, and will display a box in the corresponding position of the document, like the following one:

**MGL
image
not
found**

Notice that the first time L^AT_EX is executed, many of these boxes will appear in the document because the graphics from the MGL scripts are created, but not all are included (until L^AT_EX is run for the second time).

3 Implementation

```

1
2 \RequirePackage{keyval}
3 \RequirePackage{graphicx}
4
5 \DeclareOption{draft}{%
6   \PassOptionsToPackage{\CurrentOption}{graphicx}%
7 }
8 \DeclareOption{final}{%
9   \PassOptionsToPackage{\CurrentOption}{graphicx}%
10 }
11
12 \DeclareGraphicsExtensions{.png,.eps,.jpg,.jpeg,.bps,.pdf,.epsz,.eps.gz,.bpsz,.bps.gz,.gif}
13 %\DeclareGraphicsExtensions{.png,.eps,.jpg,.jpeg,.bps,.pdf,.gif}
14 %\DeclareGraphicsRule{*}{eps}{*}{}
15 %\DeclareGraphicsRule{.eps.gz}{eps}{.eps.bb}{'gunzip -c #1}%   gzipped EPS
16 %\DeclareGraphicsRule{.epsz}{eps}{.eps.bb}{'gunzip -c #1}%   gzipped EPS
17 %\DeclareGraphicsRule{.bps.gz}{eps}{.bps.bb}{'gunzip -c #1}%   gzipped EPS
18 %\DeclareGraphicsRule{.bpsz}{eps}{.bps.bb}{'gunzip -c #1}%   gzipped EPS
19
20 \DeclareOption{jpg}{\def\mgl@image@ext{.jpg}}
21 \DeclareOption{jpeg}{\def\mgl@image@ext{.jpeg}}
22 \DeclareOption{pdf}{\def\mgl@image@ext{.pdf}}
23 \DeclareOption{png}{\def\mgl@image@ext{.png}}
24 \DeclareOption{eps}{\def\mgl@image@ext{.eps}}
25 \DeclareOption{epsz}{\def\mgl@image@ext{.eps.gz}}
26 \DeclareOption{bps}{\def\mgl@image@ext{.bps}}
27 \DeclareOption{bpsz}{\def\mgl@image@ext{.bps.gz}}
28 \DeclareOption{gif}{\def\mgl@image@ext{.gif}}
29
30 \DeclareOption{tex}{\def\mgl@image@ext{.tex}}
31 %\DeclareOption{svg}{\def\mgl@image@ext{.svg}}
32 %\DeclareOption{svgz}{\def\mgl@image@ext{.svgz}}
33 %\DeclareOption{bmp}{\def\mgl@image@ext{.bmp}}
34 %\DeclareOption{tga}{\def\mgl@image@ext{.tga}}
35 %\DeclareOption{mgl}{\def\mgl@image@ext{.mgl}}
36 %\DeclareOption{json}{\def\mgl@image@ext{.json}}
37 %\DeclareOption{jsonz}{\def\mgl@image@ext{.jsonz}}
38 %\DeclareOption{obj}{\def\mgl@image@ext{.obj}}
39 %\DeclareOption{xyz}{\def\mgl@image@ext{.xyz}}
40 %\DeclareOption{stl}{\def\mgl@image@ext{.stl}}
41 %\DeclareOption{off}{\def\mgl@image@ext{.off}}
42 %\DeclareOption{prc}{\def\mgl@image@ext{.prc}}
43

```

```

44 \ExecuteOptions{final,eps}
45 \ProcessOptions*
46
47 \define@key{mgl@keys}{bb}{\g@addto@macro{\graph@keys}{bb=#1,}}
48 \define@key{mgl@keys}{bblx}{\g@addto@macro{\graph@keys}{bblx=#1,}}
49 \define@key{mgl@keys}{bblly}{\g@addto@macro{\graph@keys}{bblly=#1,}}
50 \define@key{mgl@keys}{bburx}{\g@addto@macro{\graph@keys}{bburx=#1,}}
51 \define@key{mgl@keys}{bbury}{\g@addto@macro{\graph@keys}{bbury=#1,}}
52 \define@key{mgl@keys}{natwidth}{\g@addto@macro{\graph@keys}{natwidth=#1,}}
53 \define@key{mgl@keys}{natheight}{\g@addto@macro{\graph@keys}{natheight=#1,}}
54 \define@key{mgl@keys}{hiresbb}{\g@addto@macro{\graph@keys}{hiresbb=#1,}}
55 \define@key{mgl@keys}{viewport}{\g@addto@macro{\graph@keys}{viewport=#1,}}
56 \define@key{mgl@keys}{trim}{\g@addto@macro{\graph@keys}{trim=#1,}}
57 \define@key{mgl@keys}{angle}{\g@addto@macro{\graph@keys}{angle=#1,}}
58 \define@key{mgl@keys}{origin}{\g@addto@macro{\graph@keys}{origin=#1,}}
59 \define@key{mgl@keys}{width}{\g@addto@macro{\graph@keys}{width=#1,}}
60 \define@key{mgl@keys}{height}{\g@addto@macro{\graph@keys}{height=#1,}}
61 \define@key{mgl@keys}{totalheight}{\g@addto@macro{\graph@keys}{totalheight=#1,}}
62 \define@key{mgl@keys}{keepaspectratio}{\g@addto@macro{\graph@keys}{keepaspectratio=#1,}}
63 \define@key{mgl@keys}{scale}{\g@addto@macro{\graph@keys}{scale=#1,}}
64 \define@key{mgl@keys}{clip}[true]{\g@addto@macro{\graph@keys}{clip=#1,}}
65 \define@key{mgl@keys}{draft}[false]{\g@addto@macro{\graph@keys}{draft=#1,}}
66 \define@key{mgl@keys}{type}{\g@addto@macro{\graph@keys}{type=#1,}}
67 \define@key{mgl@keys}{ext}{\g@addto@macro{\graph@keys}{ext=#1,}}
68 \define@key{mgl@keys}{read}{\g@addto@macro{\graph@keys}{read=#1,}}
69 \define@key{mgl@keys}{command}{\g@addto@macro{\graph@keys}{command=#1,}}
70 \define@key{mgl@keys}{imgext}{\def\mgl@image@ext{.#1}}
71
72 \define@key{mglplot@keys}{bb}{\g@addto@macro{\graph@keys}{bb=#1,}}
73 \define@key{mglplot@keys}{bblx}{\g@addto@macro{\graph@keys}{bblx=#1,}}
74 \define@key{mglplot@keys}{bblly}{\g@addto@macro{\graph@keys}{bblly=#1,}}
75 \define@key{mglplot@keys}{bburx}{\g@addto@macro{\graph@keys}{bburx=#1,}}
76 \define@key{mglplot@keys}{bbury}{\g@addto@macro{\graph@keys}{bbury=#1,}}
77 \define@key{mglplot@keys}{natwidth}{\g@addto@macro{\graph@keys}{natwidth=#1,}}
78 \define@key{mglplot@keys}{natheight}{\g@addto@macro{\graph@keys}{natheight=#1,}}
79 \define@key{mglplot@keys}{hiresbb}{\g@addto@macro{\graph@keys}{hiresbb=#1,}}
80 \define@key{mglplot@keys}{viewport}{\g@addto@macro{\graph@keys}{viewport=#1,}}
81 \define@key{mglplot@keys}{trim}{\g@addto@macro{\graph@keys}{trim=#1,}}
82 \define@key{mglplot@keys}{angle}{\g@addto@macro{\graph@keys}{angle=#1,}}
83 \define@key{mglplot@keys}{origin}{\g@addto@macro{\graph@keys}{origin=#1,}}
84 \define@key{mglplot@keys}{width}{\g@addto@macro{\graph@keys}{width=#1,}}
85 \define@key{mglplot@keys}{height}{\g@addto@macro{\graph@keys}{height=#1,}}
86 \define@key{mglplot@keys}{totalheight}{\g@addto@macro{\graph@keys}{totalheight=#1,}}
87 \define@key{mglplot@keys}{keepaspectratio}{\g@addto@macro{\graph@keys}{keepaspectratio=#1,}}
88 \define@key{mglplot@keys}{scale}{\g@addto@macro{\graph@keys}{scale=#1,}}
89 \define@key{mglplot@keys}{clip}[true]{\g@addto@macro{\graph@keys}{clip=#1,}}
90 \define@key{mglplot@keys}{draft}[false]{\g@addto@macro{\graph@keys}{draft=#1,}}
91 \define@key{mglplot@keys}{type}{\g@addto@macro{\graph@keys}{type=#1,}}
92 \define@key{mglplot@keys}{ext}{\g@addto@macro{\graph@keys}{ext=#1,}}
93 \define@key{mglplot@keys}{read}{\g@addto@macro{\graph@keys}{read=#1,}}

```

```

94 \define@key{mglplot@keys}{command}{\g@addto@macro{\graph@keys}{command=#1,}}
95 \define@key{mglplot@keys}{outext}{\def\mglplot@output@ext{. #1}}
96 \define@key{mglplot@keys}{settings}{\def\mglplot@settings{#1}}
97
98 \def\mgl@dir{}
99 \def\mgldir#1{%
100   \def\mgl@dir{#1}%
101 }
102 \@onlypreamble\mgldir
103
104 \newwrite\mgl@script
105 \AtBeginDocument{%
106   \immediate\openout\mgl@script="\mgl@dir\jobname.mgl"%
107 }
108 \AtEndDocument{%
109   \immediate\write\mgl@script{}%
110   \immediate\write\mgl@script{stop}%
111   \mgl@func%
112   \immediate\closeout\mgl@script%
113   \immediate\write18{mglconv -n "\mgl@dir\jobname.mgl"}%
114 }
115
116 \def\mglplotsettings@defined{}
117 \newcommand\mglplotsettings[1][generic]{%
118   \test@mglplotsettings@defined{#1}%
119   \expandafter\def\csname mgl@setup@#1\endcsname{\immediate\write\mgl@script{}}%
120   \g@addto@macro{\mglplotsettings@defined}{#1,}%
121   \let\do\@makeother \dospecials%
122   \endlinechar'\^^M \catcode'\^^M\active%
123   \catcode'\ =10%
124   \expandafter\mglplotsettings@write@line%
125 }
126 \def\test@mglplotsettings@defined#1{%
127   \def\this@setup{#1}%
128   \@for\mglplotsettings@name:=\mglplotsettings@defined\do{%
129     \ifx\this@mglplotsettings\mglplotsettings@name%
130       \PackageWarning{\mgl@name}{Redefining "#1" setup for \noexpand\mglplot}%
131     \fi%
132   }%
133 }
134 \begingroup%
135   \catcode'\^^M\active%
136   \gdef\mglplotsettings@write@line#1^^M{%
137     \def\next@action{%
138       \expandafter\g@addto@macro\csname mgl@setup@\this@setup\endcsname{\immediate\write\mgl@script{}}%
139       \mglplotsettings@write@line%
140     }%
141     \test@end@mglplotsettings{#1}%
142     \next@action%
143   }%

```



```

144 \endgroup
145 \begingroup%
146 \escapechar=-1 \relax%
147 \xdef\end@mglplotsettings{\string\\end\string\{mglplotsettings\string\}}%
148 \endgroup
149 \def\test@end@mglplotsettings#1{%
150 \edef\this@line{#1}%
151 \ifx\this@line\end@mglplotsettings%
152 \def\next@action{\end{mglplotsettings}}%
153 \fi%
154 }
155 \def\endmglplotsettings{}
156
157 \def\TeX@ext{.tex}
158 \newcounter{mgl@image@no}
159
160 \def\mglplot{%
161 \@ifnextchar[{\@mglplot}{\@mglplot[]}%
162 }
163 \def\@mglplot[#1]{%
164 \def\mglplot@settings{generic}%
165 \def\graph@keys{}%
166 \setkeys{mglplot@keys}{#1}%
167 \stepcounter{mgl@image@no}%
168 \ifx\csname mgl@setup@mglplot@settings\endcsname\undefined%
169 \PackageError{mgl@name}{Setup "\mglplot@settings" undefined}{}%
170 \else%
171 \csname mgl@setup@mglplot@settings\endcsname%
172 \fi%
173 \@@mglplot%
174 }
175 \long\def\@@mglplot#1{%
176 \immediate\write\mgl@script{\detokenize{#1}}%
177 \immediate\write\mgl@script{write '\mgl@dir\jobname-mgl-\arabic{mgl@image@no}\mgl@image@ext'}%
178 \immediate\write\mgl@script{reset}%
179 \mgl@include@image{\mgl@dir\jobname-mgl-\arabic{mgl@image@no}}%
180 }
181
182 \newcommand\mgl[1][]{%
183 \def\graph@keys{}%
184 \setkeys{mgl@keys}{#1}%
185 \let\do\@makeother \dospecials%
186 \endlinechar'\^^M \catcode'\^^M\active%
187 \catcode'\ =10%
188 \mgl@write@line%
189 }
190 \begingroup%
191 \escapechar=-1 \relax%
192 \xdef\end@mgl{\string\\end\string\{mgl\string\}}%
193 \endgroup

```

```

194 \begingroup%
195   \catcode'\^^M\active%
196   \gdef\mgl@write@line#1^^M{%
197     \def\next@action{%
198       \immediate\write\mgl@script{#1}%
199       \mgl@write@line%
200     }%
201     \test@end@mgl{#1}%
202     \next@action%
203   }%
204 \endgroup
205 \def\test@end@mgl#1{%
206   \edef\this@line{#1}%
207   \ifx\this@line\end@mgl%
208     \def\next@action{\end{mgl}}%
209   \fi%
210 }
211 \def\endmgl{%
212   \stepcounter{mgl@image@no}%
213   \immediate\write\mgl@script{write '\mgl@dir\jobname-mgl-\arabic{mgl@image@no}\mgl@image@ext'}%
214   \immediate\write\mgl@script{reset}%
215   \mgl@include@image{\mgl@dir\jobname-mgl-\arabic{mgl@image@no}}%
216 }
217 \def\mgl@include@image#1{%
218   \ifx\mgl@image@ext\TeX@ext%
219     \IfFileExists{#1.tex}{%
220       \include{#1}%
221     }{%
222       \mgl@img@not@found{#1}%
223     }%
224   \else%
225     \def\next@action{\mgl@img@not@found{#1}}%
226     \@for\img@ext:=\Gin@extensions\do{%
227       \IfFileExists{#1\img@ext}{%
228         \def\next@action{%
229           \expandafter\includegraphics\expandafter[\graph@keys]{#1}%
230         }%
231       }{%
232         }%
233       \next@action%
234     \fi%
235   }
236 \def\mgl@img@not@found#1{%
237   \PackageWarning{\mgl@name}{MGL image "#1" not found}%
238   \framebox[10em]{%
239     \centering%
240     \bfseries\Huge%
241     \vbox{MGL\\image\\not\\found}%
242   }%
243 }

```

```

244
245 \bgroup%
246 \escapechar=-1\relax%
247 \xdef\end@mgladdon{\string\\end\string\{mgladdon\string\\}}%
248 \egroup%
249 \newenvironment{mgladdon}{%
250 \def\test@end@mgl##1{%
251 \edef\this@line{##1}%
252 \ifx\this@line\end@mgladdon%
253 \def\next@action{\end{mgladdon}}%
254 \fi%
255 }%
256 \mgl []%
257 }{}
258
259 \def\mgl@script@written{}
260 \newwrite\mgl@out@stream
261 \newcommand\mglcode[2] [] {%
262 \def\graph@keys{}%
263 \setkeys{mgl@keys}{#1}%
264 \test@mgl@script@written{#2}%
265 \xdef\mgl@script@written{\mgl@script@written#2,}%
266 \def\this@script{#2}%
267 \immediate\openout\mgl@out@stream=\mgl@dir\this@script.mgl%
268 \let\do\@makeother \dospecials%
269 \endlinechar'\^M \catcode'\^M\active%
270 \obeyspaces%
271 \mglcode@write@line%
272 }
273 \def\test@mgl@script@written#1{%
274 \edef\this@script{#1}%
275 \@for\mgl@script@name:=\mgl@script@written\do{%
276 \ifx\this@script\mgl@script@name%
277 \PackageWarning{mgl@name}{Overwriting MGL script "\this@script.mgl"}%
278 \fi%
279 }%
280 }
281 \def\mglcode@write@line#1{%
282 \let\next@action\mglcode@write@line%
283 \expandafter\if#1\^M%
284 \immediate\write\mgl@out@stream{\the\mgl@line}%
285 \mgl@line{}%
286 \mgl@word{}%
287 \else\expandafter\if#1\space%
288 \mgl@word{}%
289 \mgl@line\expandafter{\the\mgl@line#1}%
290 \else%
291 \mgl@word\expandafter{\the\mgl@word#1}%
292 \mgl@line\expandafter{\the\mgl@line#1}%
293 \test@end@mglcode{\the\mgl@word}%

```

```

294 \fi\fi%
295 \next@action%
296 }
297 \begingroup%
298 \escapechar=-1\relax%
299 \xdef\end@mglcode{\string\\end\string\{mglcode\string\}}%
300 \endgroup%
301 \def\test@end@mglcode#1{%
302 \edef\this@word{#1}%
303 \ifx\this@word\end@mglcode%
304 \def\next@action{\end{mglcode}}%
305 \fi%
306 }
307 \def\endmglcode{%
308 \immediate\closeout@mgl@out@stream%
309 \immediate\write18{mglconv "\mgl@dir\this@script.mgl" -o "\mgl@dir\this@script@mgl@image@ext"
310 \mgl@include@image{\mgl@dir\this@script}%
311 }
312
313 \bgroup%
314 \escapechar=-1\relax%
315 \xdef\end@mglscript{\string\\end\string\{mglscript\string\}}%
316 \egroup%
317 \newenvironment{mglscript}[1]{%
318 \def\test@end@mglcode##1{%
319 \edef\this@word{##1}%
320 \ifx\this@word\end@mglscript%
321 \def\next@action{\end{mglscript}}%
322 \fi%
323 }%
324 \mglcode{#1}%
325 }{%
326 \immediate\closeout@mgl@out@stream%
327 }
328
329 \def\mglfunc@defined{}
330 \def\mgl@func{}
331 \newcommand\mglfunc[2][0]{%
332 \test@mglfunc@defined{#2}%
333 \g@addto@macro{\mglfunc@defined}{#2,}%
334 \let\do\@makeother \dospecials%
335 \endlinechar'\^^M \catcode'\^^M\active%
336 \catcode'\ =10%
337 \g@addto@macro{\mgl@func}{\immediate\write\mgl@script{}}%
338 \g@addto@macro{\mgl@func}{\immediate\write\mgl@script{func '#2' #1}}%
339 \expandafter\mglfunc@ignore@line%
340 }
341 \def\test@mglfunc@defined#1{%
342 \def\this@func{#1}%
343 \@for\mglfunc@name:=\mglfunc@defined\do{%

```

```

344 \ifx\this@func\mglfunc@name%
345 \PackageWarning{\mgl@name}{MGL function "#1" has multiple definitions}%
346 \fi%
347 }%
348 }
349 \begingroup%
350 \catcode'\^^M\active%
351 \gdef\mglfunc@ignore@line#1^^M{%
352 \expandafter\mglfunc@write@line%
353 }
354 \gdef\mglfunc@write@line#1^^M{%
355 \def\next@action{%
356 \g@addto@macro{\mgl@func}{\immediate\write\mgl@script{#1}}%
357 \mglfunc@write@line%
358 }%
359 \test@end@mglfunc{#1}%
360 \next@action%
361 }%
362 \endgroup
363 \begingroup%
364 \escapechar=-1 \relax%
365 \xdef\end@mglfunc{\string\end\string\{mglfunc\string\}}%
366 \endgroup
367 \def\test@end@mglfunc#1{%
368 \edef\this@line{#1}%
369 \ifx\this@line\end@mglfunc%
370 \def\next@action{\end{mglfunc}}%
371 \fi%
372 }
373 \def\endmglfunc{%
374 \g@addto@macro{\mgl@func}{\immediate\write\mgl@script{return}}%
375 }
376
377 \def\mglblock#1{%
378 \test@mgl@script@written{#1}%
379 \xdef\mgl@script@written{\mgl@script@written#1,}%
380 \def\this@script{#1}%
381 \let\do\@makeother \dospecials%
382 \endlinechar'\^^M \catcode'\^^M\active%
383 \obeyspaces%
384 \immediate\openout\mgl@out@stream="\mgl@dir\this@script.mgl"%
385 \mglblock@ignore@line%
386 }
387 \def\mglblock@ignore@line#1{%
388 \mglblock@write@line%
389 }
390 \def\mglblock@write@line#1{%
391 \let\next@action\mglblock@write@line%
392 \expandafter\if#1^^M%
393 \immediate\write\mgl@out@stream{\the\mgl@line}%

```

```

394 \mgl@line{%
395 \mgl@word{%
396 \else\expandafter\if#1\space%
397 \mgl@word{%
398 \mgl@line\expandafter{\the\mgl@line#1}%
399 \else%
400 \mgl@word\expandafter{\the\mgl@word#1}%
401 \mgl@line\expandafter{\the\mgl@line#1}%
402 \test@end@mglblock{\the\mgl@word}%
403 \fi\fi%
404 \next@action%
405 }
406 \begingroup%
407 \escapechar=-1\relax%
408 \xdef\end@mglblock{\string\end\string{\mglblock\string\}}%
409 \endgroup%
410 \def\test@end@mglblock#1{%
411 \edef\this@word{#1}%
412 \ifx\this@word\end@mglblock%
413 \def\next@action{\end{mglblock}}%
414 \fi%
415 }
416 \newread\mgl@in@stream
417 \def\end@mglblock{%
418 \immediate\closeout\mgl@out@stream%
419 \immediate\openin\mgl@in@stream="\mgl@dir\this@script.mgl"%
420 \begingroup%
421 \list{\itshape\footnotesize\arabic{mgl@verb@line@no}.}{}%
422 \setlength{\labelsep}{1em}%
423 \itemsep\z@skip%
424 \leftskip\z@skip\rightskip\z@skip%
425 \verbatim@font%
426 \parindent\z@\parfillskip\@flushglue\parskip\z@skip%
427 \@vobeyspaces%
428 \mglblock@read@line%
429 }
430 \def\mglblock@read@line{%
431 \stepcounter{mgl@verb@line@no}%
432 \read\mgl@in@stream to \this@line%
433 \ifeof\mgl@in@stream%
434 \def\next@action{%
435 \immediate\closein\mgl@in@stream%
436 \endlist%
437 \endgroup%
438 }%
439 \else%
440 \def\next@action{%
441 \item\mbox{\this@line}%
442 \mglblock@read@line%
443 }%

```

```

444 \fi%
445 \next@action%
446 }
447
448 \newcounter{mgl@verb@line@no}
449 \def\mglverbatim{%
450 \setcounter{mgl@verb@line@no}{0}%
451 \list{\itshape\footnotesize\arabic{mgl@verb@line@no}.}{}%
452 \setlength{\labelsep}{1em}%
453 \itemsep\z@skip%
454 \leftskip\z@skip\rightskip\z@skip%
455 \verbatim@font%
456 \parindent\z@\parfillskip\@flushglue\parskip\z@skip%
457 \let\do\@makeother \dospecials%
458 \endlinechar'\^M \catcode'\^M\active%
459 \obeyspaces%
460 \mglverbatim@ignore@line%
461 }
462 \def\mglverbatim@ignore@line#1{%
463 \mglverbatim@write@line%
464 }
465 \newtoks\mgl@word
466 \newtoks\mgl@line
467 \def\mglverbatim@write@line#1{%
468 \let\next@action\mglverbatim@write@line%
469 \expandafter\if#1\^M%
470 \stepcounter{mgl@verb@line@no}%
471 \item\mbox{\the\mgl@line}%
472 \mgl@line{}%
473 \mgl@word{}%
474 \else\expandafter\if#1\space%
475 \mgl@word{}%
476 \mgl@line\expandafter{\the\mgl@line#1}%
477 \else%
478 \mgl@word\expandafter{\the\mgl@word#1}%
479 \mgl@line\expandafter{\the\mgl@line#1}%
480 \test@end@mglverbatim{\the\mgl@word}%
481 \fi\fi%
482 \next@action%
483 }
484 \begingroup%
485 \escapechar=-1\relax%
486 \xdef\end@mglverbatim{\string\end\string\{mglverbatim\string\}}%
487 \endgroup%
488 \def\test@end@mglverbatim#1{%
489 \edef\this@word{#1}%
490 \ifx\this@word\end@mglverbatim%
491 \def\next@action{\end{mglverbatim}}%
492 \fi%
493 }

```

```

494 \def\endmglverbatim{\endlist}
495
496 \newcommand\mglgraphics[2][{}]{%
497   \def\graph@keys{%
498     \setkeys{mgl@keys}{#1}%
499     \immediate\write18{mglconv "\mgl@dir#2.mgl" -o "\mgl@dir#2\mgl@image@ext"}
500     \mgl@include@image{\mgl@dir#2}%
501   }
502
503 \def\mglinclude#1{%
504   \setcounter{mgl@verb@line@no}{0}%
505   \immediate\openin\mgl@in@stream="\mgl@dir#1.mgl"%
506   \begingroup%
507   \list{\itshape\footnotesize\arabic{mgl@verb@line@no}.}{}%
508   \setlength{\labelsep}{1em}%
509   \itemsep\z@skip%
510   \leftskip\z@skip\rightskip\z@skip%
511   \verbatim@font%
512   \parindent\z@\parfillskip\@flushglue\parskip\z@skip%
513   \let\do\@makeother \dospecials%
514   \endlinechar'\^M \catcode'\^M\active%
515   \@vobeyspaces%
516   \mglblock@read@line%
517 }
518
519 \def\mglTeX{mgl\TeX}

```