MapleTex: Embedding Maple Results in LaTeX Documents

version 2022.02.05

Athanasios Kehagias
kehagiat@ece.auth.gr

February 7, 2022

1 Introduction

MapleTex is a *very simple* package [1] intended to enable the user to *programmatically* include the results of Maple computations in LaTeX documents; the focus is on *symbolic* computations but numerics and figures can also be used. Hence it is similar to the various "sweave" packages that exist for Matlab, SageMath, R, Python etc. But the emphasis of MapleTex is on *simplicity*. [2]

2 Installation

Simply unzip the file `MapleTex.zip` into some folder. The basic files are `MapleTex.mw` and `MapleTex.mpl`. The first is a worksheet which sets some variables and invokes the second, a script (collection of Maple commands) which performs the substitutions contained in your own (mixed Maple/LaTeX) script. Additional files are `MapleTexDoc.pdf` (the current document) and `MapleTexDoc.mpl` (the mixed Maple/LaTeX script which produced `MapleTexDoc.pdf`).

To run MapleTex you must have (obviously) installed Maple and a TeX distribution. I have tested MapleTex with Maple 2018 and TeX Live 2019 on Windows 7, 10, 11. Since it only depends on Maple and LaTeX, it will probably work fine on Linux and Apple operating systems, but I have not tested this.

3 QuickStart

Execute (in Maple) the worksheet `MapleTexDoc.mw`. You will get the files `MapleTexDoc.tex` and `MapleTexDoc.pdf` (the current file). Open `MapleTexDoc.mpl` to see the code which produced the document (some of the code is explained in a later section).

4 Usage

The MapleTex workflow is as follows.

1. Write a *single* Maple script, e.g., `foo.mpl`, which contains both Maple commands and LaTeX code (embedded as comments); the LaTeX code can include the control word `latex(a)`, where `a`, is a Maple variable (details will be given a little later).

2. The file `foo.m` must be created in the same folder which contains `MapleTex.m`.

3. Open (in Maple) the worksheet `MapleTex` and substititute `fn:="MapleTexDoc":` with `fn:="foo":`

4. Run the the worksheet; when execution is completed you have the following files.

[1] Essentially a single script.
[2] Further discussion of this point appears in the Postscript.

(a) `foo.tex`: your LaTeX code with Maple results having replaced the `latex(a)` and `figur(b)` control words.

(b) `foo.pdf`: the output of `foo.tex` as compiled by pdflatex.

To use MapleTex follow the above workflow. The file `foo.mpl` must be created in the same folder where you unzipped `Mapletex.zip`. The rules for writing MapleTex files are as follows.

1. Each line contains either *only* Maple code or *only* LaTeX code.

2. Maple code is written as usual.

3. LaTeX code is also written as usual with the following two exceptions.

   (a) Every line of LaTeX code is preceded by the characters `##` (so, as far as Maple is concerned, these are comment lines).

   (b) You can use the additional command `latex( )` (entered in LaTeX code **without** a backslash!). Every occurrence of `latex(a)` in the LaTeX part of your code will be replaced by the LaTeX expression for `a`, where it is assumed that `a` has been defined in the Maple part of your code.

4. In addition you must obey the following two rules.

   (a) The first line of `foo.mpl` must be `VARS:=[]:`.

   (b) The last line of every block of Maple code must have the form `VARS:=[op(VARS),"a","b"]` where `"a"`, `"b"` etc. are variables which you want to use for LaTeX substitutions. For example

   ```
   with(LinearAlgebra):
   C:=Matrix([[a,b],[c,d]]):
   CI:=MatrixInverse(C):
   VARS:=[op(VARS),"C","CI"]:
   ```

5 Some Examples and Explanations

Let us now look at some parts of `MapleTexDoc.mpl`.

1. The file starts with the lines

   ```
   ## \documentclass{article}
   ## \usepackage{amsmath}
   ## \usepackage{graphicx}
   ```

   and continues like this with typical LaTeX preamble commands. Note that, since these are LaTeX commands, they are preceded by `##`.

2. After a while we have the lines

   ```
   ## \begin{document}
   ## \maketitle
   ## \section{Introduction}
   ## \texttt{MapleTex.m} is a \emph{very simple} script intended to
   ## implement \emph{literate programming} in \textsf{Maple}.
   ```

   and so on, where we write our LaTeX content as usual, but always using the `##` line prefix.

3. Things get more interesting when we introduce symbolic computations. So for example the code

```
ft0:=t*exp(t):
Derft0:=Diff(ft0,t):
derft0:=diff(ft0,t):
Intft0:=Int(ft0,t):
intft0:=int(ft0,t):
VARS:=[op(VARS),"ft0","Derft0","derft0","Intft0","intft0"]:
## \item Let us compute the derivative and the indefinite integral
## of \(f(t)=latex (ft0)\). They are
## \[
## latex (Derft0)=latex (derft0) \qquad \text{and}
## \qquad latex (Intft0)=latex (intft0).
## \]
```

produces the following results:

Let us compute the derivative and the indefinite integral of $f(t) = t\mathrm{e}^t$. They are

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(t\mathrm{e}^t\right) = \mathrm{e}^t + t\mathrm{e}^t \qquad \text{and} \qquad \int t\mathrm{e}^t\,\mathrm{d}t = (t-1)\,\mathrm{e}^t.$$

Similarly, the code

```
g:=1/n^2:
G1:=Sum(g,n=1..infinity):
G2:=sum(g,n=1..infinity):
G0:=evalf(G2):
VARS:=[op(VARS),"g","G1","G2","G0"]:
## \qquad Let us write the sum of \(g(n)=latex (g)\), i.e.,
## \(latex (G1)  = latex (G2) = latex (G0)\).
```

produces the following results

Let us write the sum of $g(n) = n^{-2}$, i.e., $\sum_{n=1}^{\infty} n^{-2} = 1/6\,\pi^2 = 1.644934068$.

4. Here is an example on how to include figures. Suppose you want to plot a sinusoid; you can use the following code in `foo.mpl`.

```
p1:=plot(sin(3*x),x=0..2*Pi):
plotsetup(png, plotoutput = "FIG001"):
print(p1):
## \begin{figure}[H]
## \centering
## \includegraphics[scale=0.2]{FIG001}
## \caption{A plot of \(F(x)=\sin(3x)\).}  \label{FIG001}
## \end{figure}
```

The first line of the above fragment creates a plot object, the second one sets the plot output to be the file `FIG001.png` [3] and the third prints the plot `p1` to `FIG001.pdf`. The remaining five lines are "regular" LaTeX code which includes the figure in the LaTeX document. This results in the following plot

[3] Various `plotoptions` can be included; see the Maple documentation on `plotsetup`.

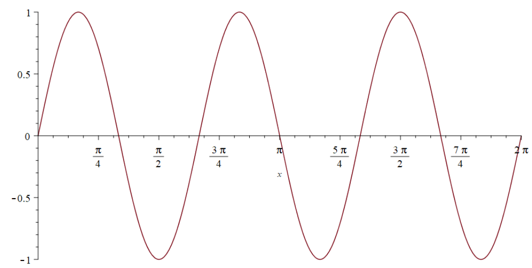

Figure 1: A plot of $F(x) = \sin(3x)$.

5. Let us now present some additional symbolic results. To see the code which generates the following lines open MapleTexDoc.mpl and look at around lines 191-275.

   (a) Let us solve the equation
   $$x^2 + x + 1 = 0.$$
   The solution is
   $$x_1 = -1/2 + i/2\sqrt{3} \qquad \text{and} \qquad x_2 = -1/2 - i/2\sqrt{3}.$$

   (b) Now let us solve the differential equation
   $$\frac{\mathrm{d}^2}{\mathrm{d}t^2} f(t) - 3\frac{\mathrm{d}}{\mathrm{d}t} f(t) + 2 f(t) = \mathrm{e}^{3t} \text{ with } f(0) = \mathrm{e} \text{ and } \mathrm{D}(f)(0) = \mathrm{e}$$
   The solution is
   $$f(t) = \left(1/2\left(\mathrm{e}^t\right)^2 - \mathrm{e}^t + \mathrm{e} + 1/2\right)\mathrm{e}^t$$

   (c) Next let us solve the integral equation
   $$\int_0^t x(s)\,\mathrm{e}^{t-s}\,\mathrm{d}s = \sin(t).$$
   The solution is
   $$x(t) = \cos(t) - \sin(t)$$

   (d) The Taylor series of $f(x) = \frac{\mathrm{e}^x}{x+1}$ is
   $$1 + 1/2\,x^2 - 1/3\,x^3 + 3/8\,x^4 - \frac{11\,x^5}{30}$$

   (e) We can do integral transforms. The Fourier transform of $f(t) = \left(t^2+1\right)^{-1}$ is
   $$F(w) = \pi\left(\mathrm{e}^{-w}\mathit{Heaviside}(w) + \mathrm{e}^{w}\mathit{Heaviside}(-w)\right)$$
   and the inverse Laplace transform of $F(s) = \frac{s+1}{s^2+s+1}$ is
   $$f(t) = 1/3\,\mathrm{e}^{-t/2}\left(3\cos\left(1/2\sqrt{3}t\right) + \sqrt{3}\sin\left(1/2\sqrt{3}t\right)\right)$$

   (f) We define a matrix and compute its powers
   $$C = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \qquad C^2 = \begin{bmatrix} a^2+bc & ab+bd \\ ca+dc & bc+d^2 \end{bmatrix}, \qquad C^{-1} = \begin{bmatrix} \frac{d}{ad-bc} & -\frac{b}{ad-bc} \\ -\frac{c}{ad-bc} & \frac{a}{ad-bc} \end{bmatrix}.$$

(g) We can also insert 3D figures such as the following.

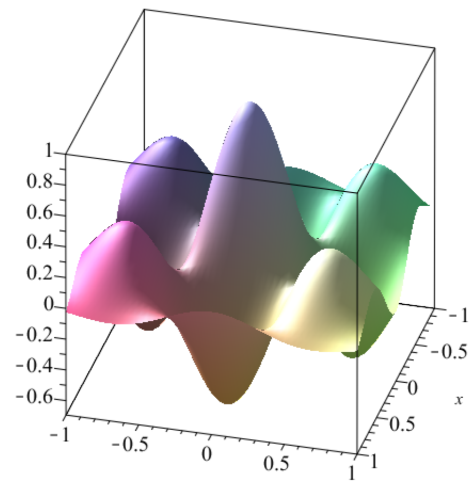

Figure 2: A plot of $G(x, y) = e^{-(x^2+y^2)} \cos(ax) \cos(by)$.

6 Postscript

My motivation for writing MapleTex was the need for a simple system to create problem sets and exams (and answers!) for my math classes. I have been using several computer algebra systems: Maple, SageMath, SymPy, the symbolic math toolboxes of Matlab and Octave, and so on. Not wanting to reinvent the wheel I looked at what was available. I started with SageTex; alas, I was never able to properly install and run it. I tried several additional packages and I found each one either too confusing to set up, or not providing the functionalities I wanted, or both.

Consequently I decided to write my own package. I am a simple guy and MapleLaTex is a simple hack. I am certain that any sufficiently interested serious programmer can produce a much better version and I will be very happy if someone does. In the meantime, MapleTex works right out of the box and does what I want it to do: programmatically incorporate Maple results into my LaTeX documents.

In conclusion, there are a few improvements / extensions on which I hope to work in the future. I list them in order of decreasing priority.

1. Automatic generation of a list of substitutions. Currently this has to be provided by the user (this is the role of the `VARS:=[op(VARS),"a","b"]` commands). I was looking for a Maple command which generates the list of all workspace variables (something like `whois` in Matlab) but I could not find it; any help will be appreciated.

2. In the current implementation, variable names cannot be "reused". If a variable `a` appears several times in the LaTeX commands, it will always be replaced by its last computed value. I would like to be able to replace each occurrence of `a` with its value as computed just before *this* occurrence.

Finally, let me mention that I am also working on OctLatex and MatLatex which are Octave and Matlab versions of the MapleTex idea.