# MATweave: Integration of MATLAB/Octave Code Inside LaTeX

Neil D. Lawrence
neil@dcs.sheffield.ac.uk
Sheffield Institute for Translational Neuroscience
and Department of Computer Science
University of Sheffield, U.K.

**Abstract**

MATweave is a set of simple tricks involving code snippets for incorporating MATLAB/Octave code in your LaTeX documents and producing the result as figures. The idea is to make it as easy as possible for MATLAB/Octave users to make their results really reproducible.

## 1 Introduction

MATweave is a really simple idea for incorporating MATLAB or Octave code into your LaTeX source code. This will allow This is an example of incorporating MATLAB or Octave code with LaTeX for really reproducible research (**??**). Really reproducible research is about ensuring that your results are usable by the rest of the community. MATweave is inspired by Sweave (**?**), an approach for combining R programs with LaTeX and dynamically updating reports.

### 1.1 The Tricks

The idea behind the system is that you can use two tricks to integrate code from MATLAB with LaTeX.

1. The first trick is to make use of block comments, introduced in Octave 3.2 and MATLAB R14. Block comments in MATLAB/Octave are opened with `%{` and closed with `%}` allowing you to write:

```
%{
This is a comment!
This line is also commented!
%}
for i = 1:10
  % Code here is outside the block comment.
  disp(i)
end
```

   The trick for using these comments is that a block comment in MATLAB/Octave is a comment in LaTeX (because it starts with a `%`), but *not* a block comment. This means

```
%{
In MATLAB/Octave this would be a comment, but in \LaTeX\ it would be
compiled! So we can write valid \LaTeX, $\tau = 2\pi$, inside a MATLAB
file. Now we need to be able to write MATLAB inside a \LaTeX\ file!
%}
```

Figure 1: Plot of the standard Gaussian density.

2. The second trick is to include the `verbatim` package and use it to define a new MATLAB/Octave environment using the following commands.

```
\newenvironment{matlab}{\comment}{\endcomment}
\newenvironment{octave}{\comment}{\endcomment}
\newenvironment{matlabv}{\verbatim}{\endverbatim}
\newenvironment{octavev}{\verbatim}{\endverbatim}
```

   This allows you to include MATLAB/Octave code that won't be read by LaTeX using e.g. `\begin{octave} ... \end{octave}` and code that will be shown in a verbatim environment using `\begin{octavev} ... \end{octavev}`. Of course you could do this using the standard `verbatim` environment, but by defining a new environment you can keep track of which bits of code are being run.

### 1.2 Example

The following code is included using the `matlabv` environment defined above. The source code for this document starts with a MATLAB/Octave open block quote, `%{`. In effect, that quote is closed at the beginning of the MATLAB/Octave code listed below so that the code will be compiled when it is read in MATLAB/Octave.

```
%}
% Matlab code starts here
tau = 2*pi;
x = linspace(-3, 3, 100)';
y = 1/sqrt(tau)*exp(-0.5*x.^2);
plot(x, y, 'r-');
set(gca, 'fontname', fontName, 'fontsize', fontSize);
print -depsc myGaussian.eps
% If you use pdflatex you will need a
% system call here to convert the
% eps to a pdf. For example
system('epstopdf myGaussian.eps');
% Matlab code ends here
%{
```

This combination of two tricks means that the source code for this document can be compiled in LaTeX or in MATLAB/Octave.

## 2 Running MATweave

You can run the LaTeX file in Octave with the command `source`. For example, save this as `myexample.tex` and run `source myexample.tex`. MATLAB is slightly more complicated.

The result of the code is included in Figure **??**.

Of course in most cases you would probably want to hide the MATLAB code that was used to generate the figure (although in early versions of the paper you might want it there for debugging). The histogram in Figure **??** has the code hidden using the `\begin{matlab} ... \end{matlab}` environment.

These tricks should allow anyone to recreate your figures if they have the LaTeX source file.

You could also use the `listings` package to display your MATLAB/Octave code to better effect.

You can even use the system to generate tables, or other text to be included directly in your LaTeX document,

Figure 2: Histogram of 1000 samples from the standard Gaussian density.

Table 1: A table of random numbers generated in MATLAB/Octave.

| $A$ | $B$ | $C$ | $D$ |
|---|---|---|---|

```
%}
% Code for generating a table of random numbers.
rows = 3;
cols = 4;
numSigFigs = 3;
resultMatrix = randn(3, 4); % Would normally come from an algorithm.
fid = fopen('results.tex', 'w');
for i = 1:rows
  for j = 1:cols
    fprintf(fid, ['$' num2str(resultMatrix(i, j), numSigFigs) '$']);
    if j < cols
      fprintf(fid, ' & ');
    end
  end
  if i < rows
    fprintf(fid, '\\\\\n');
  end
end
fclose(fid);
%{
```

The results file can then be incorporated using `\input{results.tex}` and placed as in Table **??**.

## 2.1 Tip

I like to start my LaTeX files with a series of set up commands from MATLAB, such as loading in the right path, clearing the memory space, setting the random seed. That means that you can get consistent results from running the document. Have a look at the source code to see it being done for this document.

## 2.2 Note for Beamer Users

If you use the MATweave techniques in your slides using Beamer, you will have to use the option `[fragile]` for any frame that contains the MATLAB environment, as without this option Beamer cannot handle `verbatim` or `comment` environments.

You will need to use MATLAB R14 or higher or Octave 3.2 or higher. Then, to compile this file under Unix write

```
octave --eval source\ myexample.tex
```

or

```
matlab < myexample.tex
```

then

```
pdflatex myexample
```