MathSweave v0.1

Ross Dempsey

May 3, 2020

MathSweave (or MathsWeave, in British English) allows Mathematica code and its output to be embedded directly into LaTeX source.

1 Installation

Put `mathsweave.sty` and its dependency `mmacells.sty` where LaTeX can see them (an easy short-term solution is to put them in the same directory as your main `.tex` file). To compile your document, use

`pdflatex`→`mathematica`→`pdflatex`→`pdflatex`.

This is the same pattern as for using other external programs, e.g., `bibtex`.

More specifically, when you run `pdflatex job.tex` the first time, LaTeX will produce `job.m` (along with other dependencies). You should run `math -script job.m` before re-running `pdflatex` twice.

You will need the `CellsToTeX` package installed in Mathematica. Instructions for this are here.

2 Usage

Currently MathSweave provides a single environment, `mma`, in which you can write Mathematica code. The output will be this code along with the expression it evaluates to, formatted as Mathematica cells.

For example, the source

```
    \begin{mma}
        Sum[1/n^2, {n,1,Infinity}]
    \end{mma}
```

produces the following output:

```
In[1]:=  Sum[n^(-2), {n,1,Infinity}]
```

Out[1]= $\frac{\pi^2}{6}$

You can run any Mathematica code this way. For example,

In[2]:=

```
Table[Integrate[x^n,x], {n, 0, 5}]
```

Out[2]=

$$\{x, \frac{x^2}{2}, \frac{x^3}{3}, \frac{x^4}{4}, \frac{x^5}{5}, \frac{x^6}{6}\}$$

If you have more complicated code, you can enter multi-line input and it will appear faithfully, tabs and all, in the input cell. The exception is when a line break is forced by the width of the page.

In[3]:=

```
EulerIntegrate[f_, a_, b_, n_] := With[{dx = (b-a)/n},
    Total[
        Table[
            N[f[x] dx], {x, Drop[Subdivide[a, b, n],-1]}
        ]
    ]
];
NumberForm[#, 5]&/@Table[EulerIntegrate[Log, 1, E, n], {n, {10, 100, 1000,
        10000}}]//TableForm
```

Out[3]=

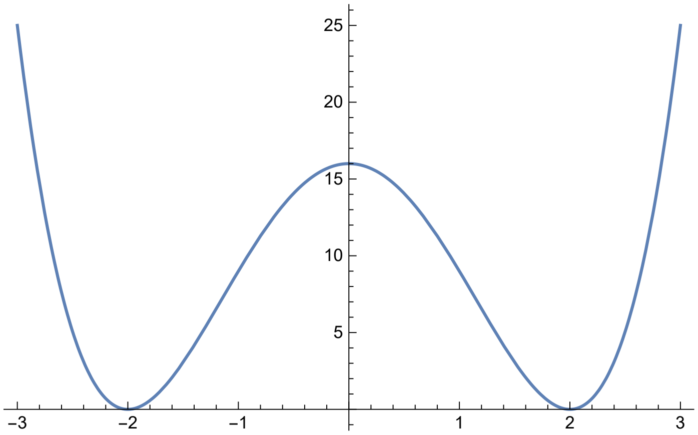
```
0.91253
0.99139
0.99914
0.99991
```

In each of the examples above, the output is formatted in LaTeX and can be highlighted and copied as text. When the output is graphical, or otherwise fails to be formatted as text, MathSweave will attempt to export the output to a pdf and embed this. For example, MathSweave can include plots in this way.

In[4]:=

```
f[x_] := (x+2)^2(x-2)^2;
Plot[f[x],{x,-3,3}]
```

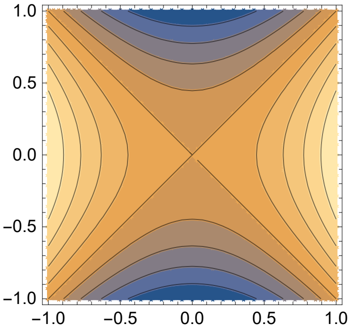
Out[4]=



In[5]:=

```
ContourPlot[x^2-y^2,{x,-1,1},{y,-1,1},ImageSize->Small]
```

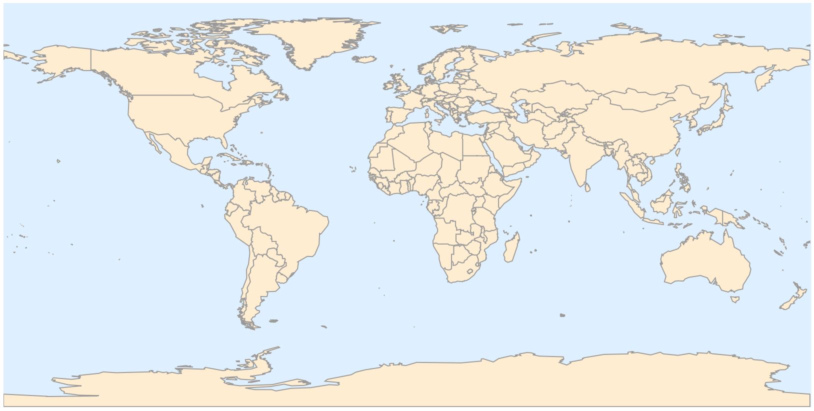
Out[5]=



All of your input is written to files and then imported into the main Mathematica script. The main script will manipulate your code as a string, but any strings in your code are properly escaped so you don't have to worry about this.

```
In[6]:=    GeoGraphics["World", GeoBackground -> "CountryBorders"]
```

Out[6]=



3 Known Bugs and Limitations

- MathSweave only captures the result of *evaluating* the input code. Messages or `Print` statements will be lost. For example,

  ```
  In[7]:=    Print["Hello world!"]

  Out[7]=    Null
  ```

- The Mathematica script has to connect to a FrontEnd kernel in order to parse the code. Occasionally this fails, in which case you should clear auxiliary files and try again.
- If any of your Mathematica code has a syntax error, behavior is undefined. Hopefully it will fail gracefully and produce a `$Failed` message for only that cell, but no promises.
- Some functions which were introduced in recent versions of Mathematica are not properly syntax-highlighted.
- All Mathematica code has to appear in the `mma` environment. In the future there will be a way of silently executing Mathematica code, so that not all code and output has to be printed to the document, or to only display the output, so that (for example) Mathematica can be used to generate a figure without having to show the code.