Passing LaTeX data to Python

There are occassions where the active Python code may require information from the LaTeX source. Since Python has no direct access to LaTeX some other means must be provided to build this bridge. The idea presented here follows a standard LaTeX pattern – run LaTeX twice, once before Python, leaving breadcrumbs for Python to pickup during its run, then a final LaTeX run to complete the job.

The example shown here is a simple proof of concept. It shows how the dimensions of plot can be specified in the LaTeX source and then accessed later by Python.

To compile this example, use

```
pdflatex      example-09
pylatex.sh -i example-09
```

The idea is to use LaTeX to create a Python dictionary saved as a `.json` file. The dictionary will contain just two entries, one for the height and one for the width of the plot. Here are the relevant lines of LaTeX.

```
\newdimen\mywidth\mywidth=17cm       % target width
\newdimen\myheight\myheight=13.5cm   % target height

\newwrite\breadcrumbs
\immediate\openout\breadcrumbs=\jobname.json                % create Json file
\immediate\write\breadcrumbs{\writebgroup}                  % {
\immediate\write\breadcrumbs{"height":\inches{\myheight},}  %    "height":5.31496,
\immediate\write\breadcrumbs{"width":\inches{\mywidth}}     %    "width":6.69292
\immediate\write\breadcrumbs{\writeegroup}                  % }
\immediate\closeout\breadcrumbs                             % close the file
```

The target dimensions are: (width, height) = (17cm, 13.5cm).

The following Python code does the job of reading the dictionary and setting the values of `height` and `width`.

```python
import io, os, json

# read the dictionary
try:
   with io.open(os.getcwd() + '/' + 'example-09.json') as inp_file:
      inp = json.load(inp_file)
except:
    inp = { "width":"6.4",
            "height":"4.8" }

# set figure dimensions in inches (yikes)
width  = inp['width']
height = inp['height']
```

This Python code does the job of plotting the Bessel functions (it is based on the code from example-04).

```python
# plot the Bessel functions
import numpy as np
import scipy.special as sp
import matplotlib.pyplot as plt

plt.matplotlib.rc('text', usetex = True)
plt.matplotlib.rc('grid', linestyle = 'dotted')
plt.matplotlib.rc('figure', figsize = (width,height))

x = np.linspace(0, 15, 500)

for v in range(0, 6):
    plt.plot(x, sp.jv(v, x))

plt.xlim((0, 15))
plt.ylim((-0.5, 1.1))
plt.legend(('${J}_0(x)$', '${J}_1(x)$', '${J}_2(x)$',
            '${J}_3(x)$', '${J}_4(x)$', '${J}_5(x)$'), loc = 0)
plt.xlabel('$x$')
plt.ylabel('${J}_n(x)$')
plt.grid(True)

plt.tight_layout(0.5)

plt.savefig('example-09-fig.pdf')

width_cm  = round (width  * 2.54, 2)   # py (width,width_cm)
height_cm = round (height * 2.54, 2)   # py (height,height_cm)
```

The following figure should have dimensions: (width, height) = (17.0 cm, 13.5 cm).

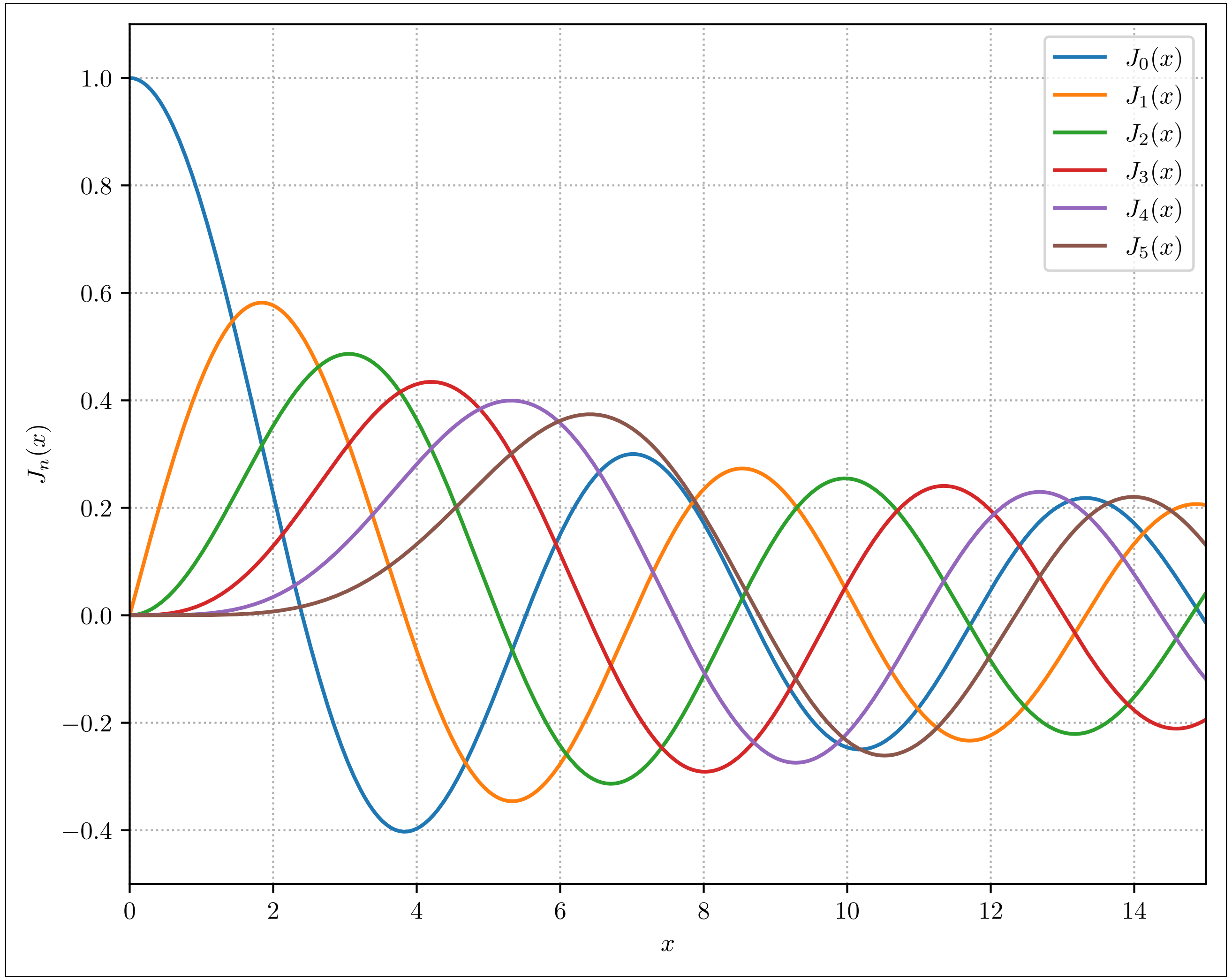

Figure 1: The first six Bessel functions.