

Jupyter notebook custom conversion

Romain Madar

August 2018

Overview

- 1 nbconvert latex test
- 2 Printing using python
- 3 Pyout (and Text Wrapping)
- 4 Image and plots
- 5 Operator Highlighting Check
- 6 Tables
- 7 Sympy output
- 8 Line Length

```
%load_ext autoreload
```

nbconvert latex test

Printing using python

Pyout (and Text Wrapping)

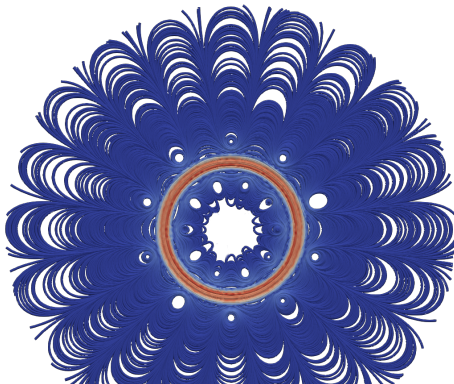
Image and plots

As plain text using markdown

Once exported as markdown and converted to latex/pdf with pandoc, the `{width=60%}` will fix the width of the picture and the `My legend` will appear as caption:

```
![My legend](figures/magnetostatics_field.png){width=50% #figlabel}
```

gives the result shown in [this figure](#).

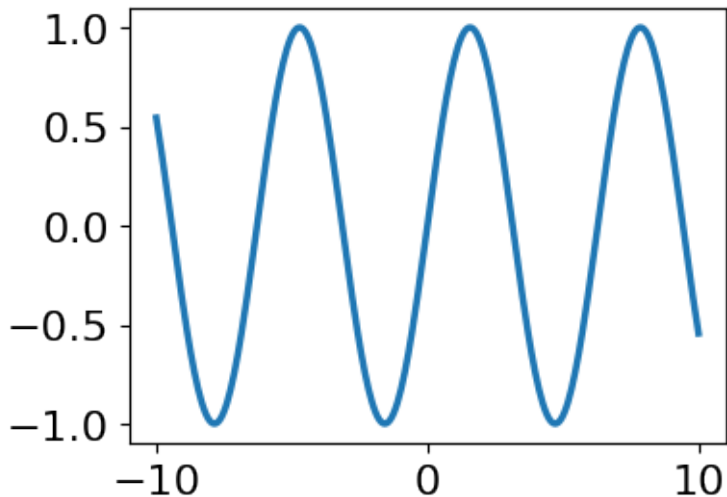


Plots produced by the code I

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(-10,10,300)
y = np.sin(x)
plt.figure(figsize=(4,3),dpi=100)
p=plt.plot(x,y)
```

Plots produced by the code II



Plots produced by the code III

Figure 3: png

```
import os  
os.getcwd()
```

```
'/home/rmadar/Documents/work/temp/pandoc-utils/examples/NotebookWithCode'
```

```
plt.figure(figsize=(8,3))  
plt.plot(x,y)  
plt.xlabel('$X_{e}$ [SI]')  
plt.ylabel('$Y_{UPDATE12}$ [SI]')  
plt.tight_layout()  
jpu.plt2md('myplot', 'This is a test of how to get proper'+\  
    'plot from Jupyter notebook in MD, '+\  
    'to be processed using PANDOC', '100%')
```

Plots produced by the code IV

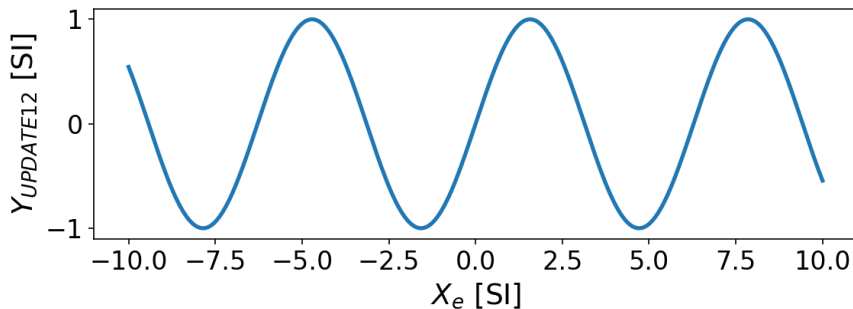


Figure 4: This is a test of how to get properplot from Jupyter notebook in MD,to be processed using PANDOC

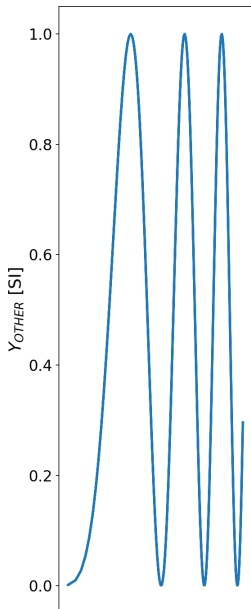
Plots produced by the code V

We can then refer to a given figure using cross-references [like this](#), obtained with:

```
[like this](#myplot)
```

```
plt.figure(figsize=(4,10))
plt.plot(np.sqrt(np.abs(x)),y**2)
plt.xlabel('$X_{e}$ [SI]')
plt.ylabel('$Y_{OTHER}$ [SI]')
plt.tight_layout()
jpu.plt2md('myplot2','This is another test of how to get proper'+\
'plot from Jupyter notebook in MD,'+\
'to be processed using PANDOC','50%')
```

Plots produced by the code VI



Operator Highlighting Check

Tables

Markdown as plain text

First a *markdown* table:

Column 1	Column 2
1	3
a	b
4	&

Pandas as default and Markdown I

```
import pandas as pd
df=pd.DataFrame(np.random.randn(10,3))
```

```
# Default printing is HTML, so it looks good
# on the web but it is not well
# rendered in pdf via ipynb->MB->pdf (using nbconvert
# and pandoc)
df
```

0

1

2

0

0.156292

1.139407

0.745845

Pandas as default and Markdown II

1

-0.460238

-0.439775

1.840901

2

1.553525

0.584756

1.364348

3

-1.104916

-0.421400

0.507061

Pandas as default and Markdown III

4

-0.523635

2.280238

1.192887

5

0.695859

-0.346761

-0.517605

6

0.442680

1.694490

-0.639051

Pandas as default and Markdown IV

7

-0.952961

1.703176

0.413999

8

1.026344

-0.111948

-0.171396

9

0.804994

1.118887

1.978886

Pandas as default and Markdown V

```
# Markdown printing well rendered
# in MD using nbconvert
jpu.df2md(df)
```

	0	1	2
0	0.156292	1.13941	0.745845
1	-0.460238	-0.439775	1.8409
2	1.55352	0.584756	1.36435
3	-1.10492	-0.4214	0.507061
4	-0.523635	2.28024	1.19289
5	0.695859	-0.346761	-0.517605
6	0.44268	1.69449	-0.639051
7	-0.952961	1.70318	0.413999
8	1.02634	-0.111948	-0.171396
9	0.804994	1.11889	1.97889

Sympy output

Line Length