

# Jupyter notebook custom conversion

Romain Madar

August 2018

## Contents

<b>1</b>	<b>nbconvert latex test</b>	<b>2</b>
<b>2</b>	<b>Printing using python</b>	<b>2</b>
<b>3</b>	<b>Pyout (and Text Wrapping)</b>	<b>3</b>
<b>4</b>	<b>Image and plots</b>	<b>5</b>
4.1	As plain text using markdown . . . . .	5
4.2	Plots produced by the code . . . . .	6
<b>5</b>	<b>Operator Highlighting Check</b>	<b>9</b>
<b>6</b>	<b>Tables</b>	<b>10</b>
6.1	Markdown as plain text . . . . .	10
6.2	Pandas as default and Markdown . . . . .	10
<b>7</b>	<b>Sympy output</b>	<b>12</b>
<b>8</b>	<b>Line Length</b>	<b>13</b>

## 1 nbconvert latex test

**Lorem ipsum** dolor sit amet, consectetur adipiscing elit. Nunc luctus bibendum felis dictum sodales. Ut suscipit, orci ut interdum imperdiet, purus ligula mollis *justo*, non malesuada nisl augue eget lorem. Donec bibendum, erat sit amet porttitor aliquam, urna lorem ornare libero, in vehicula diam diam ut ante. Nam non urna rhoncus, accumsan elit sit amet, mollis tellus. Vestibulum nec tellus metus. Vestibulum tempor, ligula et vehicula rhoncus, sapien turpis faucibus lorem, id dapibus turpis mauris ac orci. Sed volutpat vestibulum venenatis.

This is a test list:

1. item 1
  - subitem 1
  - subitem 2
2. item 2
3. item 3

## 2 Printing using python

```
next_paragraph = """
Aenean vitae diam consectetur, tempus arcu quis, ultricies urna. Vivamus
↪ venenatis sem
quis orci condimentum, sed feugiat dui porta.
"""

def identity_dec(ob):
    return ob

@identity_dec
def nifty_print(text):
    """Used to test syntax highlighting"""

    print(text * 2)

nifty_print(next_paragraph)
```

Aenean vitae diam consectetur, tempus arcu quis, ultricies urna. Vivamus  
venenatis sem  
quis orci condimentum, sed feugiat dui porta.

Aenean vitae diam consectetur, tempus arcu quis, ultricies urna. Vivamus  
venenatis sem  
quis orci condimentum, sed feugiat dui porta.

### 3 Pyout (and Text Wrapping)

```
Text = """
Aliquam blandit aliquet enim, eget scelerisque eros adipiscing quis. Nunc sed
↳ metus
ut lorem condimentum condimentum nec id enim. Sed malesuada cursus hendrerit.
↳ Praesent
et commodo justo. Interdum et malesuada fames ac ante ipsum primis in
↳ faucibus.
Curabitur et magna ante. Proin luctus tellus sit amet egestas laoreet. Sed
↳ dapibus
neque ac nulla mollis cursus. Fusce mollis egestas libero mattis facilisis.
"""
Text #Use print(Text) instead to get text wrapping in pdf
```

```
'\nAliquam blandit aliquet enim, eget scelerisque eros adipiscing quis. Nunc
sed metus \nut lorem condimentum condimentum nec id enim. Sed malesuada
cursus hendrerit. Praesent \net commodo justo. Interdum et malesuada fames ac
ante ipsum primis in faucibus. \nCurabitur et magna ante. Proin luctus tellus
sit amet egestas laoreet. Sed dapibus \nneque ac nulla mollis cursus. Fusce
mollis egestas libero mattis facilisis.\n'
```

```
print(Text)
```

Aliquam blandit aliquet enim, eget scelerisque eros adipiscing quis. Nunc sed  
metus  
ut lorem condimentum condimentum nec id enim. Sed malesuada cursus hendrerit.  
Praesent  
et commodo justo. Interdum et malesuada fames ac ante ipsum primis in  
faucibus.  
Curabitur et magna ante. Proin luctus tellus sit amet egestas laoreet. Sed  
dapibus  
neque ac nulla mollis cursus. Fusce mollis egestas libero mattis facilisis.

```
import numpy as np
```

```
a = np.random.rand(10,10)
```

```
print(a)
```

```
a
```

```
[[0.64744667 0.57636518 0.92308083 0.11504091 0.14889798 0.97040679
 0.70388468 0.54002177 0.37008402 0.22467851]
 [0.51697607 0.31691022 0.84519629 0.14154867 0.35497008 0.15084046
 0.50688335 0.67993042 0.18273082 0.07984997]
 [0.85085089 0.11873244 0.30033449 0.54641412 0.82311172 0.86073597
 0.87178896 0.83388601 0.35274997 0.66044507]
 [0.96852645 0.74269565 0.26353289 0.54570074 0.69385546 0.66444893
 0.66926216 0.38622088 0.70712221 0.08701712]
 [0.49372412 0.05628148 0.47502442 0.66210835 0.65991268 0.37482306
 0.62422273 0.35865507 0.92692913 0.41553871]
 [0.68762822 0.27144171 0.34328767 0.55213573 0.6000758 0.67550676
 0.86156187 0.21786351 0.32769464 0.98356657]
 [0.68476955 0.94943209 0.22920439 0.12072374 0.04124654 0.6085241
 0.43001339 0.01491417 0.91422717 0.05937777]
 [0.75582926 0.21751315 0.81533493 0.58812065 0.21334823 0.45587509
 0.52656256 0.73150306 0.24616793 0.51882674]
 [0.09167766 0.29436438 0.94930157 0.45402702 0.54320104 0.5981774
 0.56577774 0.36765325 0.72958224 0.86144542]
 [0.92032691 0.47876663 0.6084082 0.55141377 0.59939395 0.57299142
 0.37665782 0.27552383 0.09787032 0.83750694]]
```

```
array([[0.64744667, 0.57636518, 0.92308083, 0.11504091, 0.14889798,
        0.97040679, 0.70388468, 0.54002177, 0.37008402, 0.22467851],
       [0.51697607, 0.31691022, 0.84519629, 0.14154867, 0.35497008,
        0.15084046, 0.50688335, 0.67993042, 0.18273082, 0.07984997],
       [0.85085089, 0.11873244, 0.30033449, 0.54641412, 0.82311172,
        0.86073597, 0.87178896, 0.83388601, 0.35274997, 0.66044507],
       [0.96852645, 0.74269565, 0.26353289, 0.54570074, 0.69385546,
        0.66444893, 0.66926216, 0.38622088, 0.70712221, 0.08701712],
       [0.49372412, 0.05628148, 0.47502442, 0.66210835, 0.65991268,
        0.37482306, 0.62422273, 0.35865507, 0.92692913, 0.41553871],
       [0.68762822, 0.27144171, 0.34328767, 0.55213573, 0.6000758 ,
        0.67550676, 0.86156187, 0.21786351, 0.32769464, 0.98356657],
       [0.68476955, 0.94943209, 0.22920439, 0.12072374, 0.04124654,
```

```

0.6085241 , 0.43001339, 0.01491417, 0.91422717, 0.05937777],
[0.75582926, 0.21751315, 0.81533493, 0.58812065, 0.21334823,
0.45587509, 0.52656256, 0.73150306, 0.24616793, 0.51882674],
[0.09167766, 0.29436438, 0.94930157, 0.45402702, 0.54320104,
0.5981774 , 0.56577774, 0.36765325, 0.72958224, 0.86144542],
[0.92032691, 0.47876663, 0.6084082 , 0.55141377, 0.59939395,
0.57299142, 0.37665782, 0.27552383, 0.09787032, 0.83750694]])

```

## 4 Image and plots

### 4.1 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](#).

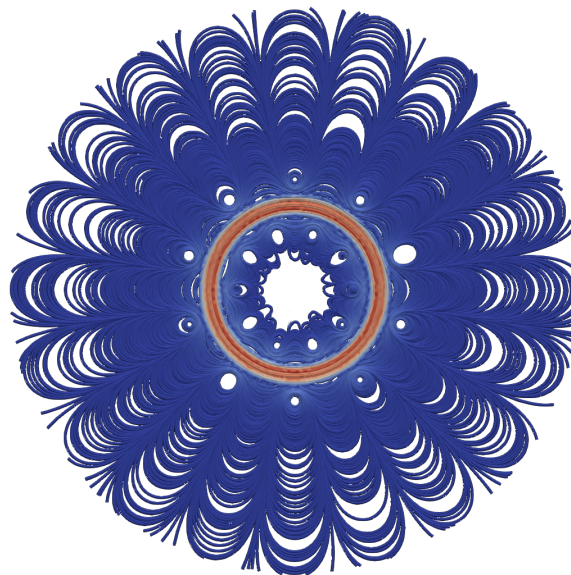


Figure 1: My legend

```

from IPython.core.display import Image
Image(data="http://ipython.org/_static/IPy_header.png")

```

# IP[y]: IPython Interactive Computing

Figure 2: png

## 4.2 Plots produced by the code

```
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)
```

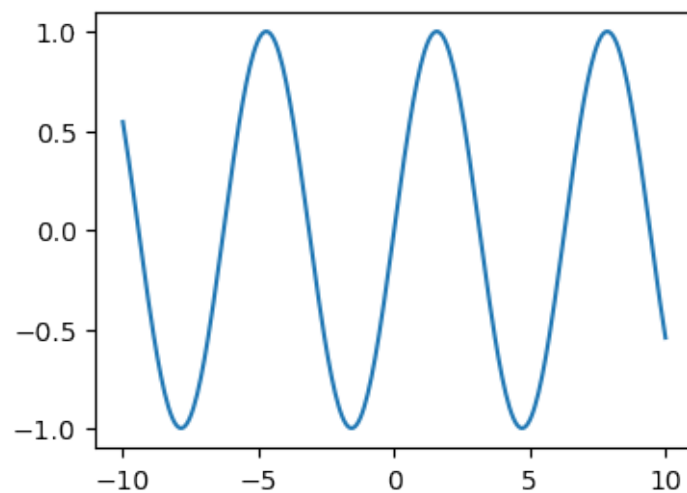


Figure 3: png

```
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%')
```

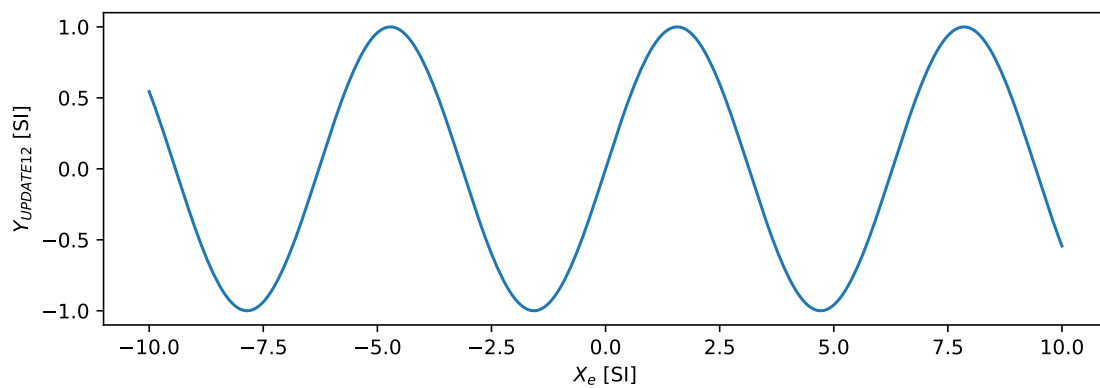


Figure 4: This is a test of how to get properplot from Jupyter notebook in MD,to be processed using PANDOC

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

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

```
plt.figure(figsize=(3,3))
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%')
```

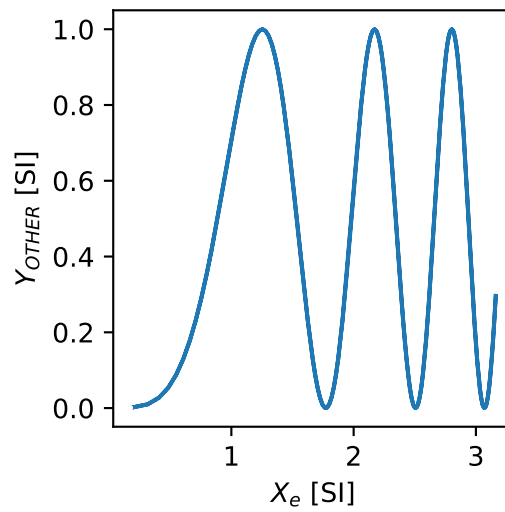


Figure 5: This is another test of how to get properplot from Jupyter notebook in MD,to be processed using PANDOC

```
plt.figure(figsize=(6,4))
plt.plot(x**3,y)
plt.plot(x**3,y+1)
plt.xlabel('test')
plt.ylabel('test2')
plt.tight_layout()
jpu.plt2md('myplot3','Adding more plot without re-creating a figure: curves
↳ cumulates','50%')
```

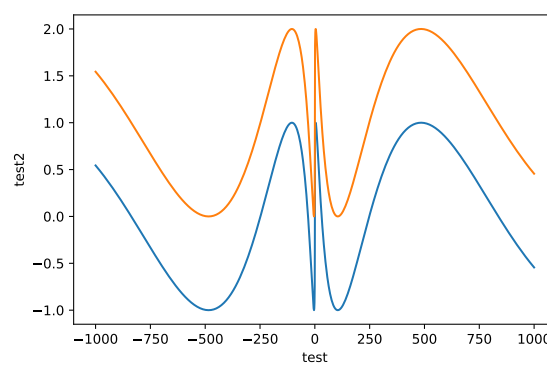


Figure 6: Adding more plot without re-creating a figure: curves cumulates



```
plt.figure(figsize=(6,3))
plt.plot(x**2,y)
plt.xlabel('$X_{\omega}$ [Hz]$')
plt.ylabel('$Y_{\varphi}$ [$\hbar$]')
plt.title('My beautiul plot')
plt.tight_layout()
jpu.plt2md('myplot4','More plot with re-creating a figure: only last
↳ curve','100%')
```

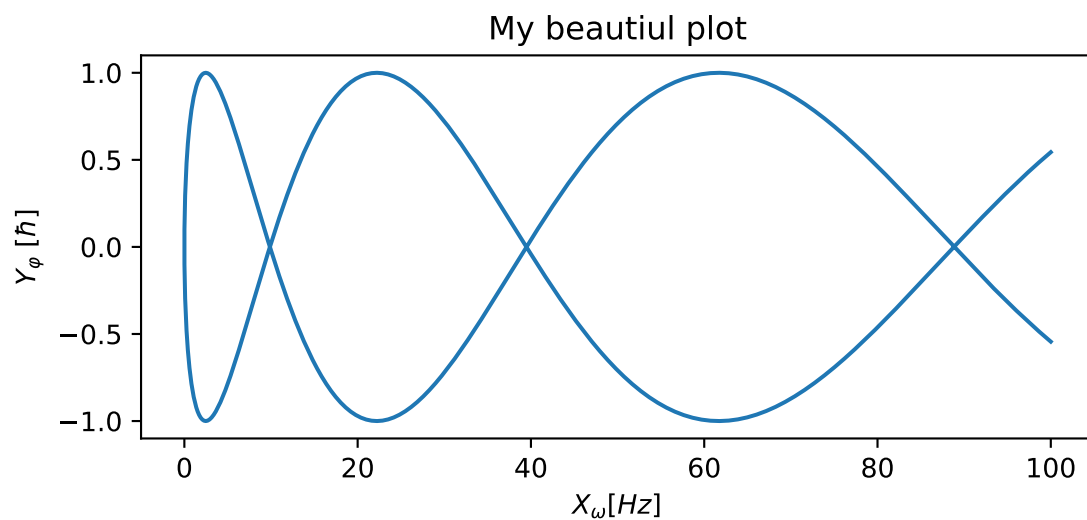


Figure 7: More plot with re-creating a figure: only last curve

## 5 Operator Highlighting Check

```
#This is a comment with an operation x @ y in it.
test = 5**9 + 2 - x @ y / (7 % 2) + True * 7
print(test)

a = set([1,2,3,4,5,6,7,8,9,0])
b = set([2,4,6,8,0])
a & b
```

1952904.9236703357

{0, 2, 4, 6, 8}

## 6 Tables

### 6.1 Markdown as plain text

First a *markdown* table:

Column 1	Column 2
1	3
a	b
4	&

### 6.2 Pandas as default and Markdown

```
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). A special function `df2md(df)` is included in the `jupy_pandoc_utils` package to write out a markdown format.

```
# Good in HTML, but not pure markdown
df
```

0

1

2

0

-0.382791

-0.004340

1.695997

1

-0.758875

-0.370357

2.157548

2

-0.554216

0.279255

0.815516

3

1.497926

2.012358

-0.087838

4

-0.682434

0.377868

-0.870262

5

-0.084107

1.232087

0.647429

6

-1.116830

1.828608

-2.012901

7

-0.075781

1.327189

-0.951601

8

-1.241108

-0.762719

0.984073

9

-0.433925

0.319959

-0.931512

```
# Good in pure MD
jpu.df2md(df)
```

	0	1	2
0	-0.382791	-0.00434025	1.696
1	-0.758875	-0.370357	2.15755
2	-0.554216	0.279255	0.815516
3	1.49793	2.01236	-0.0878384
4	-0.682434	0.377868	-0.870262
5	-0.0841065	1.23209	0.647429
6	-1.11683	1.82861	-2.0129
7	-0.0757814	1.32719	-0.951601
8	-1.24111	-0.762719	0.984073
9	-0.433925	0.319959	-0.931512

## 7 Sympy output

```
import sympy
from sympy.abc import x, n, m
sympy.init_printing()
theta = sympy.Symbol('theta')
phi = sympy.Symbol('phi')

sympy.simplify(sympy.Ynm(n,m,theta,phi).expand(func=True))
```

$$\frac{\sqrt{\frac{(2n+1)\Gamma(-m+n+1)}{\Gamma(m+n+1)}} e^{im\phi} P_n^{(m)}(\cos(\theta))}{2\sqrt{\pi}}$$

x + y as plain text.

$$\frac{P_n^{(m)}(\cos(\theta))}{2\sqrt{\pi}} \sqrt{\frac{(-m+n)!}{(m+n)!}} (2n+1)e^{im\phi}$$

## 8 Line Length

```
1 3 5 7 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54 57 60 63 66 69 72 75
↳ 78 81 84 87 90 93 96 99 103
```

File "", line 1 1 3 5 7 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54 57 60 63 66 69 72 75 78  
81 84 87 90 93 96 99 103 ^ SyntaxError: invalid syntax