Jupyter notebook custom conversion

Romain Madar

August 2018

Contents

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.
```

```python
import numpy as np

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

```
[[0.41844197 0.3667863  0.67275157 0.72439104 0.22501114 0.56419446
  0.92255218 0.39512307 0.26107298 0.82009478]
 [0.07696763 0.59596815 0.41978192 0.62461668 0.97560238 0.21585364
  0.18642731 0.7779861  0.25688042 0.19752234]
 [0.52065779 0.57688322 0.27025442 0.91467755 0.87231601 0.65652004
  0.81597039 0.87246406 0.80126596 0.95663384]
 [0.91318523 0.9079086  0.63872928 0.01411788 0.99114851 0.49813483
  0.99859485 0.5816624  0.47847939 0.91992717]
 [0.03916907 0.11878512 0.94404829 0.83577496 0.03285869 0.29221238
  0.2642943  0.62746984 0.18154101 0.01057717]
 [0.79918799 0.43523488 0.5875705  0.03507956 0.55972891 0.05432014
  0.64643669 0.1845513  0.46260932 0.47196247]
 [0.82395535 0.59212066 0.06833566 0.0025567  0.18953787 0.22702943
  0.95380079 0.96422374 0.38692497 0.21316336]
 [0.13520458 0.64980056 0.86318005 0.12036572 0.75131895 0.89279348
  0.54597081 0.42895965 0.15615489 0.05467792]
 [0.31215565 0.4052485  0.23077588 0.66183158 0.67647907 0.72088527
  0.98558451 0.0945901  0.12819637 0.34879391]
 [0.55417112 0.36654638 0.54949126 0.76853267 0.09254626 0.79564119
  0.50903894 0.22208267 0.71377305 0.44407701]]
```

```
array([[0.41844197, 0.3667863 , 0.67275157, 0.72439104, 0.22501114,
        0.56419446, 0.92255218, 0.39512307, 0.26107298, 0.82009478],
       [0.07696763, 0.59596815, 0.41978192, 0.62461668, 0.97560238,
        0.21585364, 0.18642731, 0.7779861 , 0.25688042, 0.19752234],
       [0.52065779, 0.57688322, 0.27025442, 0.91467755, 0.87231601,
        0.65652004, 0.81597039, 0.87246406, 0.80126596, 0.95663384],
       [0.91318523, 0.9079086 , 0.63872928, 0.01411788, 0.99114851,
        0.49813483, 0.99859485, 0.5816624 , 0.47847939, 0.91992717],
       [0.03916907, 0.11878512, 0.94404829, 0.83577496, 0.03285869,
        0.29221238, 0.2642943 , 0.62746984, 0.18154101, 0.01057717],
       [0.79918799, 0.43523488, 0.5875705 , 0.03507956, 0.55972891,
        0.05432014, 0.64643669, 0.1845513 , 0.46260932, 0.47196247],
       [0.82395535, 0.59212066, 0.06833566, 0.0025567 , 0.18953787,
  0.22702943, 0.95380079, 0.96422374, 0.38692497, 0.21316336],
 [0.13520458, 0.64980056, 0.86318005, 0.12036572, 0.75131895,
  0.89279348, 0.54597081, 0.42895965, 0.15615489, 0.05467792],
 [0.31215565, 0.4052485 , 0.23077588, 0.66183158, 0.67647907,
  0.72088527, 0.98558451, 0.0945901 , 0.12819637, 0.34879391],
 [0.55417112, 0.36654638, 0.54949126, 0.76853267, 0.09254626,
  0.79564119, 0.50903894, 0.22208267, 0.71377305, 0.44407701]])
```

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 showns in this figure.

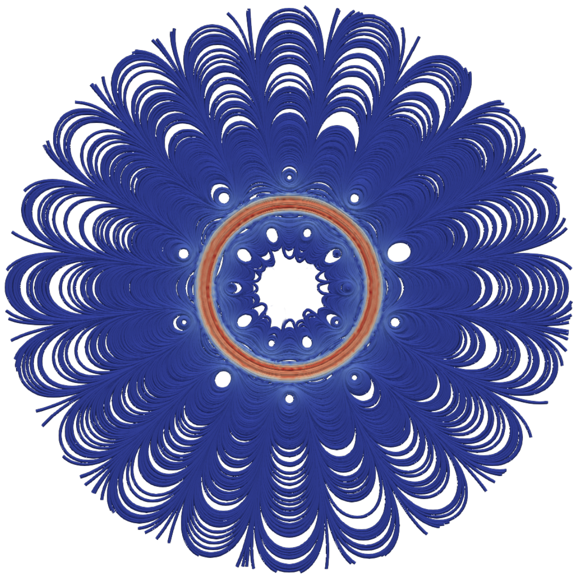

Figure 1: My legend

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

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)
```

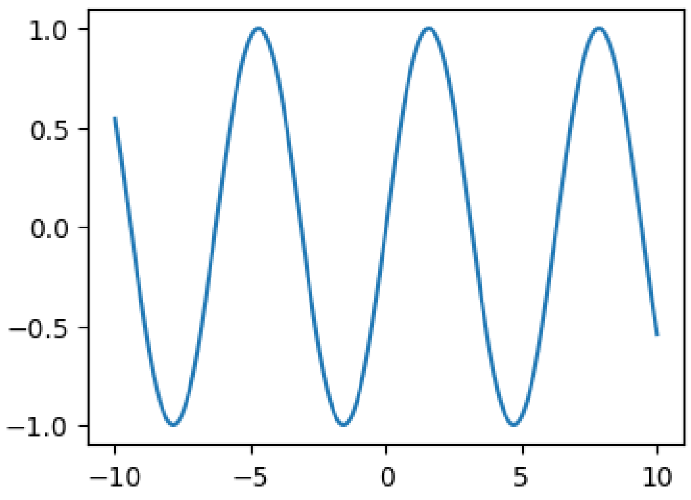

Figure 3: png

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

```
'/mnt/WorkRomain/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%')
```

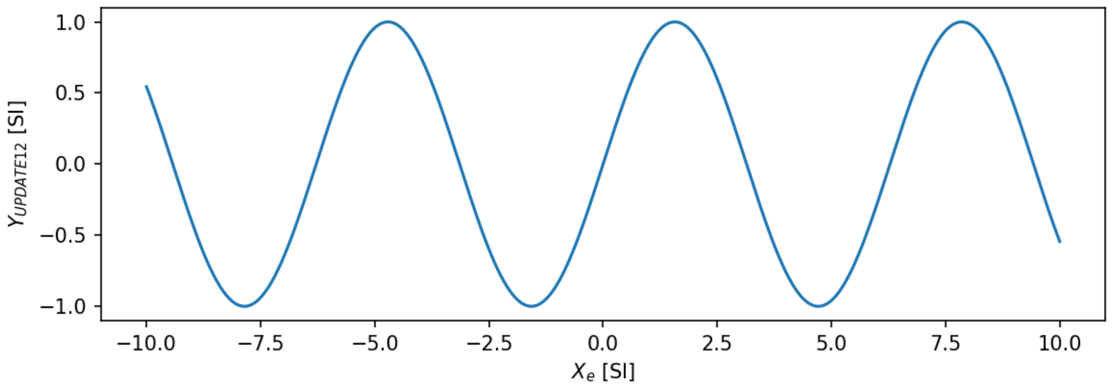

Figure 4: This is a test of how to get properplot from Jupyter notebook in MD,to be processed using PANDOC

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%')
```

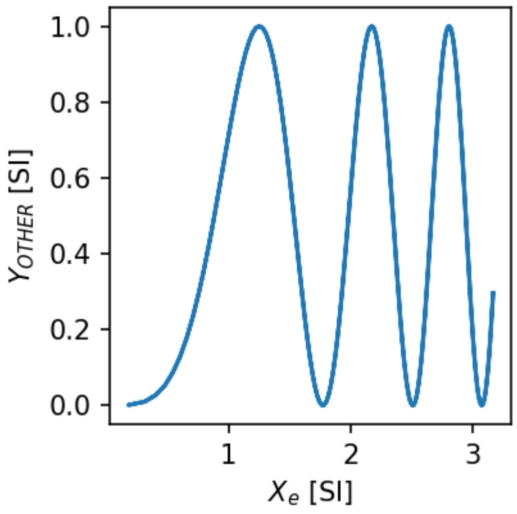

Figure 5: This is another test of how to get properplot from Jupyter notebook in MD,to be processed using PANDOC

```
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%')
```

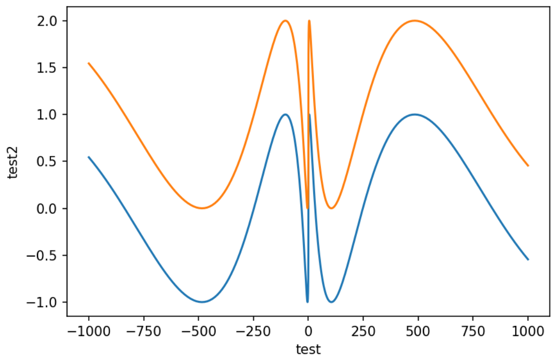

Figure 6: Adding more plot without re-creating a figure: curves cumulates

```
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%')
```

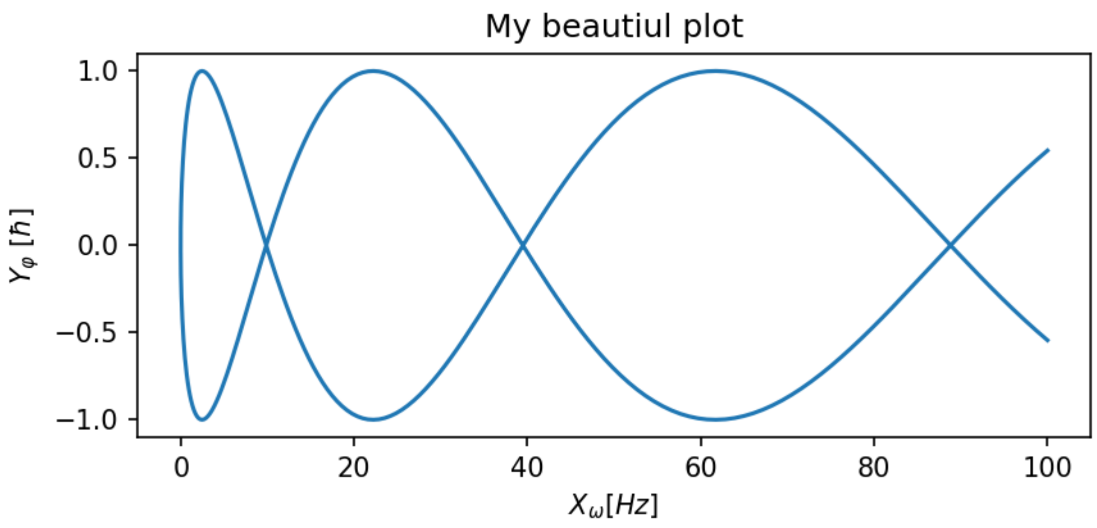

Figure 7: More plot with re-creating a figure: only last curve

5 Operator Highlighing 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:

Table 1: my caption

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. This allows to set caption and even to refer to the table in the main document like this (using vanilla pandoc) or cited like table Tbl. 2 (using pandoc-crossref filter, but hyperlink doesn't seem to work in HTML though)

```
# Good in HTML, but not pure markdown
# Impossible to put a caption
df.describe()
```

0

1

2

count

10.000000

10.000000

10.000000

mean

-0.051856

0.363273

0.418474

std

0.506242

0.846696

1.235089

min

-0.801585

-0.953806

-1.408887

25%

-0.318885

-0.242287

-0.319654

50%

-0.133994

0.529214

0.330189

75%

0.129454

1.012701

0.865129

max

0.980675

1.609233

2.810615

```
# Good in pure MD and possible to put a caption and a label
jpu.df2md(df.describe(),'Caption table. {#tbl:label2}')
```

Table 2: Caption table.

labels	0	1	2
count	10	10	10
mean	-0.0518561	0.363273	0.418474
std	0.506242	0.846696	1.23509
min	-0.801585	-0.953806	-1.40889
25%	-0.318885	-0.242287	-0.319654
50%	-0.133994	0.529214	0.330189
75%	0.129454	1.0127	0.865129
max	0.980675	1.60923	2.81062

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))
```

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

x + y as plain text.

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

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
```

```
import os
```

```
help(os.rename)
```

Help on built-in function rename in module posix:

rename(src, dst, *, src_dir_fd=None, dst_dir_fd=None) Rename a file or directory.

If either src_dir_fd or dst_dir_fd is not None, it should be a file descriptor open to a directory, and the respective path string (src or dst) should be relative; the path will then be relative to that directory. src_dir_fd and dst_dir_fd, may not be implemented on your platform. If they are unavailable, using them will raise a NotImplementedError.