
rsmf

Release 0.2

May 18, 2021

Contents:

1	How to use rsmf	1
1.1	Setup	1
1.2	Figures	2
1.3	Using rsmf with other frameworks	3
2	rsmf package	5
2.1	Submodules	5
2.2	rsmf.abstract_formatter module	5
2.3	rsmf.custom_formatter module	6
2.4	rsmf.fontsizes module	6
2.5	rsmf.quantumarticle module	7
2.6	rsmf.revtex module	8
2.7	rsmf.revtexlike module	8
2.8	rsmf.setup module	9
2.9	Module contents	9
3	Indices and tables	11
	Python Module Index	13
	Index	15

1.1 Setup

You need to tell rsmf how you set up your document by invoking `rsmf.setup`. This can be done in two ways. Either, you give rsmf the `\documentclass` string used for setting up the document, as in

```
import rsmf
formatter = rsmf.setup(r"\documentclass[a4paper,12pt,noarxiv]{quantumarticle}")
```

The `r` in front of the string is necessary so that `\d` is not mistaken for an escape sequence. If you have your document stored locally, there is an even more convenient way: you can just supply rsmf with the path to your main tex file (the one containing the document setup) and it will find that out for itself:

```
formatter = rsmf.setup("example.tex")
```

This is especially cool because rsmf will automatically adjust the plots when the underlying document class is changed without any needs to change python code! This makes swapping journals a lot easier.

1.1.1 Custom

If the document class you're preparing figures for is not supported by rsmf you can still use it to prepare your figures. In this case you will have to measure the column widths yourself. To do so, you have to insert the following command into your text

```
\begin{figure}
  \the\columnwidth
\end{figure}
```

This will give you the width of a single-column figure. If your document class also supports two-column mode, you also need to extract the width of wide figures via

```
\begin{figure*}
  \the\columnwidth
\end{figure*}
```

Both commands will output something along the lines of 246.0pt. As `matplotlib` expects measurements to be in inches, `rsmf` does too. You therefore have to multiply the measurement in points with 0.01389 to get the correct measurement in inches.

It is also important to see if your document class loads packages that change the rendering of fonts, e.g. `\usepackage{times}`. If this is the case, you have to provide them as a separate preamble that is then used in the PGF backend.

With these informations at hand, you can invoke `rsmf`'s `CustomFormatter`:

```
from rsmf import CustomFormatter

formatter = CustomFormatter(
    columnwidth=246 * 0.01389,
    wide_columnwidth=512 * 0.01389,
    fontsizes=11,
    pgf_preamble=r"\usepackage{lmodern}",
)
```

1.2 Figures

The setup routine will return a formatter. This formatter can then be used to create `matplotlib` figure objects by invoking the method `formatter.figure`. It has three arguments:

- `aspect_ratio` (float, optional): the aspect ratio (width/height) of your plot. Defaults to the golden ratio.
- `width_ratio` (float, optional): the width of your plot in multiples of `\columnwidth`. Defaults to 1.0.
- **`wide` (bool, optional): indicates if the figures spans two columns in `twocolumn` mode, i.e. if the `figure*` environment is used, has no effect in `onecolumn` mode . Defaults to False.**

This is the place where you set the width of your plots, *not in the LaTeX document*. If you include the resulting figure with a different width, the font sizes will not match the surrounding document!

For example, a regular figure is created via

```
fig = formatter.figure(aspect_ratio=.5)

# ... some plotting ...
plt.savefig("example.pdf")
```

and included via

```
\begin{figure}
  \centering
  \includegraphics{example}
  \caption{...}
\end{figure}
```

A wide figure that spans 80% of the page on the other hand is created by

```
fig = formatter.figure(width_ratio=.8, wide=True)

# ... some plotting ...
plt.savefig("example_wide.pdf")
```

and included via the multi-column `figure*` environment:

```
\begin{figure*}
  \centering
  \includegraphics{example_wide}
  \caption{...}
\end{figure*}
```

Note that you should always save your figures in some sort of vectorized format, like pdf and that calling `plt.tight_layout()` before saving usually makes your plots nicer.

1.2.1 Custom

If you want more control about the creation of your figure, you can make use of `formatter.columnwidth` and `formatter.wide_columnwidth` to create them yourself.

1.2.2 Other features

You can access the underlying font sizes via `formatter.fontsizes`. The nomenclature follows that of LaTeX itself, so we have

```
formatter.fontsizes.tiny
formatter.fontsizes.scriptsize
formatter.fontsizes.footnotesize
formatter.fontsizes.small
formatter.fontsizes.normalsize
formatter.fontsizes.large
formatter.fontsizes.Large
formatter.fontsizes.LARGE
formatter.fontsizes.huge
formatter.fontsizes.Huge
```

This is especially useful if you want to tweak titles, legends and annotations while still having proper (LaTeX) font-sizes.

1.3 Using rsmf with other frameworks

You can use rsmf together with your favorite plotting framework, for example `seaborn`. There is only one catch: if you use matplotlib styles or seaborn styles, you might overwrite the settings imposed by rsmf, especially regarding font-size. To this end, the formatters have a method `formatter.set_default_fontsizes` that only change the underlying font sizes. An example use would be

```
fig = formatter.figure(wide=True)
sns.set(style="ticks")
formatter.set_default_fontsizes()

# ... some plotting ...
```

Sometimes these styles also overwrite other things, like the font family (serif/sans-serif). There is no correction method for that yet.

2.1 Submodules

2.2 rsmf.abstract_formatter module

Abstract base class for all formatters.

class rsmf.abstract_formatter.**AbstractFormatter**

Bases: abc.ABC

Base class for formatter implementations.

columnwidth

columnwidth of the document.

figure (*aspect_ratio=0.6172839506172839, width_ratio=1.0, wide=False*)

Sets up the plot with the fitting arguments so that the font sizes of the plot and the font sizes of the document are well aligned.

Parameters

- **aspect_ratio** (*float, optional*) – the aspect ratio (width/height) of your plot. Defaults to the golden ratio.
- **width_ratio** (*float, optional*) – the width of your plot in multiples of column-width. Defaults to 1.0.
- **wide** (*bool, optional*) – indicates if the figures spans two columns in twocolumn mode, i.e. if the figure* environment is used, has no effect in onecolumn mode. Defaults to False.

Returns The matplotlib Figure object

Return type matplotlib.Figure

fontsizes

Fontsizes as specified by the underlying document.

set_default_fontsizes()
Adjust the fontsizes in rcParams to the default values matching the surrounding document.

set_rcParams()
Adjust the rcParams to the default values.

wide_columnwidth
Wide columnwidth of the document.

wide_width
Wide columnwidth of the document. (Deprecated)

width()
columnwidth of the document. (Deprecated)

2.3 rsmf.custom_formatter module

Custom formatter that can be used if the intended document class is not supported.

```
class rsmf.custom_formatter.CustomFormatter (columnwidth=None,  
                                             wide_columnwidth=None,   fontsizes=10,  
                                             pgf_preamble="")
```

Bases: *rsmf.abstract_formatter.AbstractFormatter*

Allows to use rsmf even if the intended document class is not supported.

Parameters

- **columnwidth** (*float, optional*) – Width of a single column (figure) plot in inches. Defaults to None.
- **wide_columnwidth** (*float, optional*) – Width of a two column (figure*) plot in inches. Defaults to width.
- **fontsizes** (*Union[int, Fontsizes], optional*) – Latex base fontsize or Font-sizes object. Defaults to 10.
- **pgf_preamble** (*str, optional*) – Additional packages to include in the PGF preamble, e.g. for exchanging fonts or defining commands. Defaults to “”.

columnwidth
columnwidth of the document.

fontsizes
Fontsises as specified by the underlying document.

set_rcParams()
Adjust the rcParams to the default values.

wide_columnwidth
Wide columnwidth of the document.

2.4 rsmf.fontsizes module

Provides an encapsulation for the different named fontsizes in LaTeX.

```
rsmf.fontsizes.DEFAULT_FONTSTIZES = {10: <rsmf.fontsizes.Fontsizes object>, 11: <rsmf.font  
Default fontsize palettes for a given normal size.
```

```
rsmf.fontsizes.DEFAULT_FONTSIZES_10 = <rsmf.fontsizes.Fontsizes object>
    Default fontsize palette based on normal size 10.

rsmf.fontsizes.DEFAULT_FONTSIZES_11 = <rsmf.fontsizes.Fontsizes object>
    Default fontsize palette based on normal size 11.

rsmf.fontsizes.DEFAULT_FONTSIZES_12 = <rsmf.fontsizes.Fontsizes object>
    Default fontsize palette based on normal size 12.

class rsmf.fontsizes.Fontsizes (tiny=5, scriptsize=7, footnotesize=8, small=9, normalsize=10,
                                large=12, Large=14, LARGE=17, huge=20, Huge=25)
    Bases: object
    Encapsulates the standard named font sizes used in LaTeX.
    Defaults to the default LaTeX font sizes based on normal size 10.
```

Parameters

- **tiny** (*int, optional*) – Tiny text. Defaults to 5.
- **scriptsize** (*int, optional*) – Scriptsize text. Defaults to 7.
- **footnotesize** (*int, optional*) – Footnotesize text. Defaults to 8.
- **small** (*int, optional*) – Small text. Defaults to 9.
- **normalsize** (*int, optional*) – Normal text. Defaults to 10.
- **large** (*int, optional*) – Larger text. Defaults to 12.
- **Large** (*int, optional*) – Even larger text. Defaults to 14.
- **LARGE** (*int, optional*) – Even more large text. Defaults to 17.
- **huge** (*int, optional*) – Huge text. Defaults to 20.
- **Huge** (*int, optional*) – Even more huge text. Defaults to 25.

2.5 rsmf.quantumarticle module

Support for the quantumarticle documentclass of Quantum journal.

```
class rsmf.quantumarticle.QuantumarticleFormatter (columns='twocolumn',          pa-
                                                    per='a4paper',          fontsize=10,
                                                    **kwargs)
```

Bases: *rsmf.revtextlike.RevtexLikeFormatter*

Sets up the plot with the fitting arguments so that the font sizes of the plot and the font sizes of the document are well aligned.

Parameters

- **columns** (*str, optional*) – the columns you used to set up your quantumarticle, either “onecolumn” or “twocolumn”. Defaults to “twocolumn”.
- **paper** (*str, optional*) – the paper size you used to set up your quantumarticle, either “a4paper” or “letterpaper”. Defaults to “a4paper”.
- **fontsize** (*int, optional*) – the fontsize you used to set up your quantumarticle, either 10, 11 or 12. Defaults to 10.

colors

Named colors for Quantumarticle. Contains quantumviolet and quantumgray.

```
set_rcParams ()
```

Adjust the rcParams to the default values for Quantumarticle.

```
rsmf.quantumarticle.quantumarticle_parser = <rsmf.revtextlike.RevtextLikeParser object>
```

Parser for the quantumarticle document class.

2.6 rsmf.revtext module

Support for the revtex4-1 and revtex4-2 documentclasses of the APS journals.

```
class rsmf.revtext.RevtextFormatter (columns='twocolumn', fontsize=10, **kwargs)
```

Bases: *rsmf.revtextlike.RevtextLikeFormatter*

Sets up the plot with the fitting arguments so that the font sizes of the plot and the font sizes of the document are well aligned.

Parameters

- **columns** (*str, optional*) – the columns you used to set up your quantumarticle, either “onecolumn” or “twocolumn”. Defaults to “twocolumn”.
- **fontsize** (*int, optional*) – the fontsize you used to set up your quantumarticle, either 10, 11 or 12. Defaults to 10.

```
set_rcParams ()
```

Adjust the rcParams to the default values for Revtext.

```
rsmf.revtext.revtext_parser = <rsmf.revtextlike.RevtextLikeParser object>
```

Parser for the revtext document classes.

2.7 rsmf.revtextlike module

Base for implementations of document classes alike to revtex.

```
class rsmf.revtextlike.RevtextLikeFormatter (columns, paper, fontsize)
```

Bases: *rsmf.abstract_formatter.AbstractFormatter*

A generic Formatter for revtex-like journals.

Inherited classes should set the following: self._columnwidths self._wide_columnwidths

columnwidth

columnwidth of the document.

fontsizes

Fontsizes as specified by the underlying document.

wide_columnwidth

Wide columnwidth of the document.

```
class rsmf.revtextlike.RevtextLikeParser (documentclass_identifiers, formatter_class)
```

Bases: *object*

Generic parser for revtex-like document classes.

Parameters

- **documentclass_identifiers** (*List[string]*) – Strings identifying the supported document class.

- **formatter_class** (*class*) – Class object describing the formatter.

2.8 rsmf.setup module

Main routines to invoke the module from code.

`rsmf.setup.setup` (*arg*)

Get a formatter corresponding to the document's class.

Parameters *arg* (*str*) – Either path to a tex file or preamble of a tex file, containing at least the `documentclass` command.

Raises `Exception` – When no formatter for the given document was found.

Returns A formatter for the given document/preamble.

Return type object

2.9 Module contents

Load all methods that are made accessible as the API.

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

r

- rsmf, 9
- rsmf.abstract_formatter, 5
- rsmf.custom_formatter, 6
- rsmf.fontsizes, 6
- rsmf.quantumarticle, 7
- rsmf.revtex, 8
- rsmf.revtexlike, 8
- rsmf.setup, 9

A

AbstractFormatter (class in *rsmf.abstract_formatter*), 5

C

colors (*rsmf.quantumarticle.QuantumarticleFormatter* attribute), 7
columnwidth (*rsmf.abstract_formatter.AbstractFormatter* attribute), 5
columnwidth (*rsmf.custom_formatter.CustomFormatter* attribute), 6
columnwidth (*rsmf.revtextlike.RevtextLikeFormatter* attribute), 8
CustomFormatter (class in *rsmf.custom_formatter*), 6

D

DEFAULT_FONTSIZES (in module *rsmf.fontsizes*), 6
DEFAULT_FONTSIZES_10 (in module *rsmf.fontsizes*), 6
DEFAULT_FONTSIZES_11 (in module *rsmf.fontsizes*), 7
DEFAULT_FONTSIZES_12 (in module *rsmf.fontsizes*), 7

F

figure() (*rsmf.abstract_formatter.AbstractFormatter* method), 5
Fontsizes (class in *rsmf.fontsizes*), 7
fontsizes (*rsmf.abstract_formatter.AbstractFormatter* attribute), 5
fontsizes (*rsmf.custom_formatter.CustomFormatter* attribute), 6
fontsizes (*rsmf.revtextlike.RevtextLikeFormatter* attribute), 8

Q

quantumarticle_parser (in module *rsmf.quantumarticle*), 8

QuantumarticleFormatter (class in *rsmf.quantumarticle*), 7

R

revtex_parser (in module *rsmf.revtext*), 8
RevtexFormatter (class in *rsmf.revtext*), 8
RevtexLikeFormatter (class in *rsmf.revtextlike*), 8
RevtexLikeParser (class in *rsmf.revtextlike*), 8
rsmf (module), 9
rsmf.abstract_formatter (module), 5
rsmf.custom_formatter (module), 6
rsmf.fontsizes (module), 6
rsmf.quantumarticle (module), 7
rsmf.revtext (module), 8
rsmf.revtextlike (module), 8
rsmf.setup (module), 9

S

set_default_fontsizes() (*rsmf.abstract_formatter.AbstractFormatter* method), 5
set_rcParams() (*rsmf.abstract_formatter.AbstractFormatter* method), 6
set_rcParams() (*rsmf.custom_formatter.CustomFormatter* method), 6
set_rcParams() (*rsmf.quantumarticle.QuantumarticleFormatter* method), 7
set_rcParams() (*rsmf.revtext.RevtexFormatter* method), 8
setup() (in module *rsmf.setup*), 9

W

wide_columnwidth (*rsmf.abstract_formatter.AbstractFormatter* attribute), 6
wide_columnwidth (*rsmf.custom_formatter.CustomFormatter* attribute), 6
wide_columnwidth (*rsmf.revtextlike.RevtexLikeFormatter* attribute), 8
wide_width (*rsmf.abstract_formatter.AbstractFormatter* attribute), 6

```
width() (rsmf.abstract_formatter.AbstractFormatter  
        method), 6
```