



Decorator Documentation

Release 3.5

/ELSA/MU-10020/V3.5

Nov 21, 2022

CONTENTS

1	Preamble	1
2	List of functions	3
3	Contents	5
3.1	Actions	5
4	Index	9

PREAMBLE

CPlot can generate images of meshes or flow fields. Those image can be further enhanced with any additional matplotlib items using this module.

This module is part of Cassiopee, a free open-source pre- and post-processor for CFD simulations.

To import CPlot.Decorator module:

```
import CPlot.Decorator as Decorator
```


LIST OF FUNCTIONS

– Actions

<code>CPlot.Decorator.createSubPlot([img, title])</code>	Create a sub plot figure.
<code>CPlot.Decorator.createColorBar(fig, ax[, ...])</code>	Create a color bar.
<code>CPlot.Decorator.createText(ax[, posx, posy, ...])</code>	Create text.
<code>CPlot.Decorator.savefig(fileName[, pad])</code>	Save current figure.
<code>CPlot.Decorator.show()</code>	Show current figure.

CONTENTS

3.1 Actions

`CPlot.Decorator.createSubPlot(title=None)`

Create a matplotlib figure and axe from an image generated by `CPlot.display`. The returned figure and axis can be further modified with any matplotlib command.

Parameters `title` (string or None) – title of subplot

Example of use:

- Create subplot with osmesa (pyTree):

```
# - Decorator (pyTree) -
import CPlot.PyTree as CPlot
import CPlot.Decorator as Decorator
import Generator.PyTree as G
import Converter.PyTree as C
import Converter.Internal as Internal

Decorator.setBatch(True)

a = G.cart((0,0,0), (1,1,1), (10,10,1))
C._initVars(a, '{F} = {CoordinateX}')

CPlot.display(a, mode='Scalar',
              scalarField='F', isoScales=['F',12,0.,10.],
              export=CPlot.decorator, exportResolution='3840x2160', offscreen=1,
              isoEdges=1., colormap=0, bgColor=1)

fig, ax = Decorator.createSubPlot()
ax.set_title('Computation of the year', size=40)
Decorator.createText(ax, 0.02, 0.9, "Fast LES", size=40, box=True)
cbar = Decorator.createColorBar(fig, ax, title=r'$\mu_t / \mu$')
cbar.ax.tick_params(labelcolor='tab:red')
```

(continues on next page)

(continued from previous page)

```
Decorator.savefig('out.png')
```

- Create subplot with openGL (pyTree):

```
# - Decorator (pyTree) -
import CPlot.PyTree as CPlot
import CPlot.Decorator as Decorator
import Generator.PyTree as G
import Converter.PyTree as C
import Converter.Internal as Internal

Decorator.setBatch(True)

a = G.cart((0,0,0), (1,1,1), (11,10,1))
C._initVars(a, '{F} = {CoordinateX}')

CPlot.display(a, mode='scalar',
              scalarField='F', isoScales=['F',12,0.,10.],
              export=CPlot.decorator,
              offscreen=2,
              isoEdges=1., colormap=16, bgColor=1)
CPlot.finalizeExport()

fig, ax = Decorator.createSubPlot()
ax.set_title('Computation of the year', size=40)
Decorator.createText(ax, 0.02, 0.9, "Fast LES", size=40, box=True)
cbar = Decorator.createColorBar(fig, ax, title=r'$\mu_t / \mu$')
cbar.ax.tick_params(labelcolor='tab:red')

Decorator.savefig('out.png')
import os; os._exit(0)
```

`CPlot.Decorator.createColorBar`(*fig*, *ax*, *levels=None*, *title=None*, *cmap=None*,
valueFormat='%0.3f', *discrete=True*)

Create a colorbar on a subplot figure and axe.

Parameters

- **fig** (matplotlib figure) – subplot figure
- **ax** (matplotlib axis) – subplot axis

- **levels** (None or list or numpy) – if given, the list of levels to display in colorbar. If none, levels are taken from CPlot.
 - **title** (string) – title of createColorBar
 - **cmap** (None or string) – colormap name ('Blue2Red', ...). If none, cmap is taken from CPlot.
 - **valueFormat** (string) – format of values in colorbar
 - **discrete** (boolean) – if True, discrete levels else continuous levels
-

CPlot.Decorator.**createText**(*ax, posx=0, posy=0, text="", size=20, box=False*)
Create a text on figure.

Parameters

- **posx** (float) – position of text in axis
 - **posy** (float) – position of text in axis
 - **text** (string) – text to be displayed
 - **size** (int) – size of text
 - **box** (boolean) – if True, display a box around text
-

CPlot.Decorator.**savefig**(*fileName, pad=0.2*)
Save figure un fileName. A padding space can be added.

Parameters

- **fileName** (string) – name of the file to export to
 - **pad** (float) – padding space size
-

CPlot.Decorator.**show**()
Display figure.

CHAPTER
FOUR

INDEX

- genindex
- modindex
- search