Decorator Documentation

Release 3.7

/ELSA/MU-10020/V3.7

Oct 19, 2023

CONTENTS

1 Preamble 1

2 List of functions 3

3 Contents 5
3.1 Actions 5

4 Index 9

CHAPTER

ONE

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
```

CHAPTER

TWO

LIST OF FUNCTIONS

– Actions

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

CHAPTER

THREE

CONTENTS

3.1 Actions

CPlot.Decorator.**createSubPlot**(*title=None*, *box=False*)

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
- **box** (boolean) – if True, display a box around image

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

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)
```

(continues on next page)

(continued from previous page)

```
fig, ax = Decorator.createSubPlot()
ax.set_title('Computation of the year', size=40)
Decorator.createText(ax, "Fast LES", 0.02, 0.9, 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')
```

- 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

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()
Decorator.createText(ax, "Fast LES", 0.4, 0.95, size=40, box=True)
cbar = Decorator.createColorBar(fig, ax, title=r'$\mu_t / \mu$',␣
↪location="right", color="black", fontSize=10, pad=-2.)

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*, *fontSize=20*, *color='black'*, *location='right'*)

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, a list [no, minLevel, maxLevel] 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
- **fontSize** (int) – size of font used in colorbar
- **color** (string) – color used for fonts and ticks
- **location** (string) – location of colorbar in "left", "right", "top", "bottom"

CPlot.Decorator.createText(ax, posx=0, posy=0, text='', size=20, color="black", box=False, box=False, boxColor="black", boxBackColor="white")

Create a text on figure.

Parameters

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

CPlot.Decorator.**savefig**(*fileName*, *pad=0.*)

Save figure in fileName. A padding space can be added.

Parameters

- **fileName** (string) – name of the file to export to

- **pad** (`float`) – padding space around image

`CPlot.Decorator.`**show**`()`

Display figure.

CHAPTER

FOUR

INDEX

- genindex
- modindex
- search