

## Diff:

### Differences between given skeleton and solution

In order to make the sample solution easier to understand, the differences between it and the given skeleton source code were highlighted with the help of the program `diff`.

### Legend:

- Gray: unchanged text (only excerpts).
- Green: new lines
- Yellow: changed lines
- Red: deleted lines

Note: Files not listed have not been changed.

This document was created with the help of [diff2html](#) erstellt.

```
diff -u ../course04-2d-visualization/exercise/code/01_plot.py ../course04-2d-visualization/exercise/solution/01_plot.py
```

../course04-2d-visualization/exercise/code/01_plot.py	../course04-2d-visualization/exercise/solution/01_plot.py
:	:
21 zz0 = measurment_data[:, 0] # first column of measurement data -> initial value of the simulation	21 zz0 = measurment_data[:, 0] # first column of measurement data -> initial value of the simulation
22	22
23 # Create empty list for collecting intermediate optimization results:	23 # Create empty list for collecting intermediate optimization results:
24 resList = XXX	24 resList = []
25	25
26	26
27 # this is copied from exercise 03.2 (+ one line added)	27 # this is copied from exercise 03.2 (+ one line added)
:	:
76 # Exercise 04.1:	76 # Exercise 04.1:
77 # That single line is new:	77 # That single line is new:
78 # save the current simulation result (state array) in the prepared list	78 # save the current simulation result (state array) in the prepared list
79 XXX.append(XXX)	79 resList.append(zz_res)
80	80
81	81
82	82
:	:
113	113
114	114
115## After successful saving, insert a 0 here:	115## After successful saving, insert a 0 here:
116if 1:	116if 0:
117 # Optimization - result are the two parameters m2 and l	117 # Optimization - result are the two parameters m2 and l
118	118
119 # intially guessed values for optimization	119 # intially guessed values for optimization
:	:
143fig = plt.figure(figsize=(300*mm, 170*mm))	143fig = plt.figure(figsize=(300*mm, 170*mm))
144	144
145# subplot for unoptimized curves	145# subplot for unoptimized curves
146ax1 = fig.add_subplot(XXX, XXX, XXX)	146ax1 = fig.add_subplot(3, 1, 1)
147ax1.plot(tt, XXX[XXX, :], label="measurement")	147ax1.plot(tt, measurment_data[0, :], label="measurement")
148ax1.plot(tt, resList[0][XXX, XXX], label="model (initial guess)")	148ax1.plot(tt, resList[0][0, :], label="model (initial guess)")
149ax1.legend()	149ax1.legend()
150ax1.grid()	150ax1.grid()
151ax1.set_ylabel("distance \$x \sim [m]\$")	151ax1.set_ylabel("distance \$x \sim [m]\$")
152	152
153	153
154# subplot for optimization result	154# subplot for optimization result
155ax2 = XXX.add_XXX(XXX, XXX, XXX)	155ax2 = fig.add_subplot(3, 1, 2)
156ax2.plot(XXX, XXX, lw=3, label="measurement")	156ax2.plot(tt, measurment_data[0, :], lw=3, label="measurement")
157# ...	157ax2.plot(tt, resList[-1][0, :], label="model (optimized parameters)")
	158ax2.legend()
	159ax2.grid()
	160ax2.set_ylabel("distance \$x \sim [m]\$")
158	161
159	162
160# subplot for remaining error of x position	163# subplot for remaining error of x position
161# ..	164ax3 = fig.add_subplot(3, 1, 3)
162	165ax3.plot(tt, measurment_data[0, :]-resList[-1][0, :])
	166ax3.grid()
	167ax3.set_xlabel("time \$t \sim [s]\$")
	168ax3.set_ylabel("error \$\epsilon \sim [m]\$")
163	169
164plt.savefig("fig1.pdf", bbox_inches="tight")	170plt.savefig("fig1.pdf", bbox_inches="tight")
165# plt.show()	171# plt.show()
:	:

<pre> 176 colStep = 0.9 / len(resList) 177 gray_value = str(0.01 + (0.9 - i*colStep)) 178 179 plt.plot(XXX, XXX[XXX][XXX, XXX], color=gray_value ) 180 181# Plotting the measurement and the simulation 182plt.plot(XXX, XXX, color="#3366FF", lw=4, label="measurement") 183plt.plot(XXX, XXX[-1][XXX, XXX], color="#FF9900", ls="--", lw=2, label="model (optimized parameters)") 184 185# grid 186# legend 187# x label 188# y label 189 190 191# here we use png because due to the many individual lines the pdf would be quite big and slow 192plt.savefig("fig2.png", bbox_inches="tight") 193 194 211 212# overview on color maps: 213# https://matplotlib.org/examples/color/colormaps_reference.html 214ax.plot_surface(XXX, rstride=1, cstride=1, cmap=cm.XXX) 215ax.set_xlabel("x") 216ax.set_ylabel("y") 217ax.set_zlabel("z") Nur in ../course04-2d-visualization/exercise/solution/: __pycache__. Nur in ../course04-2d-visualization/exercise/solution/: res_list.pcl. </pre>	<pre> 182 colStep = 0.9 / len(resList) 183 gray_value = str(0.01 + (0.9 - i*colStep)) 184 185 plt.plot(tt, resList[i][0, :], color=gray_value ) 186 187# Plotting the measurement and the simulation 188plt.plot(tt, measurment_data[0, :], color="#3366FF", lw=4, label="measurement") 189plt.plot(tt, resList[-1][0, :], color="#FF9900", ls="--", lw=2, label="model (optimized parameters)") 190plt.grid() 191plt.legend() 192plt.xlabel("time \$t \sim [s]\$") 193plt.ylabel("distance \$x \sim [m]\$") 194 195# here we use png because due to the many individual lines the pdf would be quite big and slow 196plt.savefig("fig2.png", bbox_inches="tight") 197 198 215 216# overview on color maps: 217# https://matplotlib.org/examples/color/colormaps_reference.html 218ax.plot_surface(X, Y, Z, rstride=1, cstride=1, cmap=cm.viridis) 219ax.set_xlabel("x") 220ax.set_ylabel("y") 221ax.set_zlabel("z") </pre>
---	---