

Diff: Übung 07b

Unterschiede zwischen Vorgabe und Lösung

Um die Musterlösung leichter nachvollziehbar zu gestalten wurden die Unterschiede zur Quelltext-Vorgabe mit Hilfe des Programms `diff` grafisch aufbereitet.

Legende:

- Grau: unveränderter Text (nur auszugsweise)
- Grün: neue Zeilen
- Gelb: veränderte Zeilen
- Rot: gelöschte Zeilen (kommt hier nicht vor)

Hinweis: nicht aufgeführte Dateien wurden nicht verändert.

Dieses Dokument wurde mit Hilfe von [diff2html](#) erstellt.

diff -u ../kurs07b_Performanceoptimierung/uebung/code/aufgabe1.py ../kurs07b_Performanceoptimierung/uebung/loesung/aufgabe1.py

../kurs07b_Performanceoptimierung/uebung/code/aufgabe1.py	../kurs07b_Performanceoptimierung/uebung/loesung/aufgabe1.py
:	:
123 grid.set_bc_function(bc)	123 grid.set_bc_function(bc)
124 solver = Solver(grid)	124 solver = Solver(grid)
125	125
126 solver.solve(n_iter=n_iter, eps=eps)	126 t = time.clock()
127	127 solver.solve(n_iter=n_iter, eps=eps)
128 def run_visualization(Solver, nx=100, ny=100, n_iter=100):	128 return time.clock() - t
129	129
:	:
155	157
156 times()	158 times()
157 run_visualization(LaplaceSolver)	159 run_visualization(LaplaceSolver)
158 run_time(LaplaceSolver)	160 print run_time(LaplaceSolver)
159	161
160	162
161 if __name__ == "__main__":	163 if __name__ == "__main__":

diff -u ../kurs07b_Performanceoptimierung/uebung/code/aufgabe2.py ../kurs07b_Performanceoptimierung/uebung/loesung/aufgabe2.py

../kurs07b_Performanceoptimierung/uebung/code/aufgabe2.py	../kurs07b_Performanceoptimierung/uebung/loesung/aufgabe2.py
:	:
103 u = g.u	103 u = g.u
104 g.old_u = u.copy()	104 g.old_u = u.copy()
105	105
106 # Hier u-update Implementieren	106 u[1:-1, 1:-1] = ((u[0:-2, 1:-1] + u[2:, 1:-1])*dy2 +
107	107 (u[1:-1, 0:-2] + u[1:-1, 2:])*dx2)*dnr_inv
108 return g.compute_error()	108
109	109 return g.compute_error()
:	110
120 u = g.u	:
121 g.old_u = u.copy()	121 u = g.u
122	122 g.old_u = u.copy()
123 # Hier u-update aus NumpySolver als String eintragen.	123
124 expr = ""	124 expr = "u[1:-1, 1:-1] = ((u[0:-2, 1:-1] + u[2:, 1:-1])*dy2 + "\
125	125 "(u[1:-1, 0:-2] + u[1:-1, 2:])*dx2)*dnr_inv"
126 weave.blitz(expr, check_size=0)	126 weave.blitz(expr, check_size=0)
127	127
128 return g.compute_error()	128 return g.compute_error()