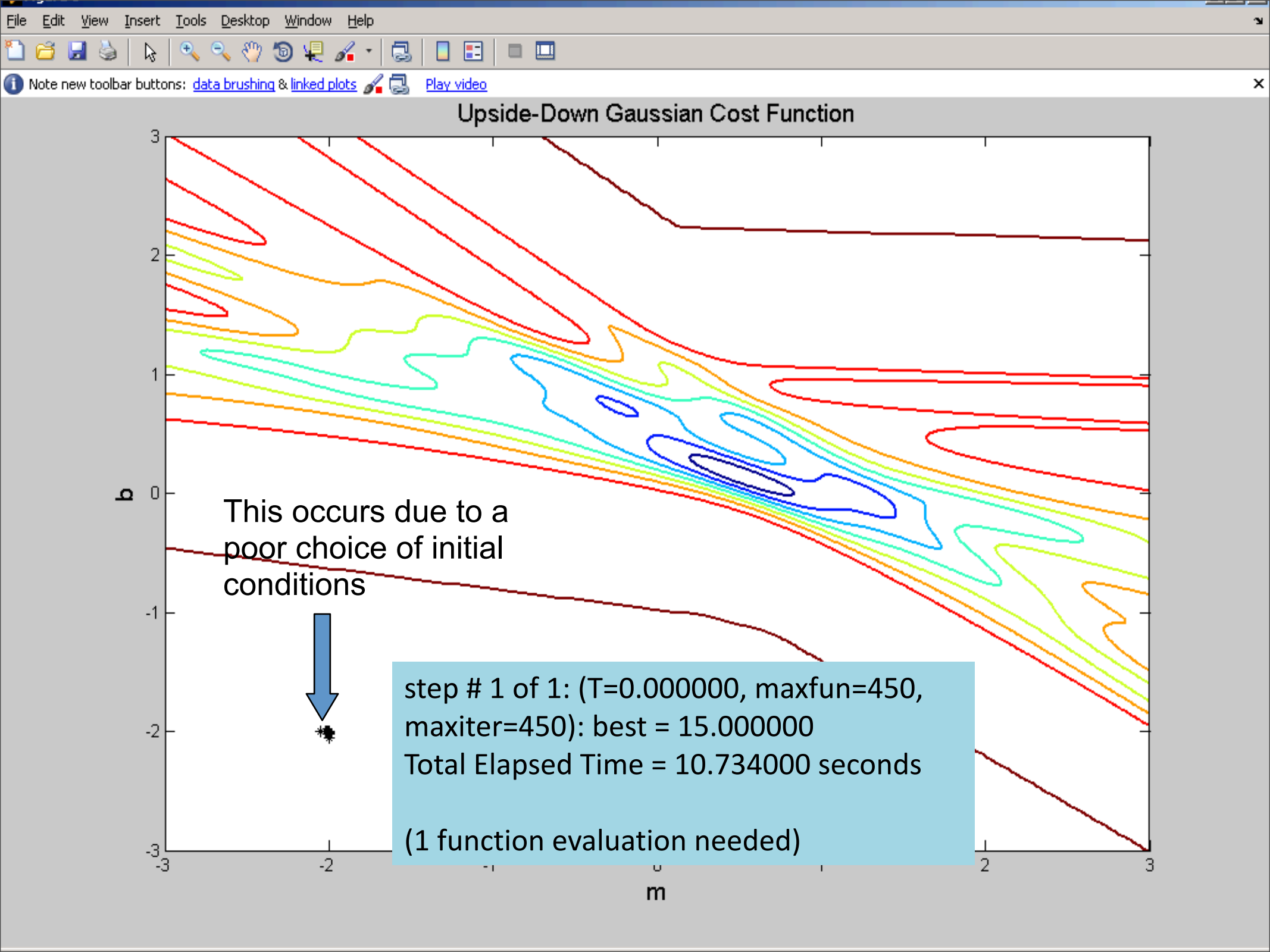
Lab W

Optimization


File Edit View Insert Tools Desktop Window Help
Note new toolbar buttons: data brushing & linked plots Play video
Upside-Down Gaussian Cost Function
This occurs due to a poor choice of initial conditions
step # 1 of 1: (T=0.000000, maxfun=450, maxiter=450): best = 15.000000
Total Elapsed Time = 10.734000 seconds
(1 function evaluation needed)
b
m
3
2
1
0
-1
-2
-3

I lowered the deviation to dev = 0.075 to generate a more difficult problem for the UG cost function.

tfactor 0.9 = 11.28487

Even when I slowed down the tfactor significantly, it did not generate a good estimate. (In fact, it was worse -- tfactor 0.5 = 12.87417)

When I assumed any of the starting origin coordinate point to be a 0, it produced a very awful optimization behavior when observed via animation. Instead of going to the desired point, it actually just jumped around either plotting a horizontal line (when b_o = 0) or a vertical line (when m_o = 0). This occurs because of how the program code was set up.

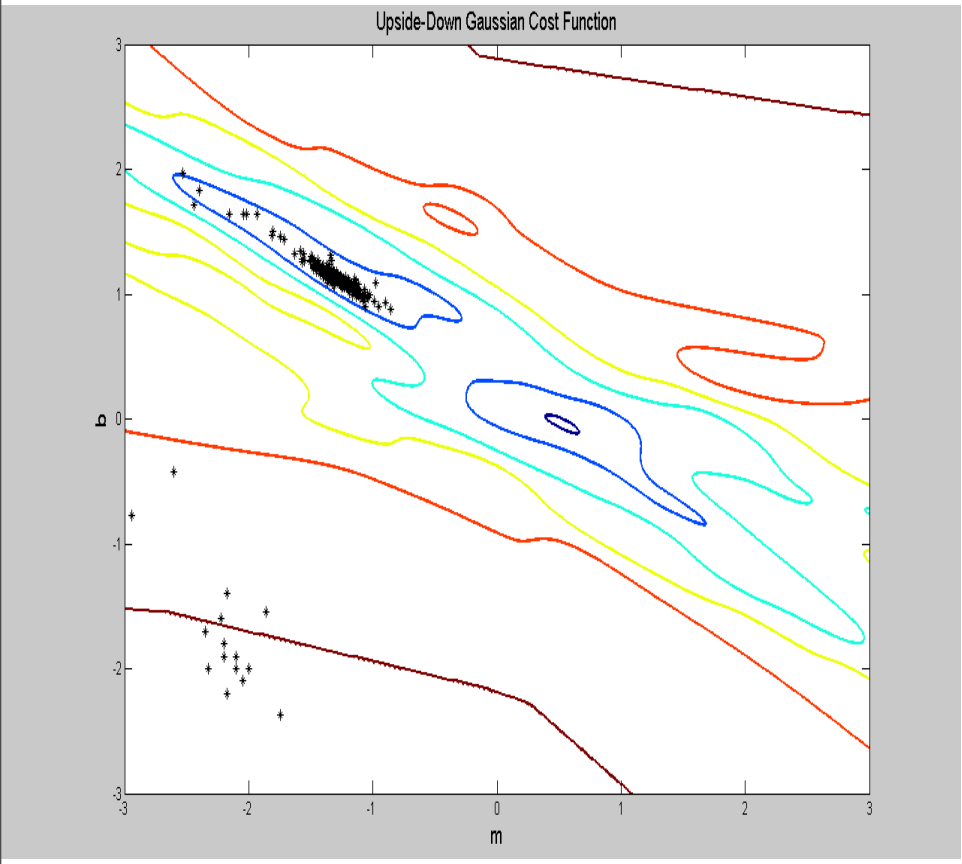
Homework W.3: Use UG cost function to come up with a hard problem for optimization. Run Anneal with quench and 3 cooling schedules (fast, medium, and slow) and compare results. The results will depend on your problem.

- When tfactor was 0.1, best = 16.291807
- When tfactor was 0.5, best = 16.291807
- When tfactor was 0.9, best = 11.751268
- The slower the cooling schedule is, the less accurate the annealing becomes.

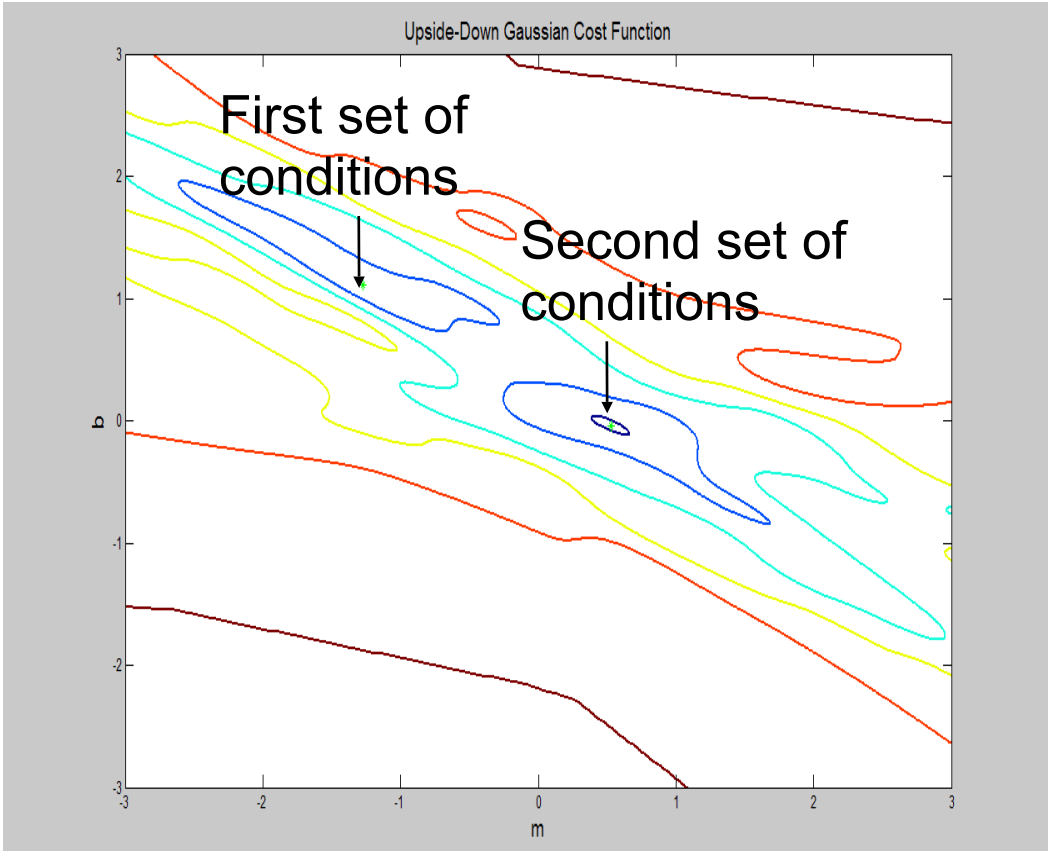
Results of 3 different cooling schedules

- Fast t=0.1; best = 40.379886
- Medium t=0.5; best = 40.379886
- Slow t=0.9; best = 40.379886
- So I guess the cooling schedules for my problem are relatively slow, because all three conditions have the same objective best values.

W.3(2) – 3 Annealing rates



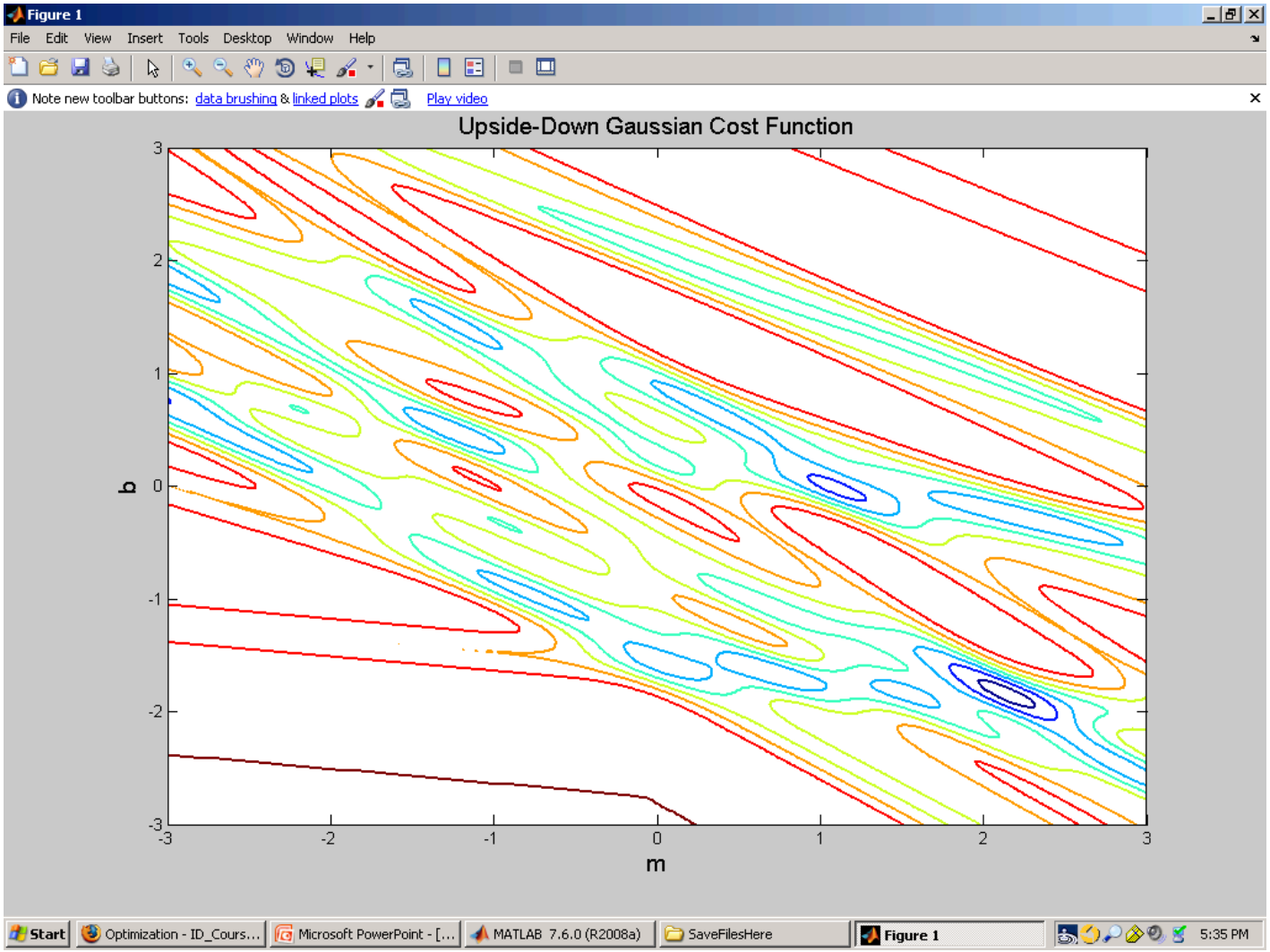
Default altered rates (from the example) tinitial=2, etc.



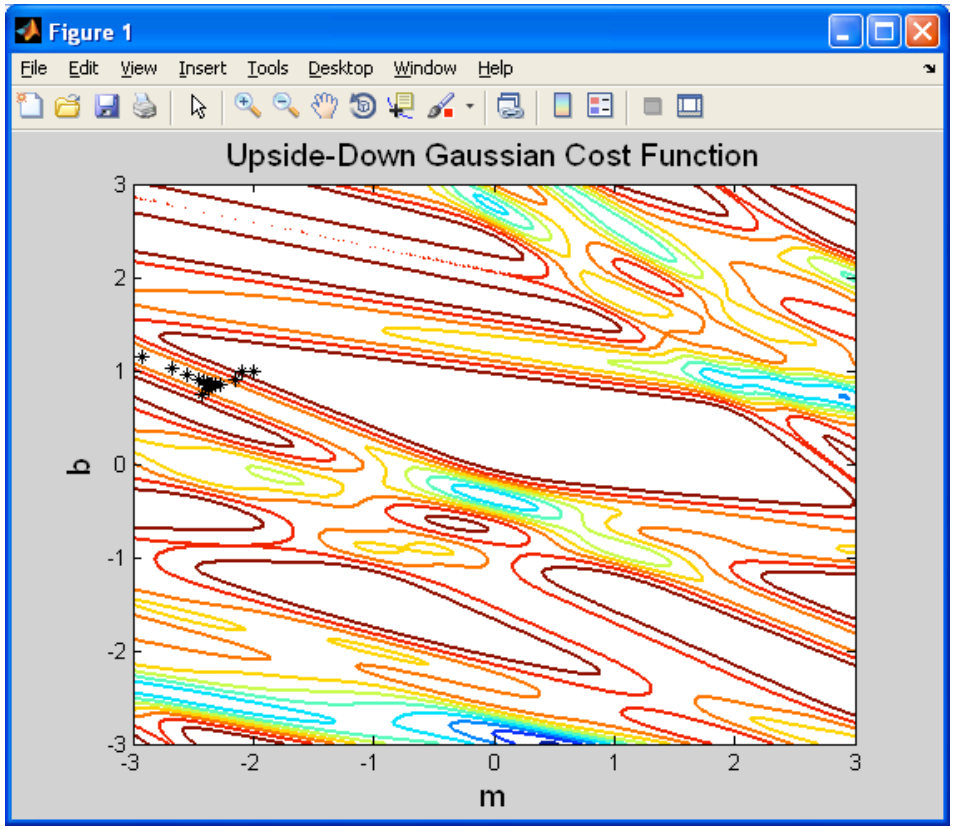
Due to the time required, I just ran it without counting steps, and plotted the result. Here tinitial was set to 20, tfactor = .025, numsteps = 100. (much faster than the rate on the left.)

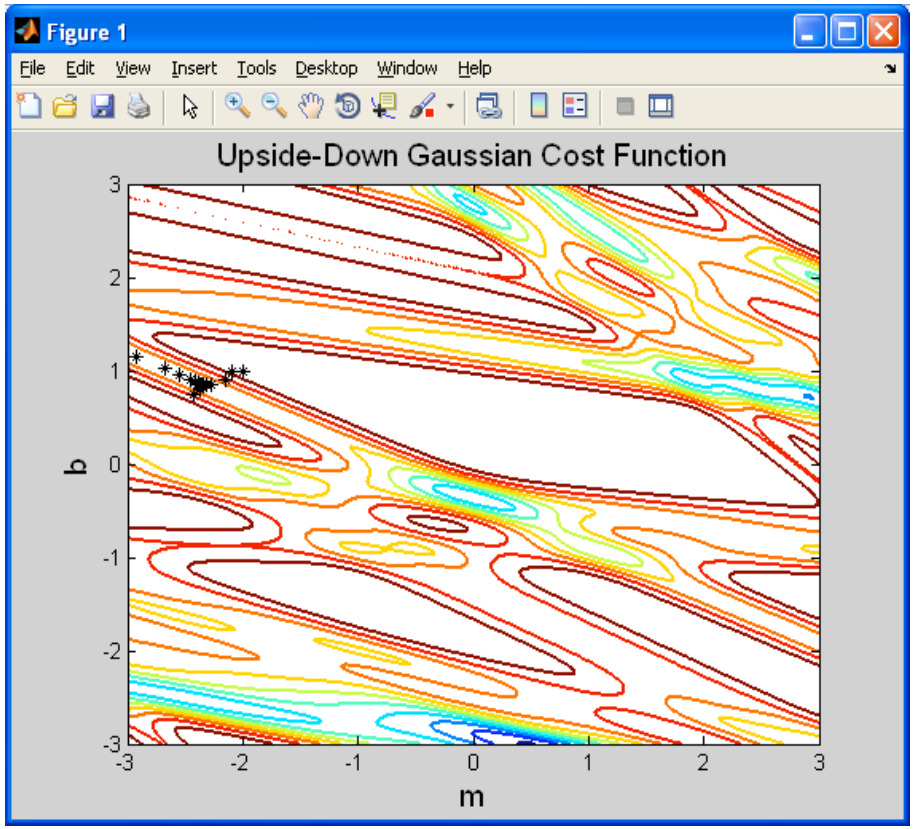
In the second set, tinitial was set to 0.1, but tfactor =0.99. This much slower rate allowed for the correct peak to be identified.

Homework W.3
Creating Hard Optimization



Q3





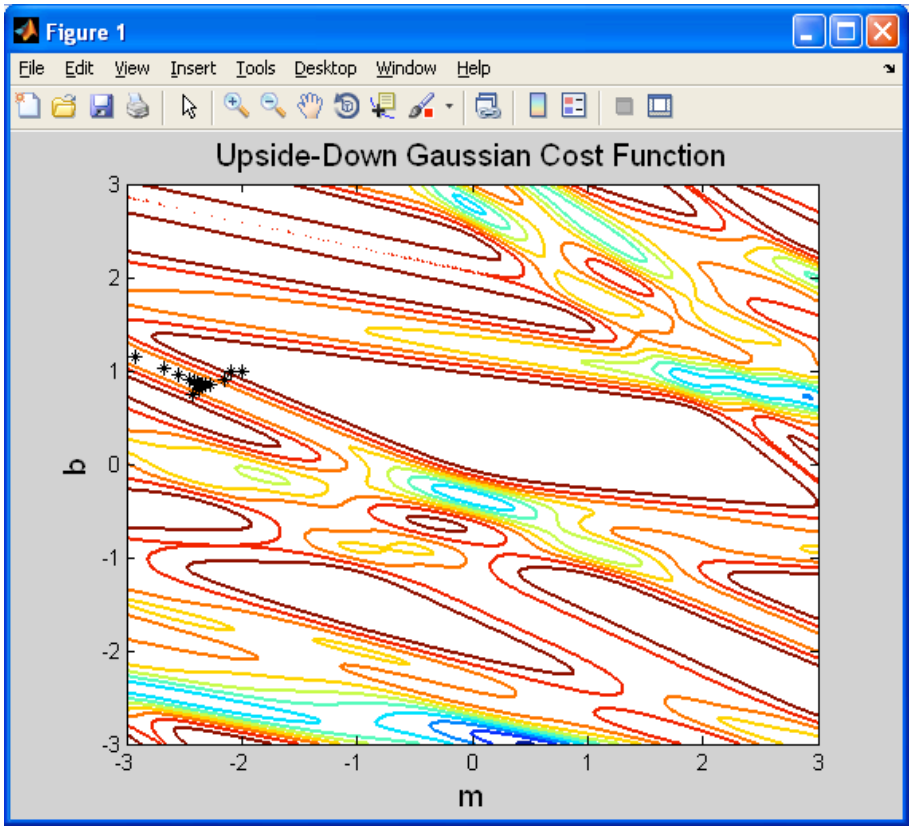
Annealing with quench

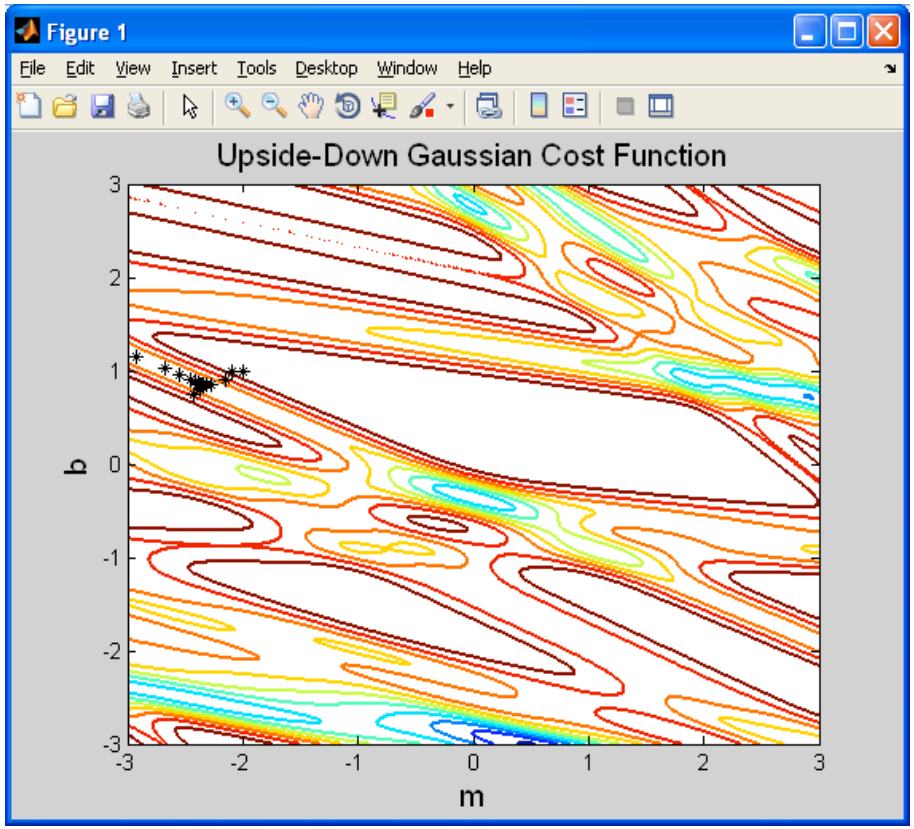
best = 46.745416

Annealing (fast): tfactor = .9

Best =
46.745416
Begin = 5

Q3





Annealing (fast): tfactor = .5

Best =
45.569768
Begin = 5

Annealing (fast): tfactor = .1

Best =
45.569768
Begin = 5

W.2

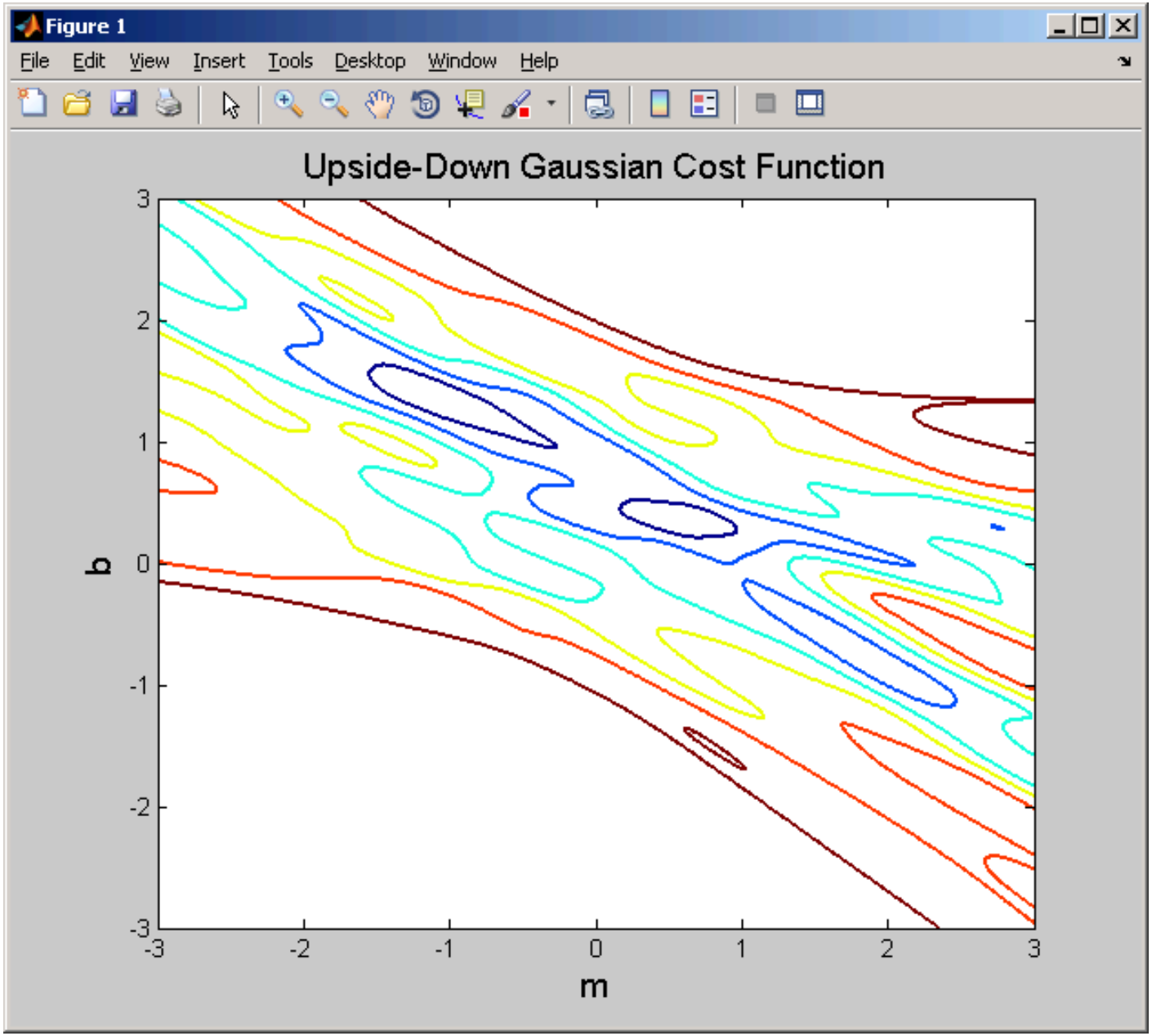
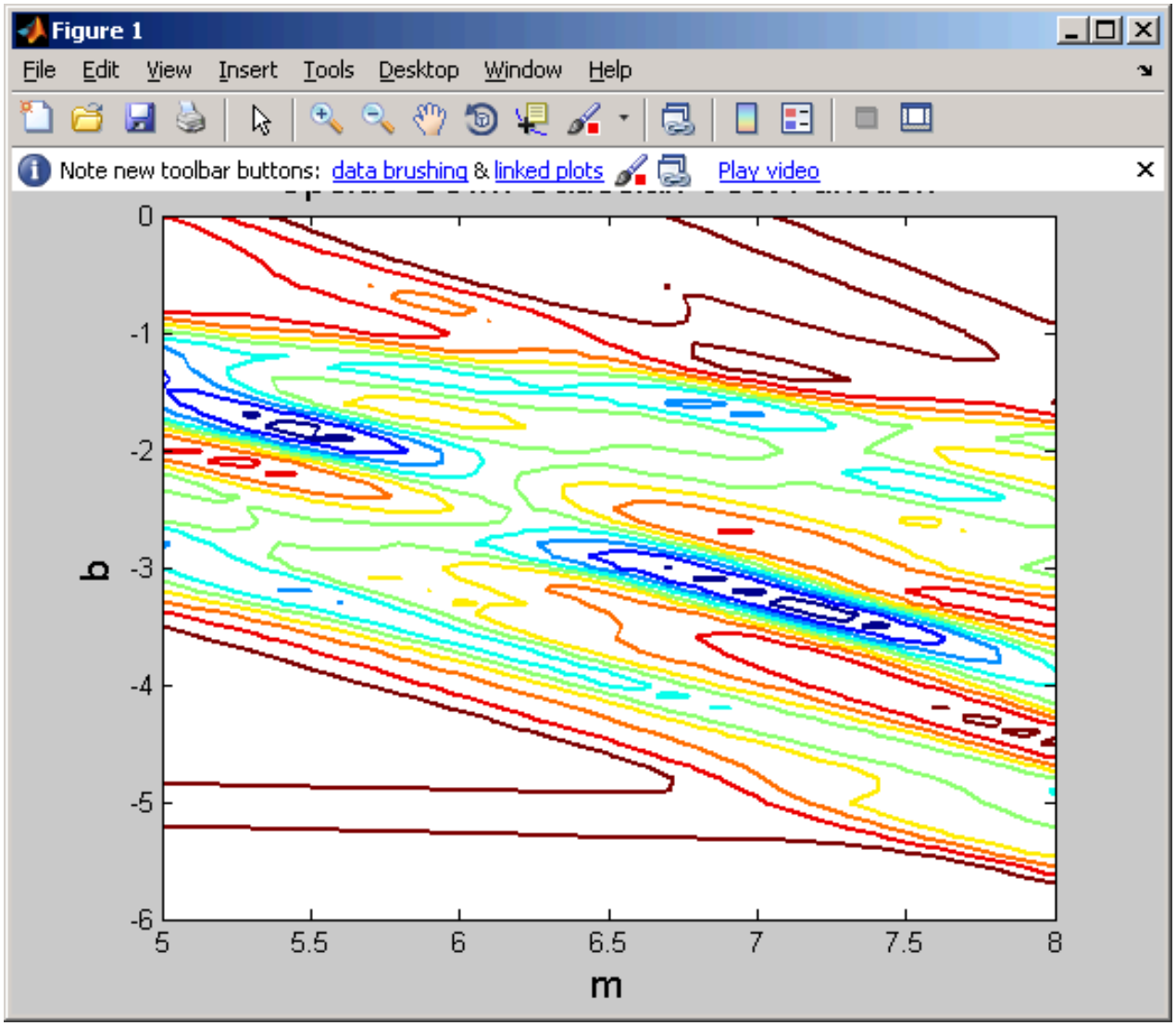


Figure 1
File Edit View Insert Tools Desktop Window Help
Upside-Down Gaussian Cost Function
b
m
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3

W.3



- UG Cost function

W*. Increasing the Number of Iterations and Function Evals Per Cooling Step

- I changed maxiter and maxfunevals to 100
  - Using medium annealing (tfactor = 0.9)
    - Minimum error: 140.675976 (worse than when using maxiter and maxfunevals = 30, min error: 133.549618)
    - Total Elapsed Time = 0.922000 seconds (took much longer, as expected)
    - Best values found: m = 10.022, b = 0.0665
  - Using fast (tfactor = 0.5) and slow annealing (tfactor = 0.99) yielded the same results as before, except for longer evaluation times