LQR Optimization Landscape Visualizations

Dhruv Malik

February 19th 2018

One Dimensional Landscape

Consider the following matrices:

$A = \begin{bmatrix} 10 & 0 \\ 7 & 9 \end{bmatrix}$

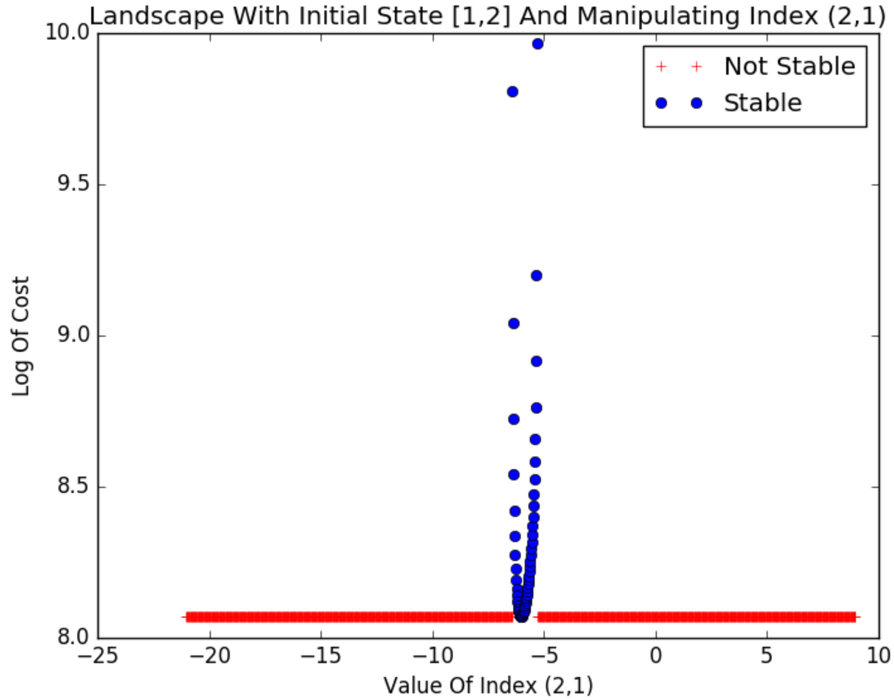
$B = \begin{bmatrix} -8 & 2 \\ 3 & 1 \end{bmatrix}$

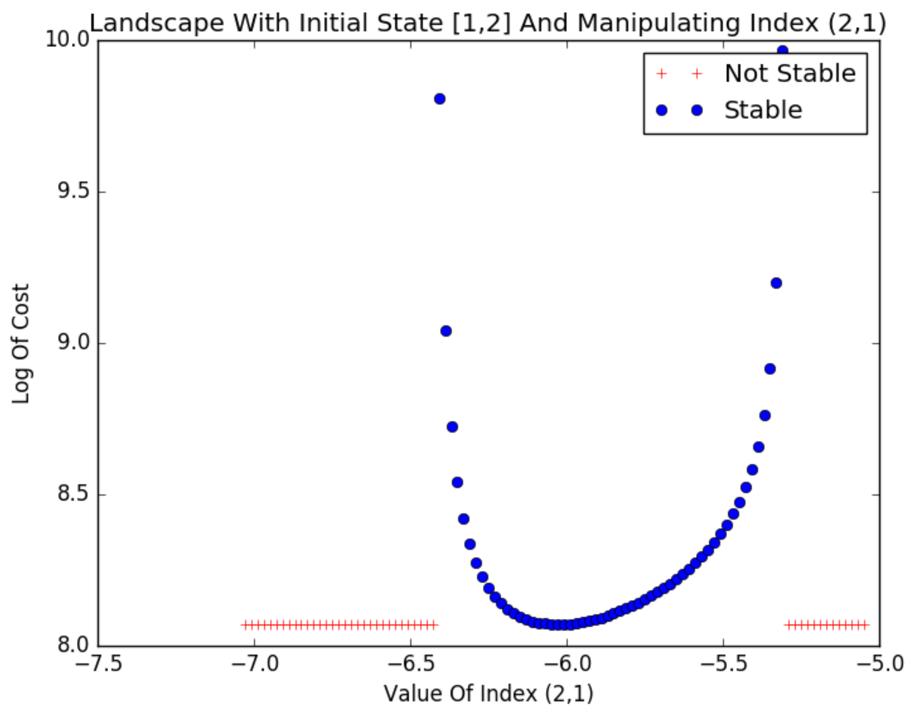
$Q = \begin{bmatrix} 58 & 2 \\ 2 & 58 \end{bmatrix}$

$R = \begin{bmatrix} 9 & 5 \\ 5 & 9 \end{bmatrix}$

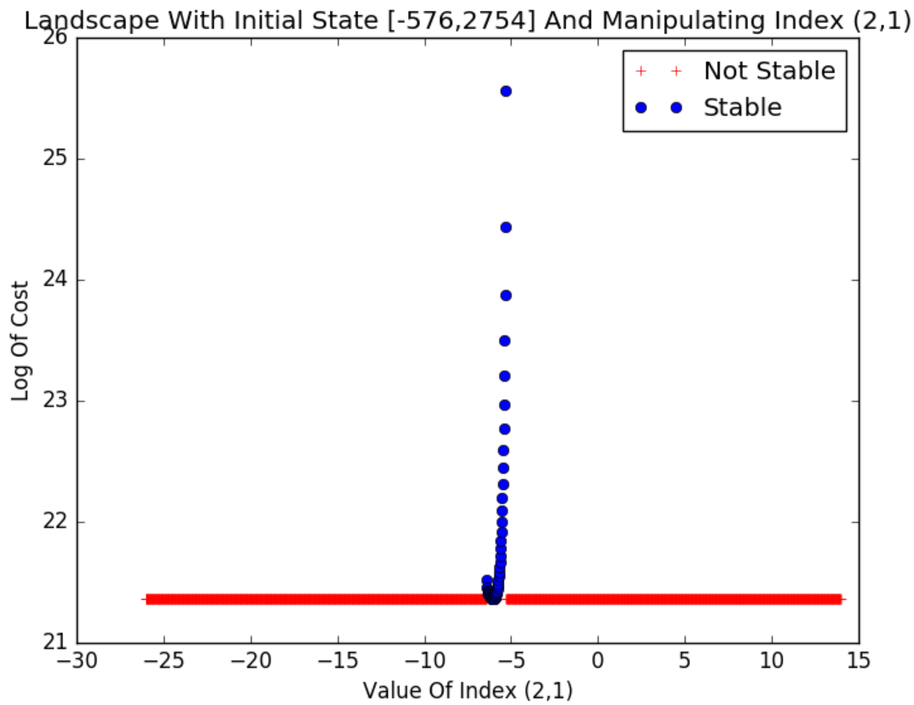
The optimal control matrix is: $K^* = \begin{bmatrix} -0.23267255 & -1.23395293 \\ -6.02894846 & -5.03248989 \end{bmatrix}$

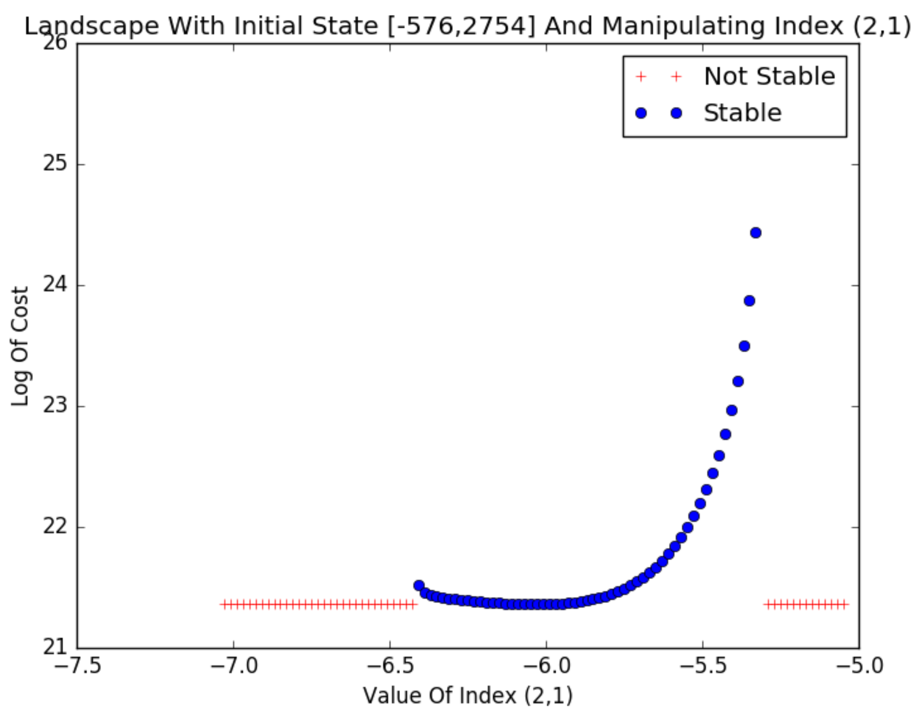
With initial state $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$, the optimal cost is 3198.79217912. In the plots below, we depict the cost of policies in two different neighborhoods of K^*. We manipulate a single index (the second row and first column). I manipulated other indices as well, but the plots were very similar and so I haven't shown them here. Note that the blue dots represent policies which are stable, while the red crosses represent policies which are not stable. Also note that the Y axis displays the logarithm of the cost. I had also tried an experiment where I looked in an interval of 1000 around the optimal index, but it was basically a red line (there were no new regions of stability found).





With initial state $\begin{bmatrix} -576 \\ 2754 \end{bmatrix}$, the optimal cost is 1894003233.26. I show the same plots below, with this initial state.





Observations

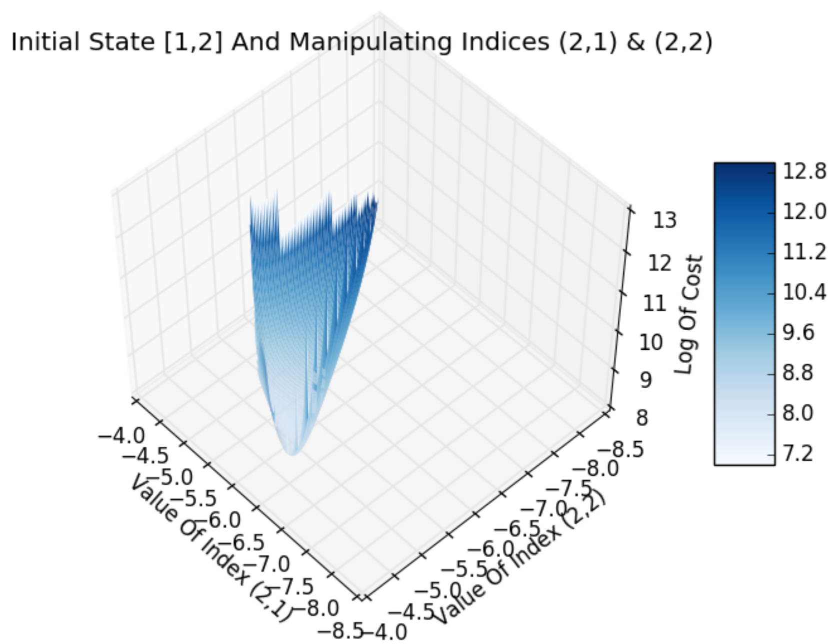
Here are the main takeaways from these one dimensional experiments:

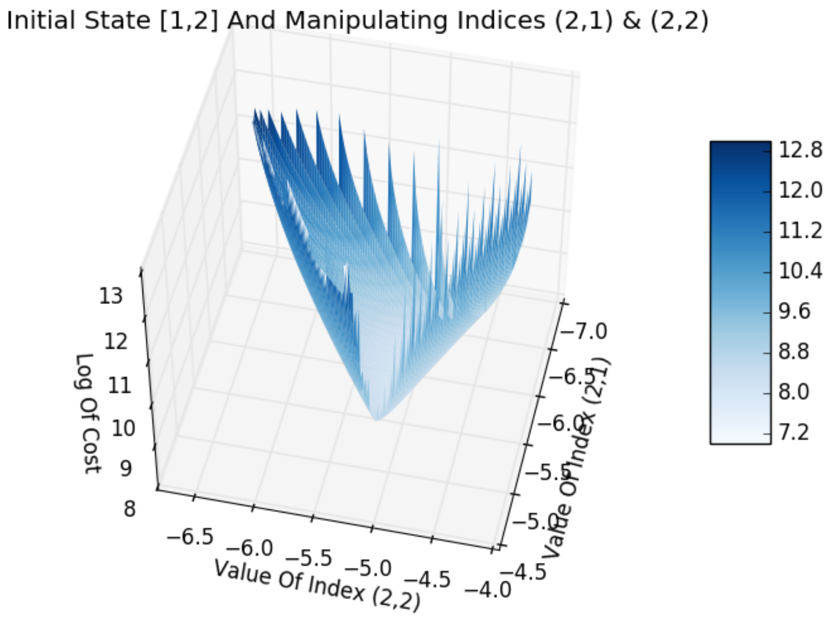
- There appears to be a single region of stability (with a single unique global minimum). I tried different matrices, and different intervals (including one of size 2000) around different indices and all the graphs look pretty much the same.
- The stable regions can be *extremely* steep. This is evidenced by the fact that a change of 0.75 in the value of an index results in exponentially large changes in cost.
- In the region of stability itself, the cost function looks convex. Indeed, these initial plots suggest that it is very strongly convex, because even with the log scale on the Y axis it appears as a quadratic.

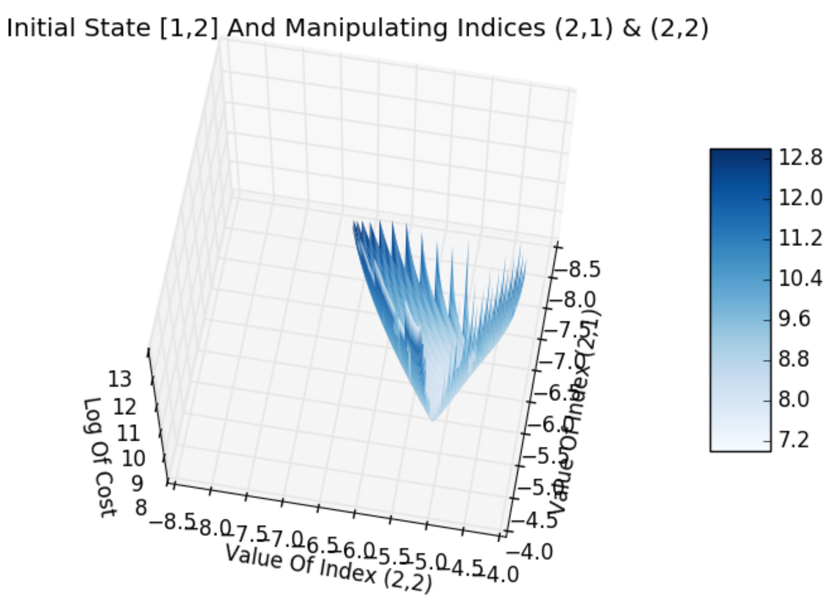
Two Dimensional Landscape

First Example

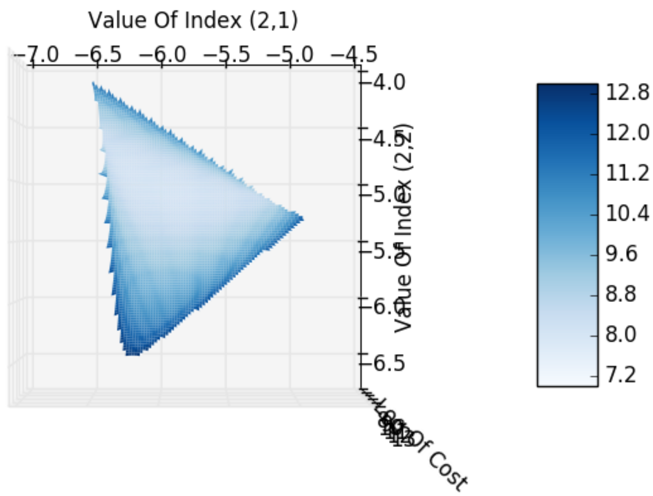
With the same matrices as in the previous section, we show a number of plots, where this time we vary indices (2,1) and (2,2). We give a number of different angles to view these plots below.



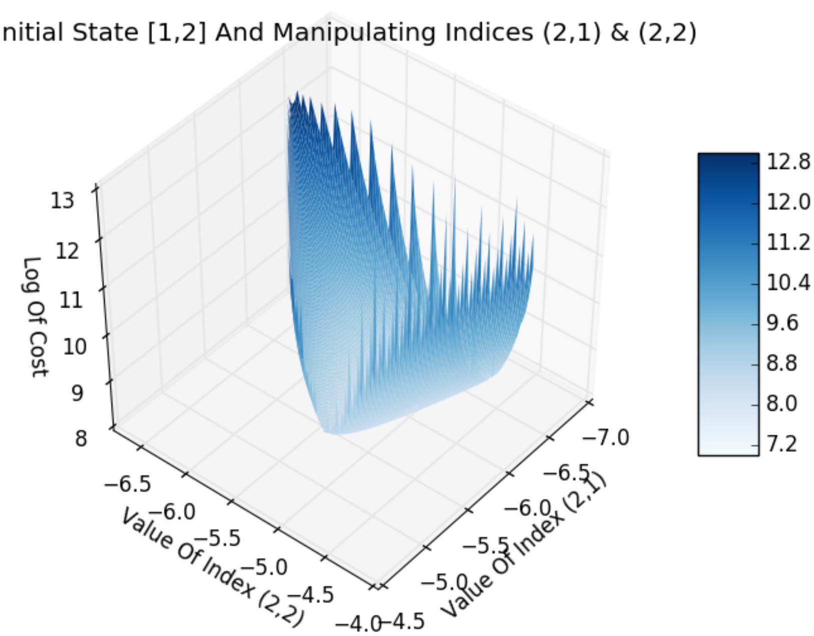

Initial State [1,2] And Manipulating Indices (2,1) & (2,2)
Log Of Cost
Value Of Index (2,2)
Value Of Index (2,1)
12.8
12.0
11.2
10.4
9.6
8.8
8.0
7.2

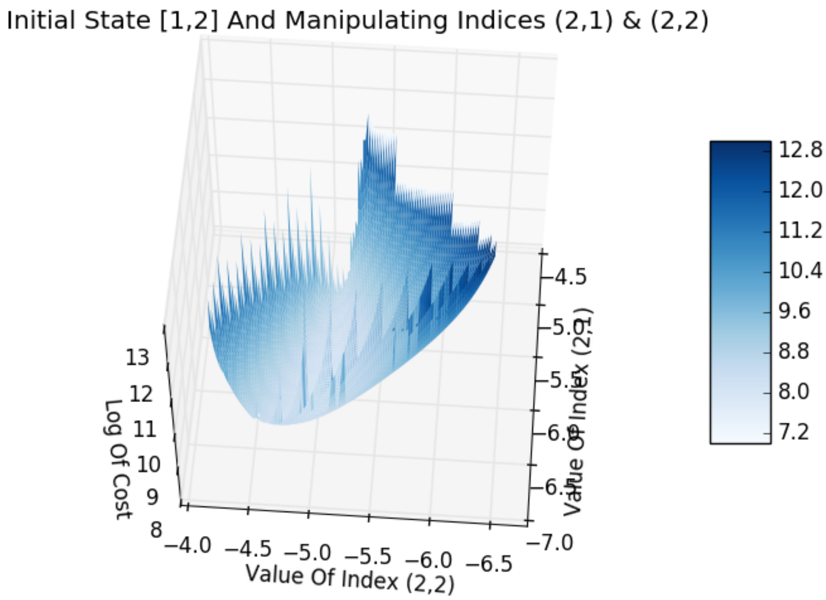

Initial State [1,2] And Manipulating Indices (2,1) & (2,2)
Log Of Cost
Value Of Index (2,2)
12.8
12.0
11.2
10.4
9.6
8.8
8.0
7.2

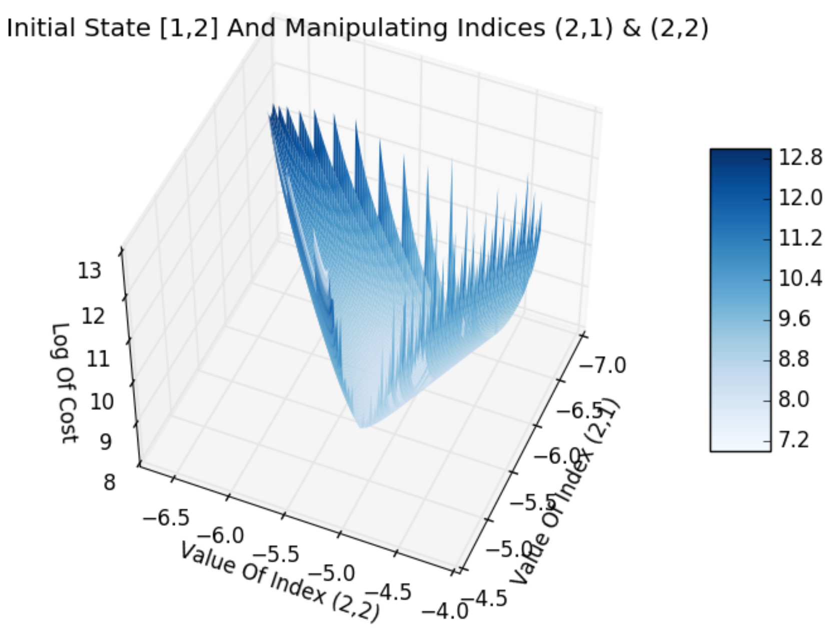
Initial State [1,2] And Manipulating Indices (2,1) & (2,2)

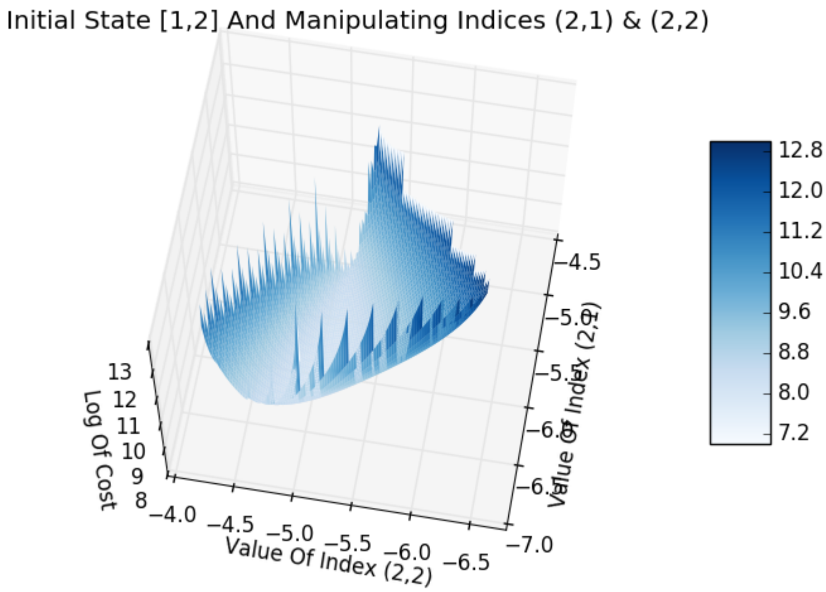


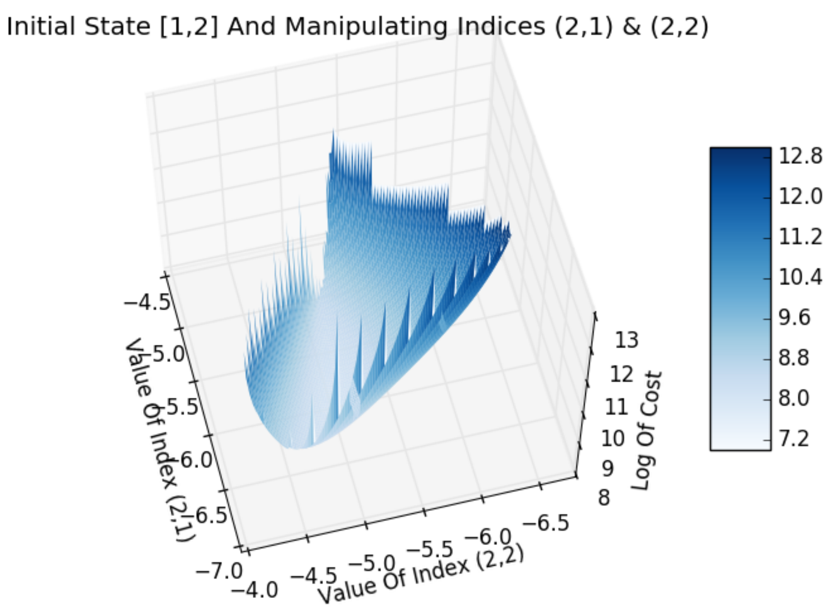
Initial State [1,2] And Manipulating Indices (2,1) & (2,2)




Initial State [1,2] And Manipulating Indices (2,1) & (2,2)
Log Of Cost
Value Of Index (2,2)
Value Of Index (2,1)
13
12
11
10
9
8
−4.0
−4.5
−5.0
−5.5
−6.0
−6.5
−7.0
12.8
12.0
11.2
10.4
9.6
8.8
8.0
7.2


Initial State [1,2] And Manipulating Indices (2,1) & (2,2)
Log Of Cost
Value Of Index (2,2)
Value Of Index (2,1)
13
12
11
10
9
8
−6.5
−6.0
−5.5
−5.0
−4.5
−4.0
−7.0
12.8
12.0
11.2
10.4
9.6
8.8
8.0
7.2

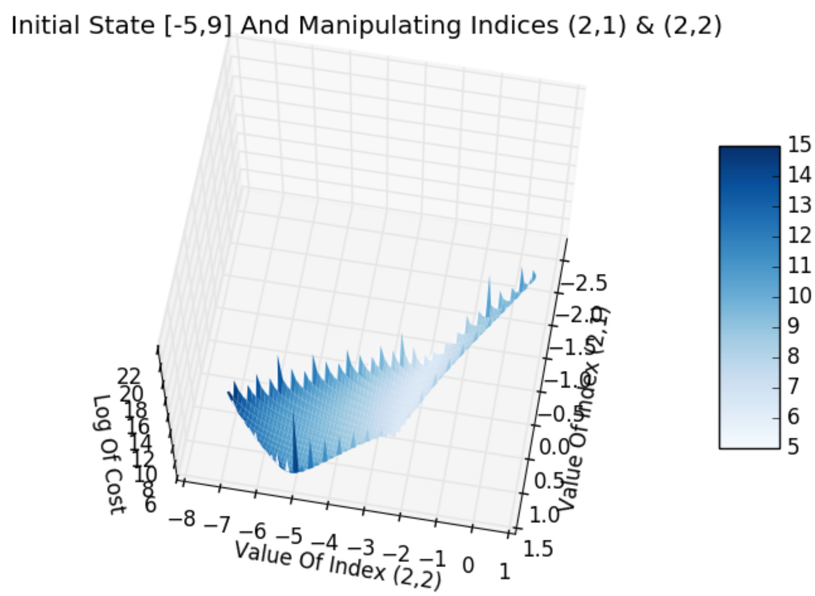




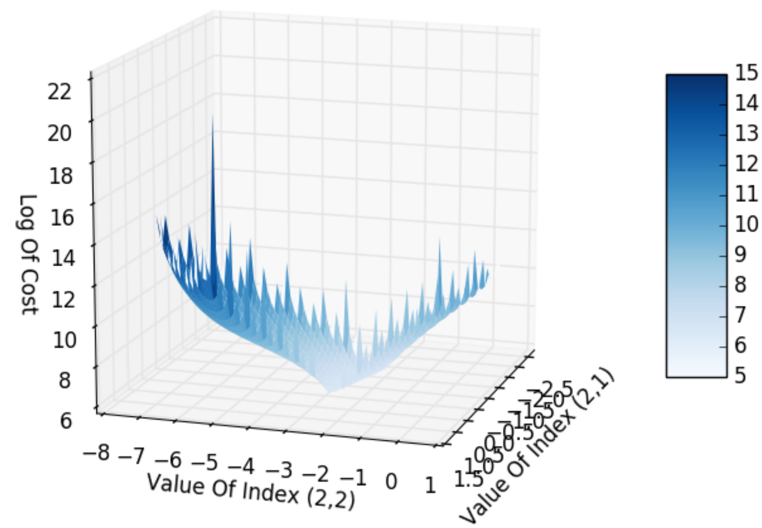
Second Example

Now consider the following matrices:

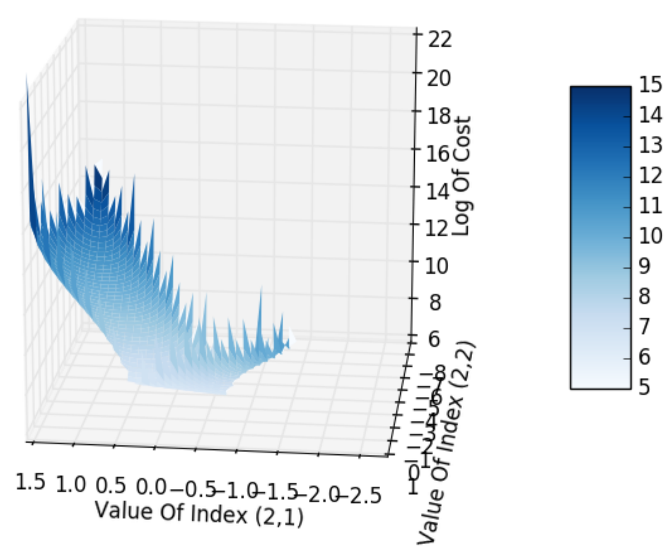
$A = \begin{bmatrix} -10 & 0 \\ 4 & 4 \end{bmatrix}$
$B = \begin{bmatrix} -8 & 2 \\ 3 & 1 \end{bmatrix}$
$Q = \begin{bmatrix} 100 & 0 \\ 0 & 40 \end{bmatrix}$
$R = \begin{bmatrix} 9 & 5 \\ 5 & 9 \end{bmatrix}$
The optimal control matrix is: $K^* = \begin{bmatrix} -1.27665239 & -0.55002008 \\ -0.10966854 & -2.199548 \end{bmatrix}$
With initial state $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$, the optimal cost is 614.01946945. Note that the titles in the plots below are wrong, they should say initial state $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$.



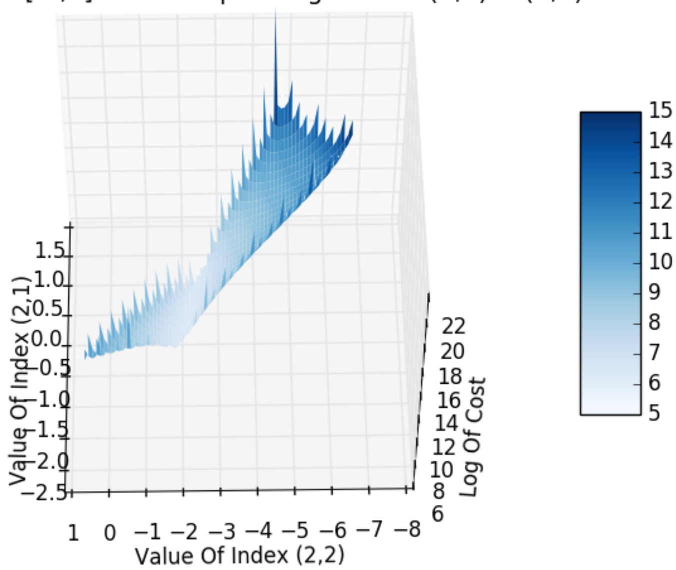
Initial State [-5,9] And Manipulating Indices (2,1) & (2,2)



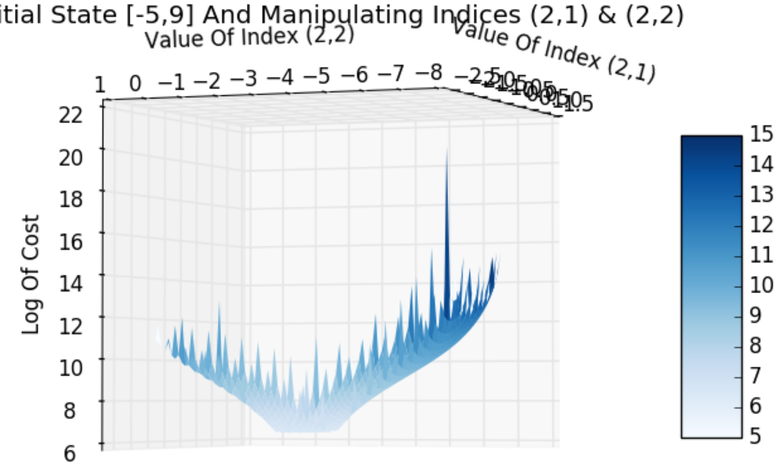
Initial State [-5,9] And Manipulating Indices (2,1) & (2,2)



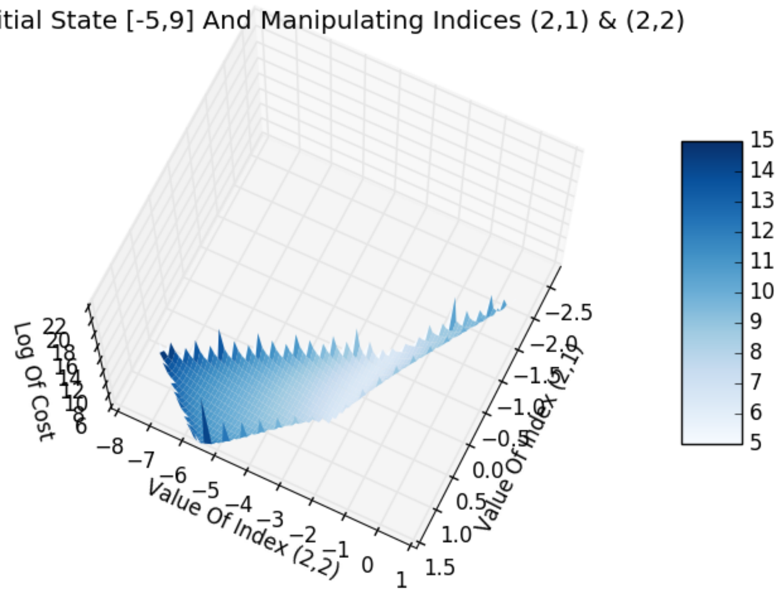
Initial State [-5,9] And Manipulating Indices (2,1) & (2,2)



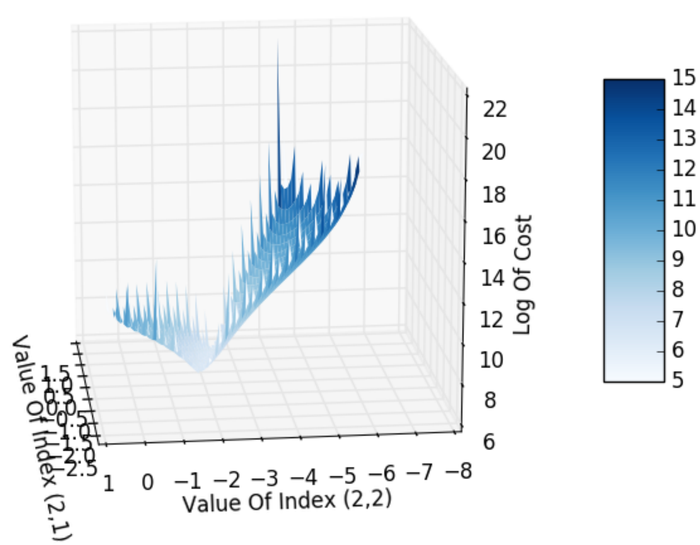
Initial State [-5,9] And Manipulating Indices (2,1) & (2,2)

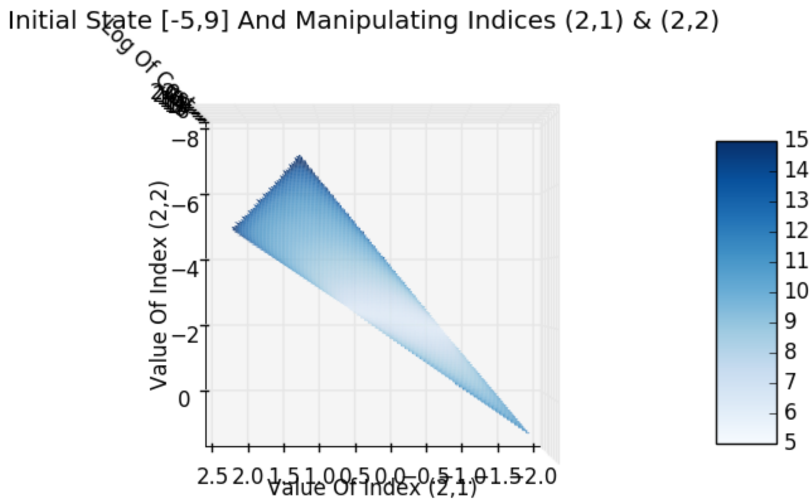


Initial State [-5,9] And Manipulating Indices (2,1) & (2,2)



Initial State [-5,9] And Manipulating Indices (2,1) & (2,2)





Observations

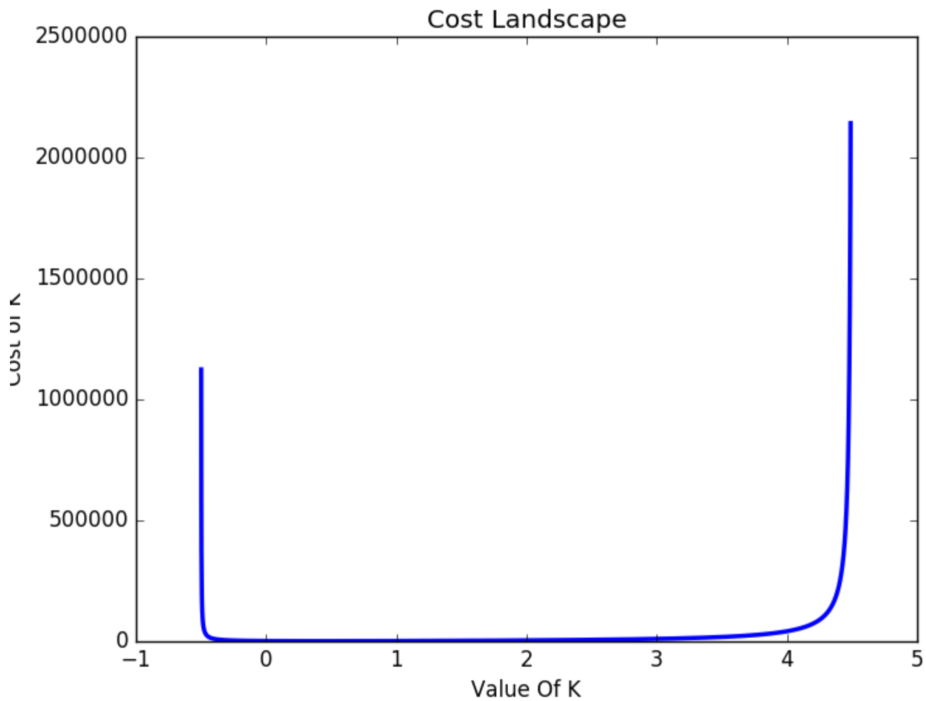
- These plots confirm the fact the region of stability can be extremely steep. The cost function also looks strongly convex again (like a quadratic).
- The 4th figure in the first example of this section, where we have a vertical view of the region of stability, is quite informative. This shows us that the set is non convex, but one can characterize most of the nonconvexity to come from the edges, which appear jagged. This rings a bell: the Rina Barber Foygel paper! Could be useful. The γ term would be quite small I think. Also ask for other resources relating to this (maybe finding the largest convex set in this set and then doing some optimization there)? However, note that this jaggedness along the edges could simply be due to the fact that I don't have enough resolution in my plot.
- Note that everything outside of the blue is outside the region of stability. I tried to do an experiment where I greatly increased the intervals around the indices and looked for other regions of stability, but my computer couldn't handle it.

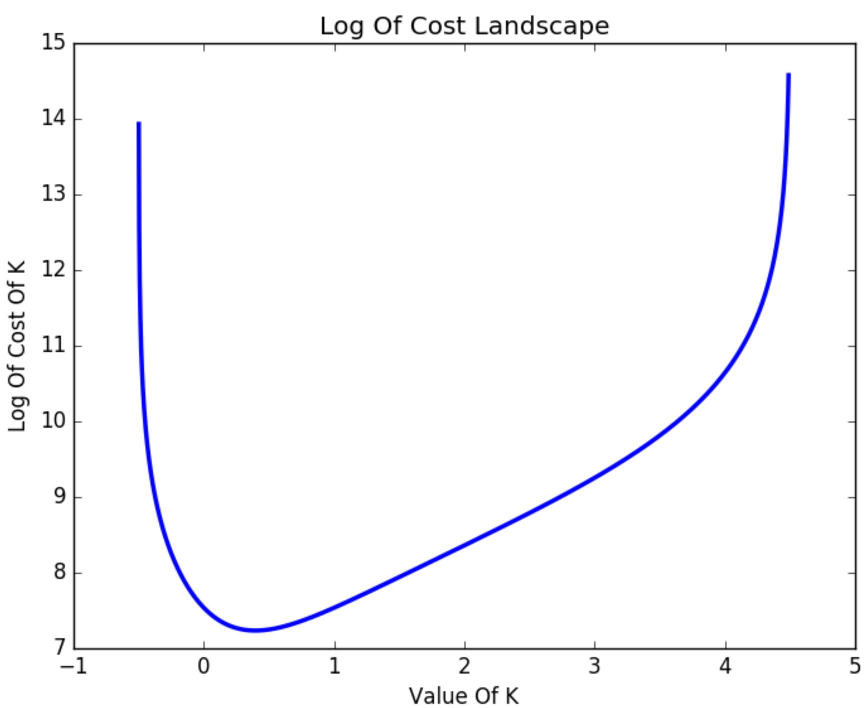
Progress Of Gradient Descent

Consider the following one dimensional LQR system:
$A = 0.8$
$B = 0.4$
$Q = 3$
$R = 4$

Initial State = 15

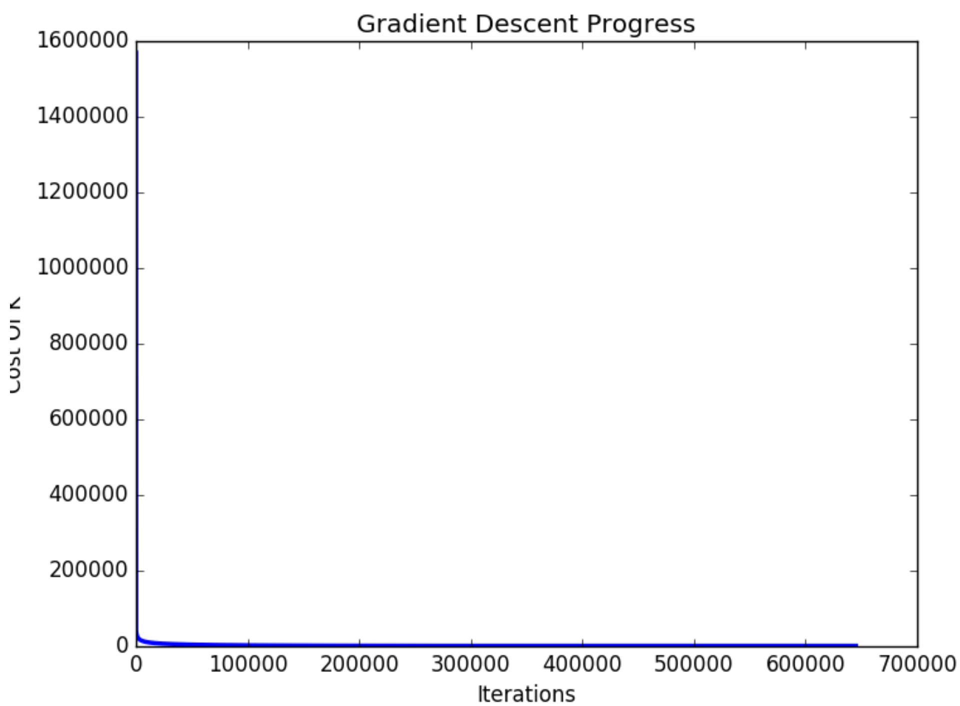
Let's examine the following plots to get an idea of the cost landscape:

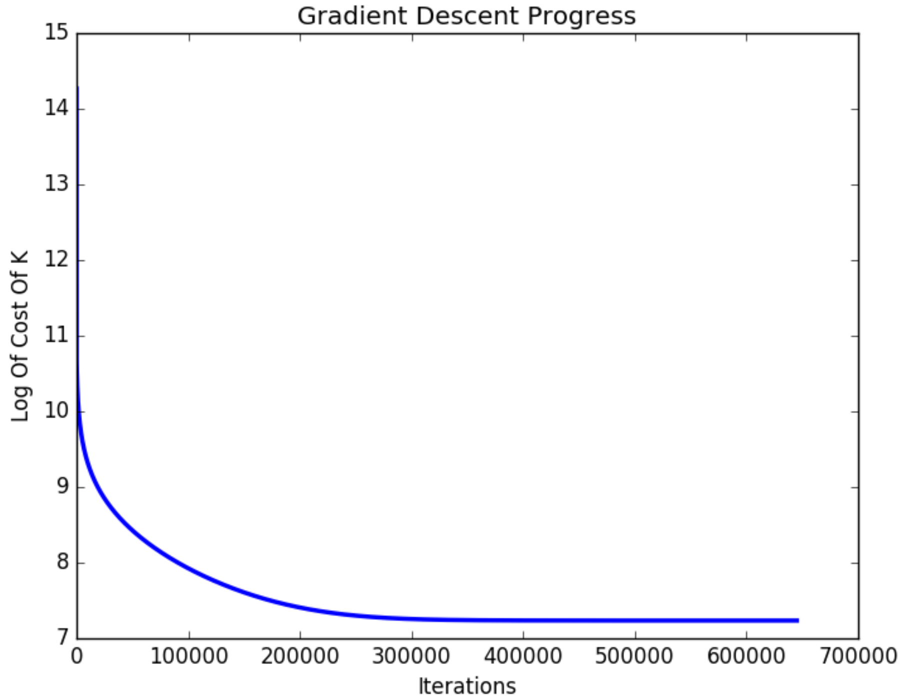


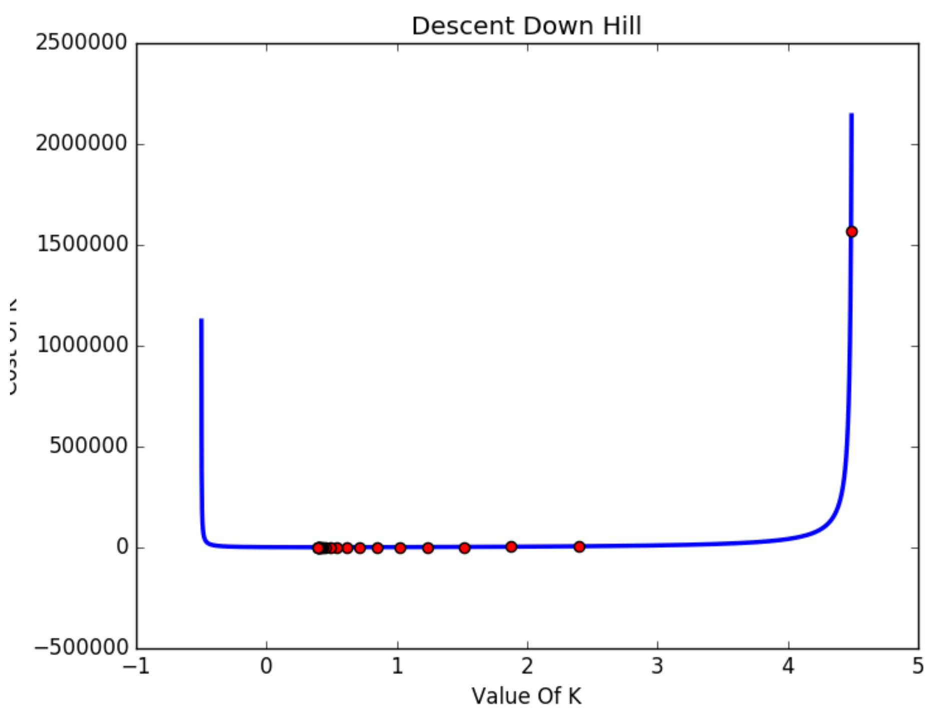


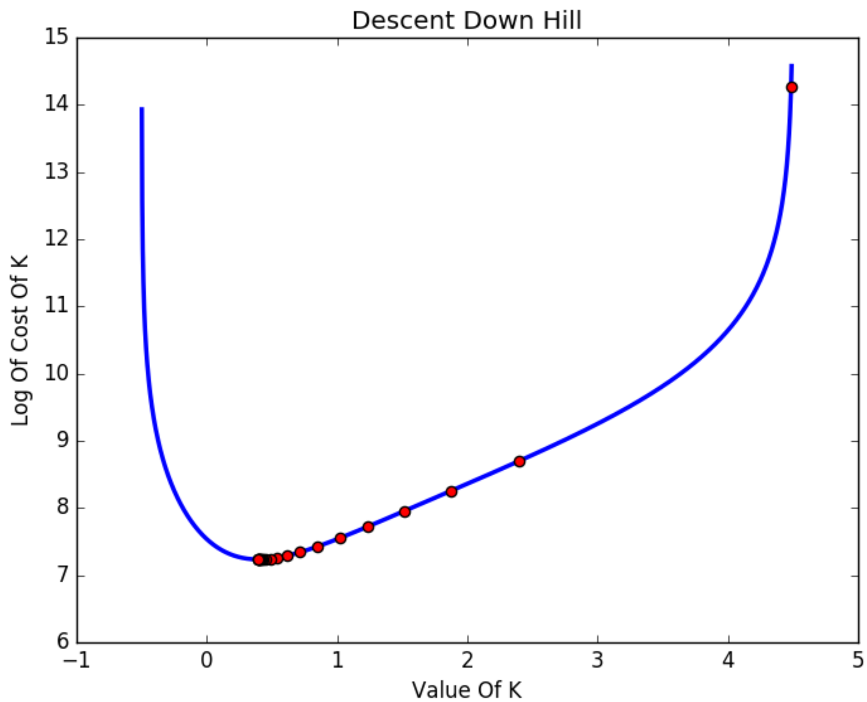
The main takeaway is that the function is extremely steep, and so we don't the ability to bound gradients. According to local smoothness, this can't be improved even near the optimum. The reason is that this property is an equality, which means that we are fundamentally constrained no matter whether we are close or far from the optimum.

If we run gradient descent on this function (starting from K = 4.485) then the plots below show us the progress made by this algorithm:




Gradient Descent Progress
Log Of Cost Of K
15
14
13
12
11
10
9
8
7
0
100000
200000
300000
400000
500000
600000
700000
Iterations


Descent Down Hill
2500000
2000000
1500000
1000000
500000
0
−500000
−1
0
1
2
3
4
5
Value Of K


Descent Down Hill
Log Of Cost Of K
Value Of K
15
14
13
12
11
10
9
8
7
6
−1
0
1
2
3
4
5