# Objective

The objective of this assignment is to write the code for the simulated annealing algorithm that allows for a one-pass penalty and quadratic penalty. Then, using this simulated annealing code find an appropriate c and epsilon value given the bounds using the bump function. Then, use this value for running the SA algorithm on a 10-bar truss for optimizing the weight of the trusses.

# Code Structure

Please see the docstrings on the Python files.

# Bump Function Results

The primary procedure for getting a good value for the cooling schedule parameter, c, is by trial and error after fixing a value for the rest of the parameters. As suggested by the assignment, the following parameter values are used that were fixed, so that only one parameter will be changed at a time:

t\_start (starting temperature) = 1000

max\_iter (maximum number of iterations) = 5000

epsilon = 2

The epsilon value was chosen as per the suggestion on the lecture slide which says it is typically between 0.1-0.3 – and I simply chose the middle value which is 0.2. However, since I did not normalize move.py, epsilon had to be multiplied by the range of the bounds i.e. upper bound – lower bound. The upper bound is 10, therefore, 0.2\*10 = 2.

The initial design variable value, x0 is [1, 1] (since n=2, per the assignment). Also, per the assignment, lower bound is [0, 0] and upper bound is [10, 10] and are all inputted into move.py as floats.

Discuss the procedure you adopted to obtain a good value of the exponential cooling schedule parameter c (include averaged convergence trends for typical values of c to justify your conclusions)

Report the best solution (x & f)

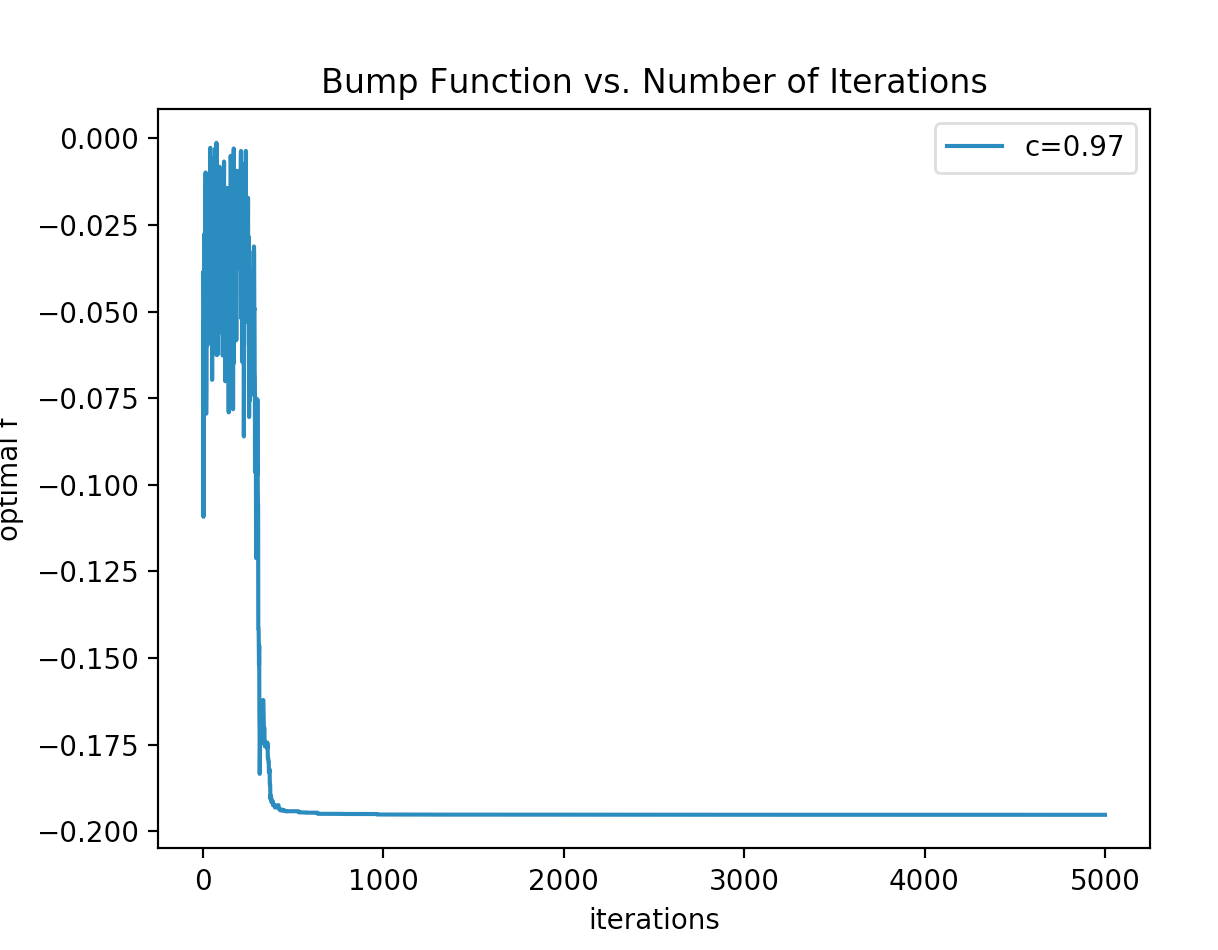
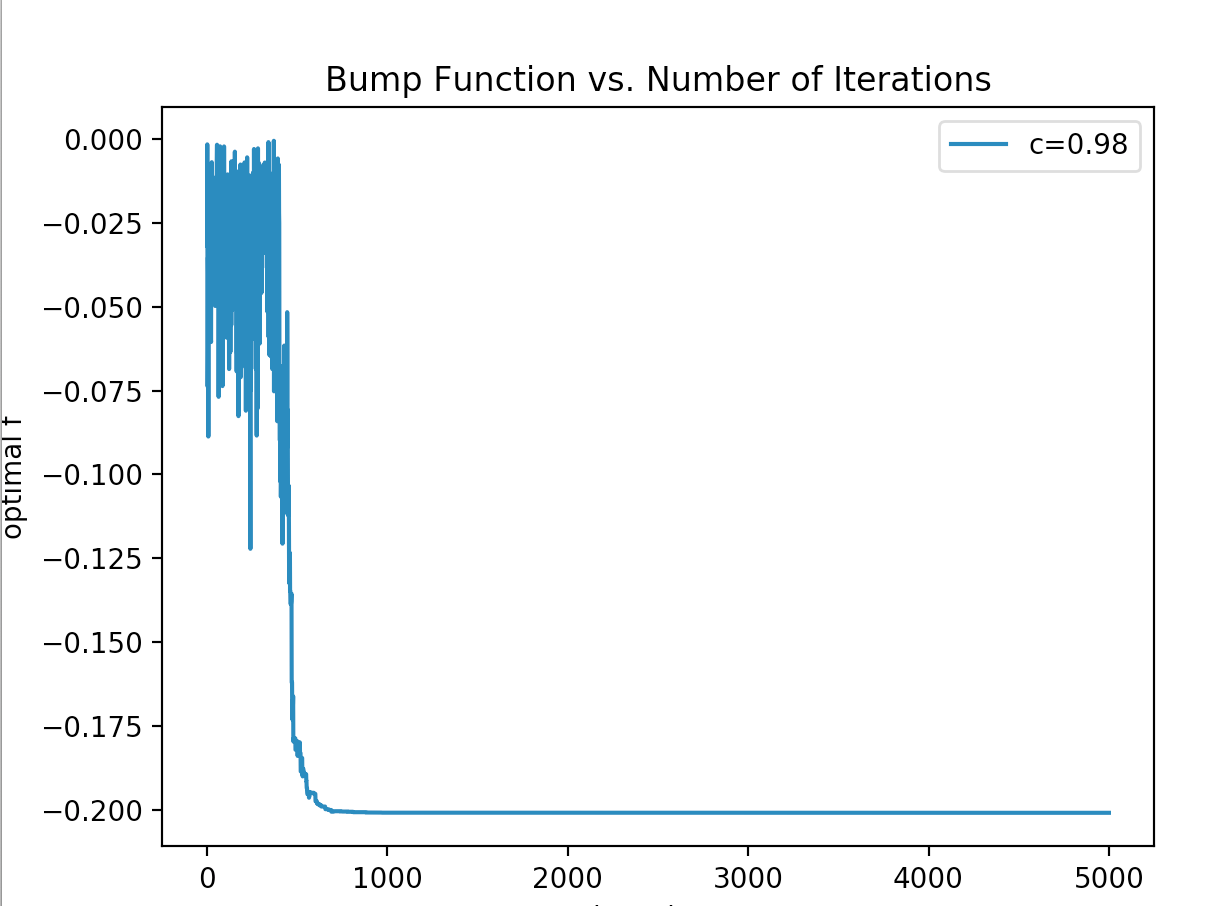
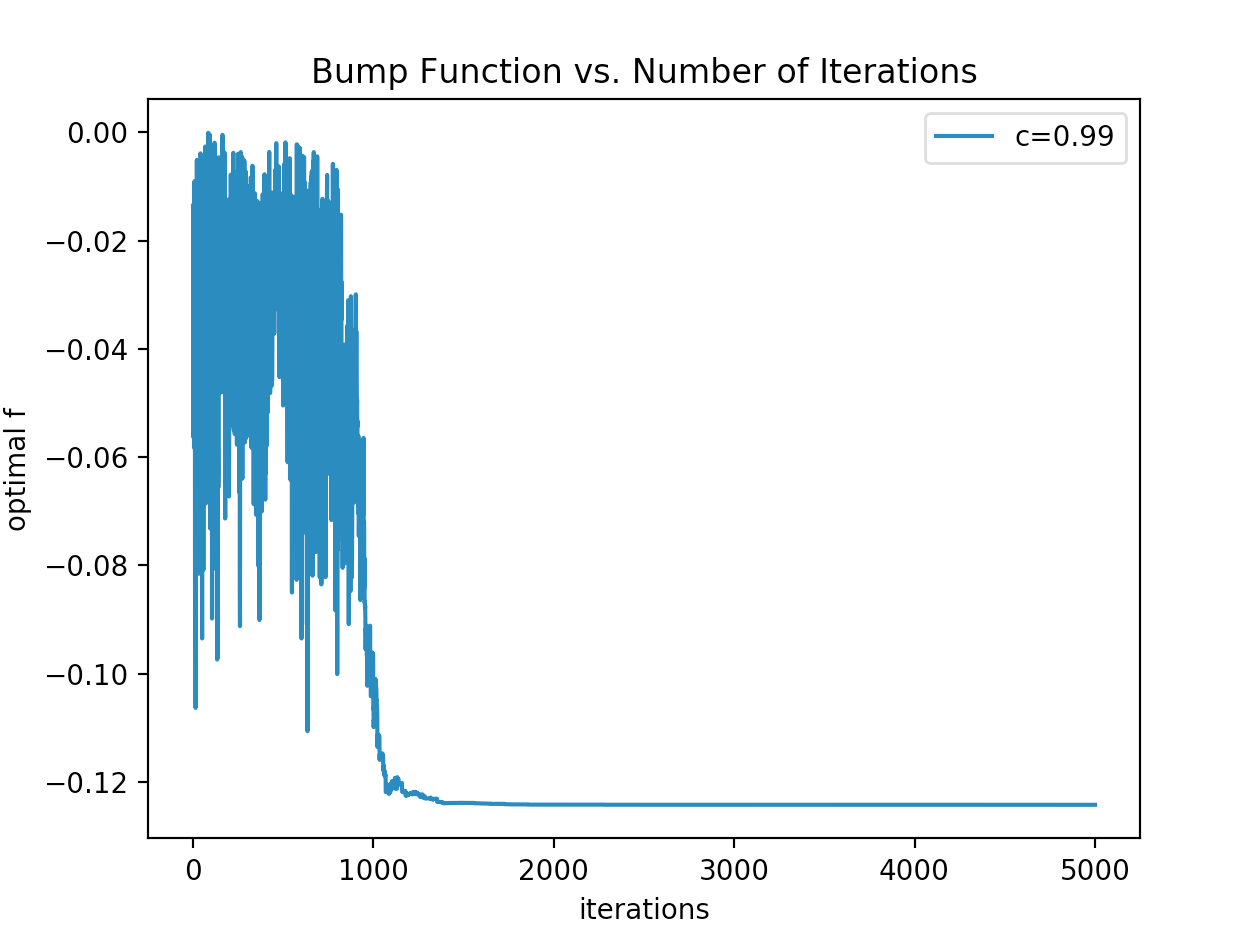
The best solution for x and f, outputted as xopt and fopt in my Assignment3.py file over multiple times of running the same file and averaging the values are the following:

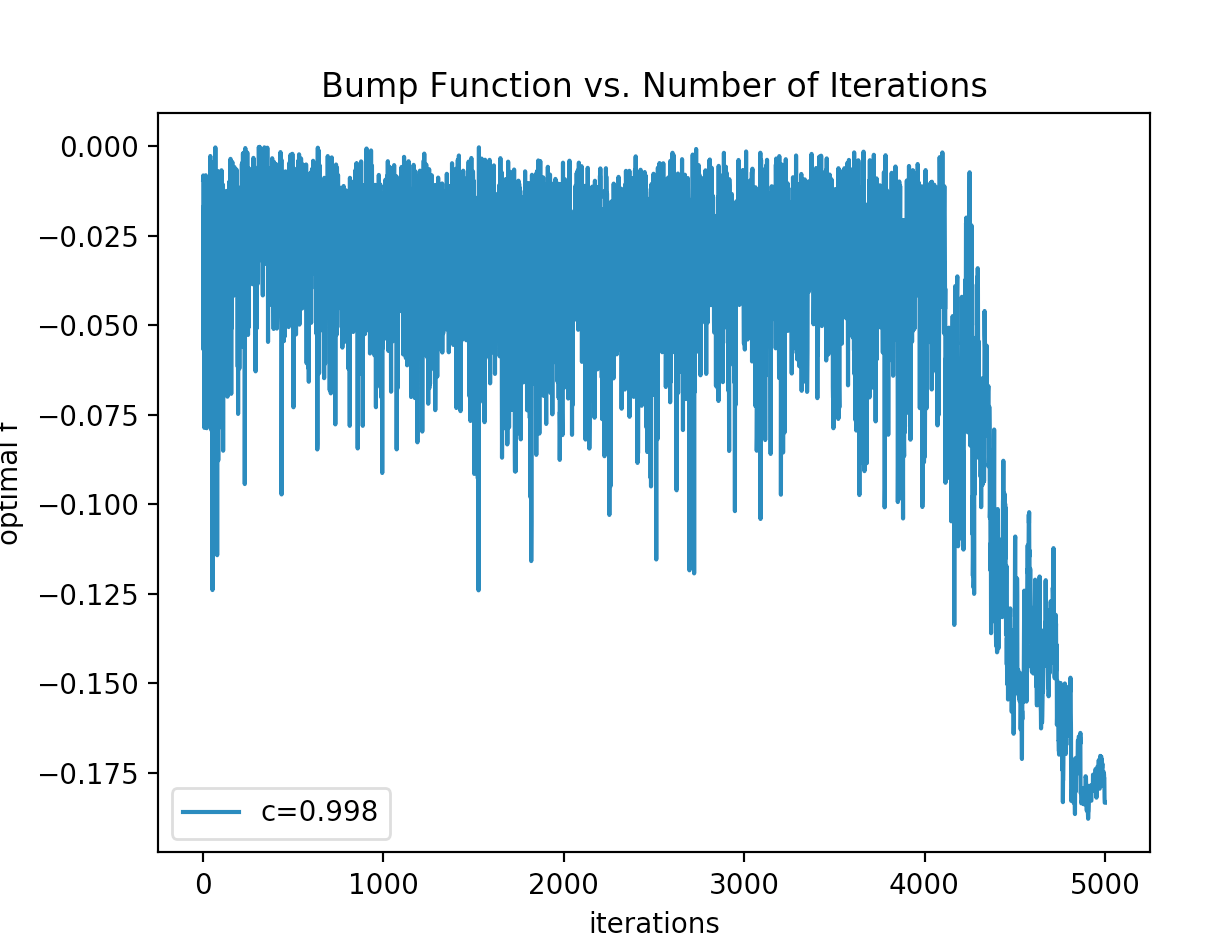
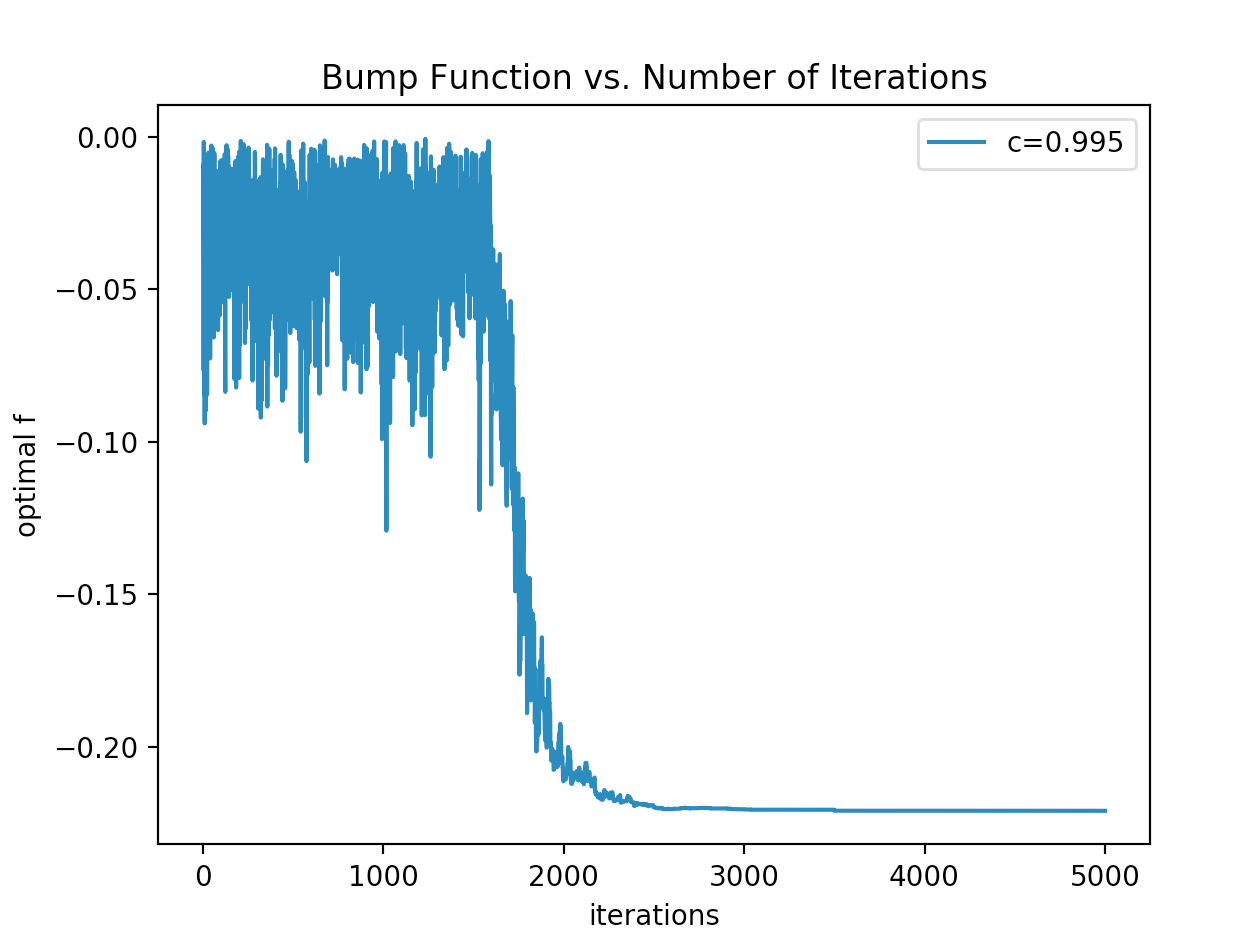
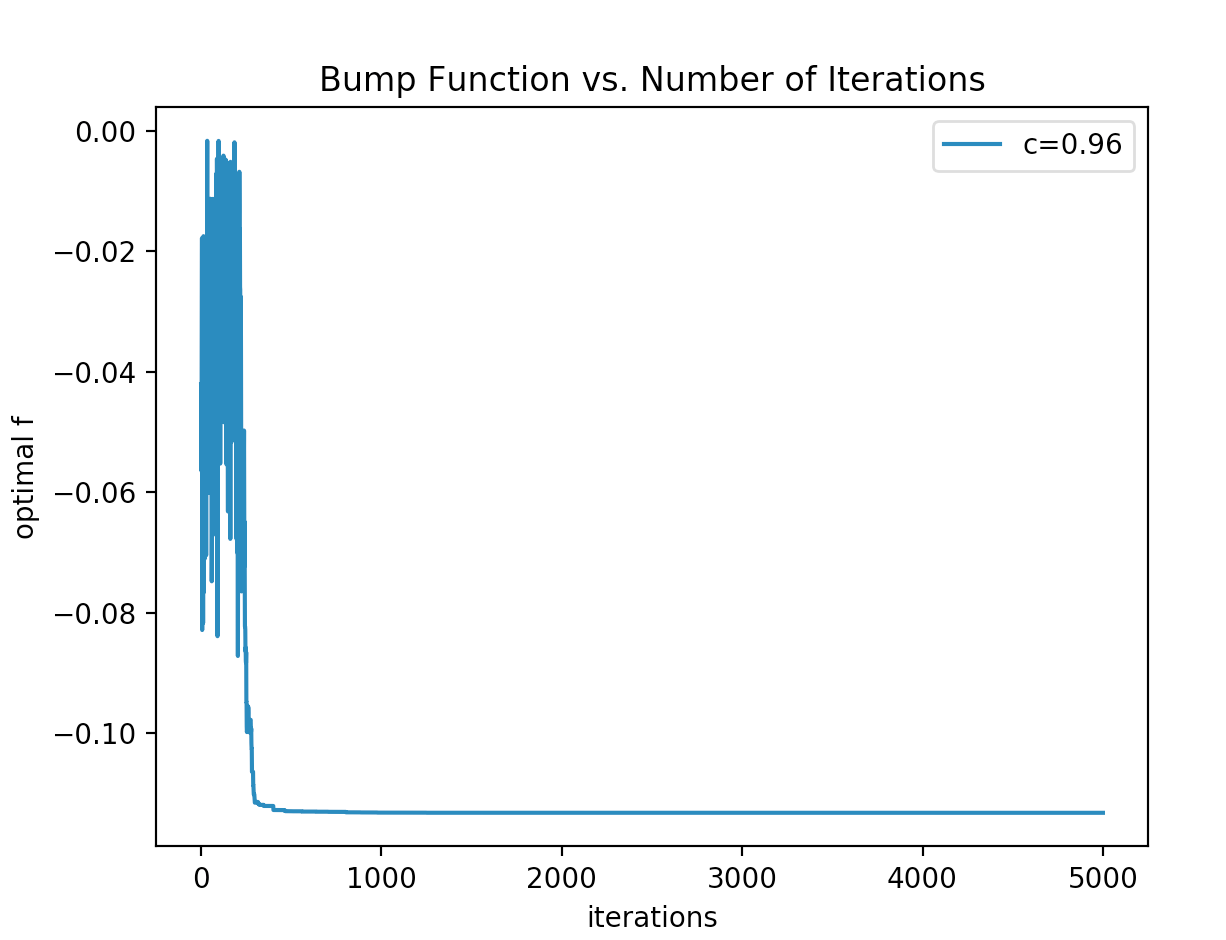
xopt [ 1.59415299 0.47298035]

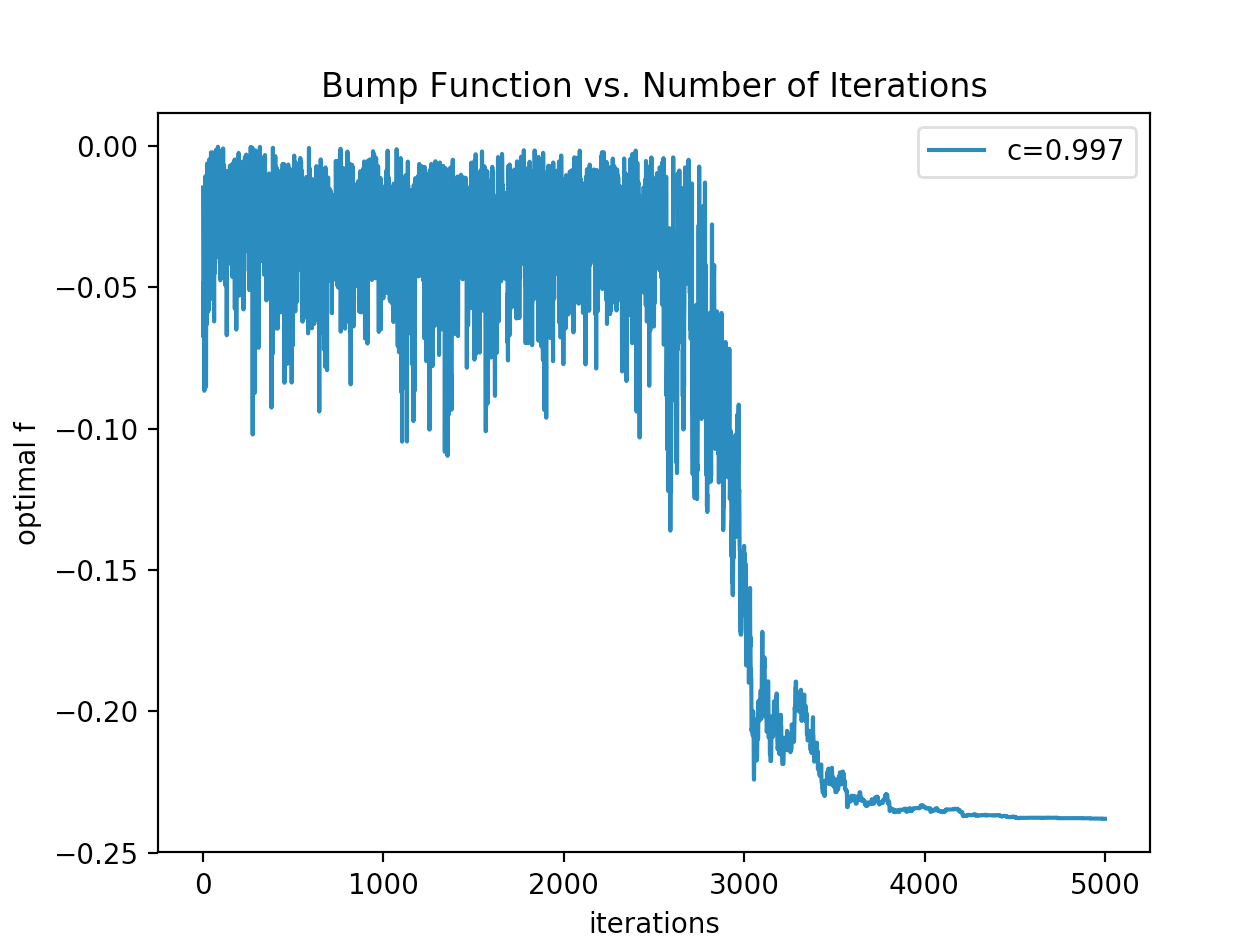
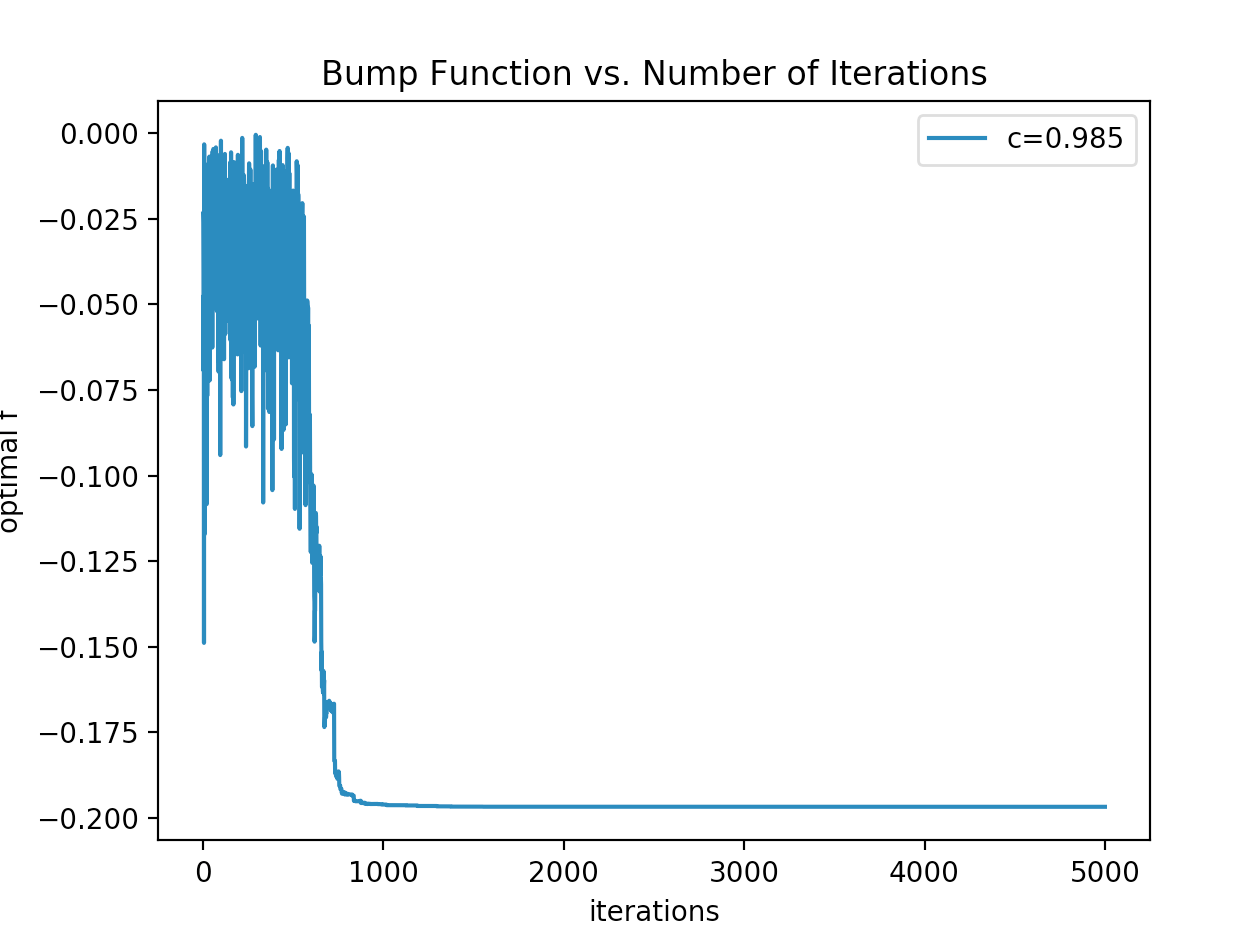
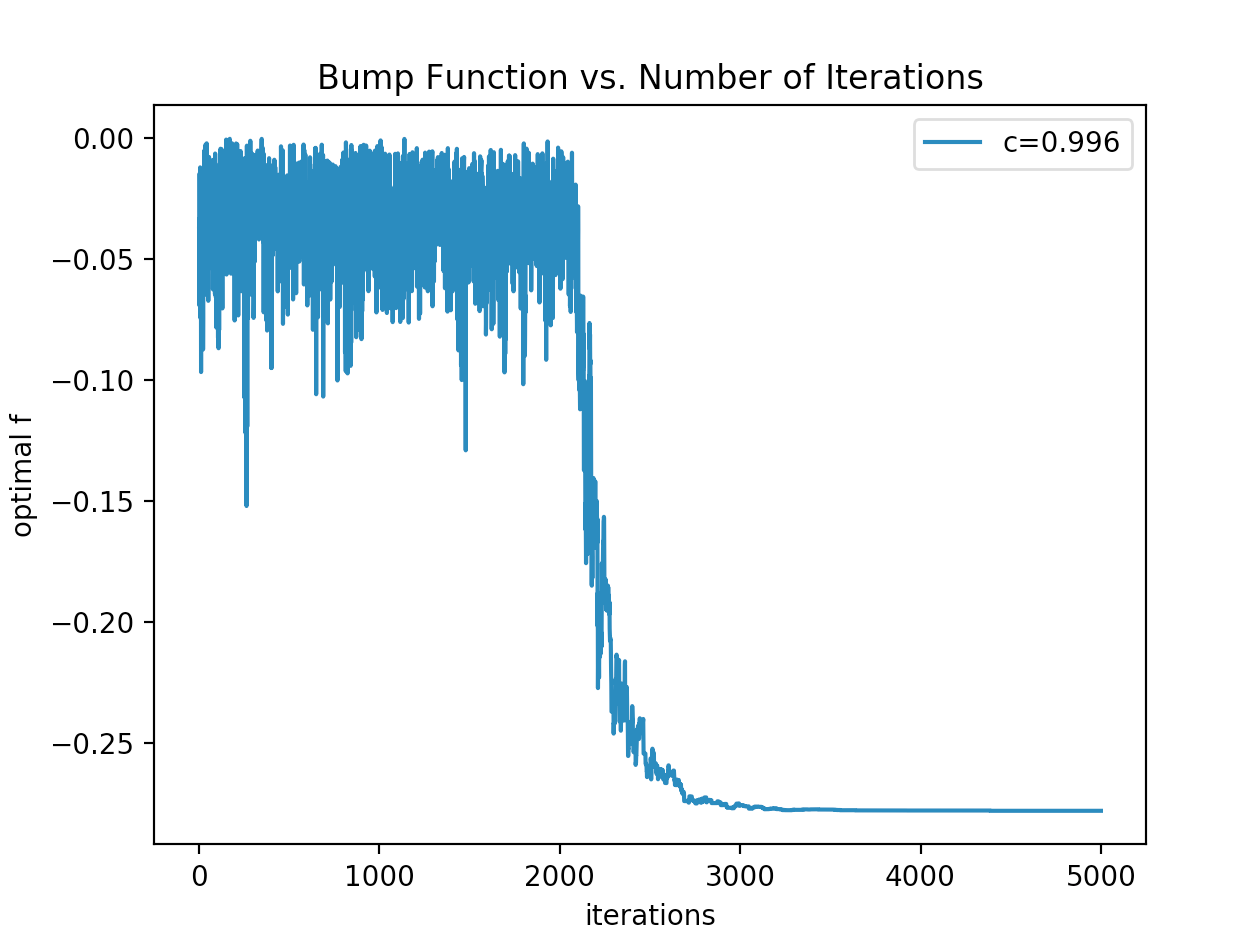
fopt -0.362774184804

The best value for the parameter c is 0.996

A more rigourous approach is to create take a lot of values close to 1 since that’s the trend I noticed by trial and error and run each of these c values ran 5 times and these runs are averaged. The graphs for the convergence are shown below, as can be seen at aroung c=0.998 it is not converging with 5000 iterations and 0.996 has the lowest average of fopt values. I also added a random seed to reduce the randomness, though this did not have a noticeable effect.







# 10-bar truss optimality

{ Optimal design of the 10 Bar truss structure [4pts]

Using one of the runs for **a one-pass penalty** the following output is generated

fopt 0.0120198275488

xopt [ 7.03251211e-07 1.68089179e-07 1.25370675e-07 2.94704677e-07

5.05055563e-07 3.34481807e-07 6.56773300e-07 1.31546161e-07

4.16116607e-07 3.77203532e-07]

Using one of the runs with **alpha=1.5** using a quadratic penalty the following output is generated:

fopt 0.010418926282

xopt [ 4.55822415e-07 2.09544049e-07 2.92426427e-07 2.22938374e-07

4.68722825e-07 4.88261034e-08 1.00062975e-06 5.92857355e-08

2.52154918e-07 2.18754901e-07]

fopt is optimized total weight of the truss, calculated assuming the density and yield strength of a 6061 aluminium alloy. This has a density of 2700kg/m^3 and a yield strength of 270 \* 10^6 N/m^2.

xopt is the vector of cross-sectional area of each bar with the indices corresponding to the bar elements shown in Assignment 1.

## One-Pass Penalty vs Quadratic Penalty

The results from the one-pass and quadratic penalty are similar both around 10g.

### Convergence Trend of one-pass penalty

### Compare 3 penalty parameters and convergence using quadratic penalty

I chose the penalty parameters based on the lecture notes showing that alpha is the value that can be changed and is within [1,2]. So, I took the two extreme cases, and the middle case. The figure below shows the convergence trends with varying alpha values.

Compare averaged convergence trends for the one-pass penalty and

quadratic penalty function approach (3 typical values of the penalty

parameter)

**Choose at least 3 penalty parameter by changing alpha and fixing everything else.**

Discuss how you went about selecting appropriate values of the penalty

parameter for the study

# Comparison

Report the best solution obtained using both approaches

Are your optimal designs feasible?

Because there is no buckling it is an incredibly thin and light weight truss structure. It is not feasible because we have to account for buckling in truss designs.