Problem Set #4

MACSS 40000
Luxi Han, 10449918

Problem 1

1.

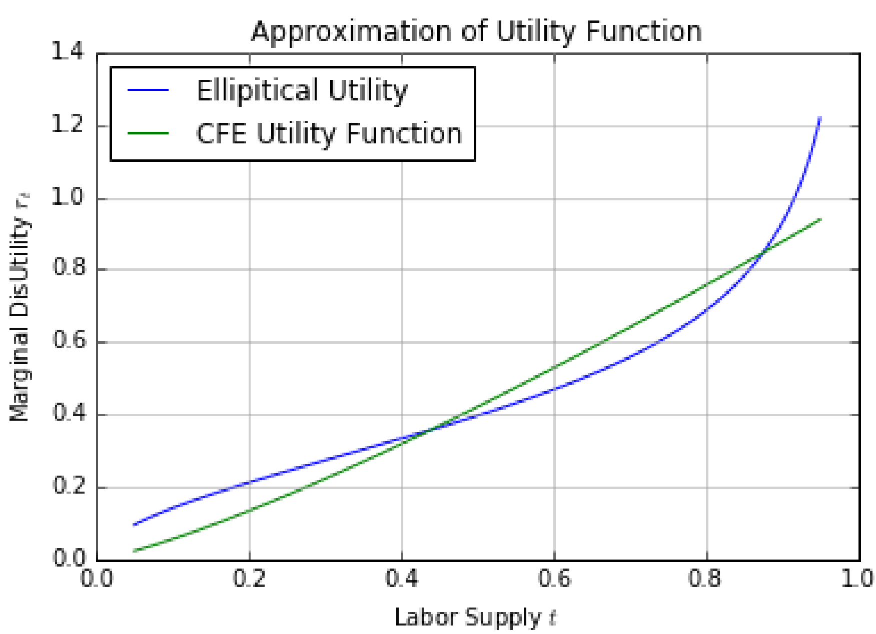
Now take first derivative with respect to l. Then we can get the following expression:

$$v_{cfe}(l) = (1-l)^{\frac{1}{\theta}}$$
$$v_{elp}(l) = b(1-l)^{\mu-1}[1-(1-l)^{\mu}]^{\frac{1-\mu}{\mu}}$$

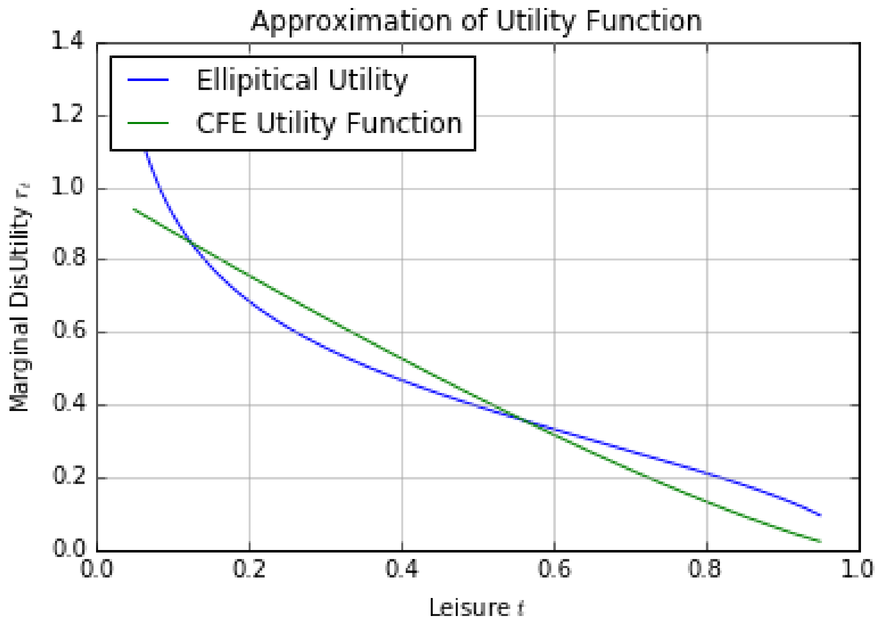
Notice that they are both bigger than 0.

2.

For this problem, we plot both the marginal utility of leisure as well as the absolute value of disutility of labor.
Absolute value of marginal disutility of labor supply is shown as following



Marginal Utility for leisure is shown as below:



Problem 2

1.

If I choose `nvec_guess = 0.95 * np.ones(S)`, then I will get the follwing result:

```
f_1 = \
    (array([False, False, False, False, False, False, False,
False, False, False], dtype=bool),
     array([False, False, False, False, False, False, False,
 False, False, False], dtype=bool),
     array([ True, False, False, False, False, False, False,
 False, False], dtype=bool),
     array([ True, False, False, False, False, False, False,
False, False, False], dtype=bool),
     False)\\
```

We can see that the first consumption in the first period and the second period saving are too high.
This means that the agent saves too much in the first period so that his/her consumption goes to negative.

2.

If I choose `nvec_guess = 0.95 * np.ones(S)` and `bvec_guess = np.append([0.5], np.ones(S - 1))`, then I will get the follwing result:

```
f_2 = \
    (array([False, False, False, False, False, False, False,
 False, False, False], dtype=bool),
     array([False, False, False, False, False, False, False,
 False, False, False], dtype=bool),
     array([ True,  True, False, False, False, False, False,
 False, False], dtype=bool),
     array([False,  True, False, False, False, False, False,
 False, False, False], dtype=bool),
     False)
```

As we can see, this time, second and third period saving and second period consumption is violated.
This is because, the agent saves too little in the first period and yet save too much for the second period.

3.

If I choose `nvec_guess = 0.95 * np.ones(S)` and
`bvec_guess = np.append([0.5,], np.ones(S - 1))`, then I will get the follwing result:

```
f_3 = \
    (array([False, False, False, False, False, False,
False, False, False, False], dtype=bool),
     array([False, False, False, False, False, False, False,
 False, False, False], dtype=bool),
     array([False, False, False, False, False, False,
False, False, False], dtype=bool),
     array([False, False, False, False, False, False,
False, False, False, False], dtype=bool),
     False)
```

Now, no constraints are violated.

4.

If I choose the following guess:

```
nvec_guess = 0.5 * np.ones(S)
bvec_guess = \
       np.array([-0.01, 0.1, 0.2, 0.23, 0.25, 0.23, 0.2, 0.1,
       -0.01])
```

We get the following result:

```
f_4 = \
    (array([False, False, False, False, False, False,
False, False, False, False], dtype=bool),
     array([False, False, False, False, False, False,
 False, False, False, False], dtype=bool),
     array([False, False, False, False, False, False,
False, False, False], dtype=bool),
     array([False, False, False, False, False, False,
 False, False, False, False], dtype=bool),
     False)
```

No contraint is violated.

Problem 3

1.

The following is my result:

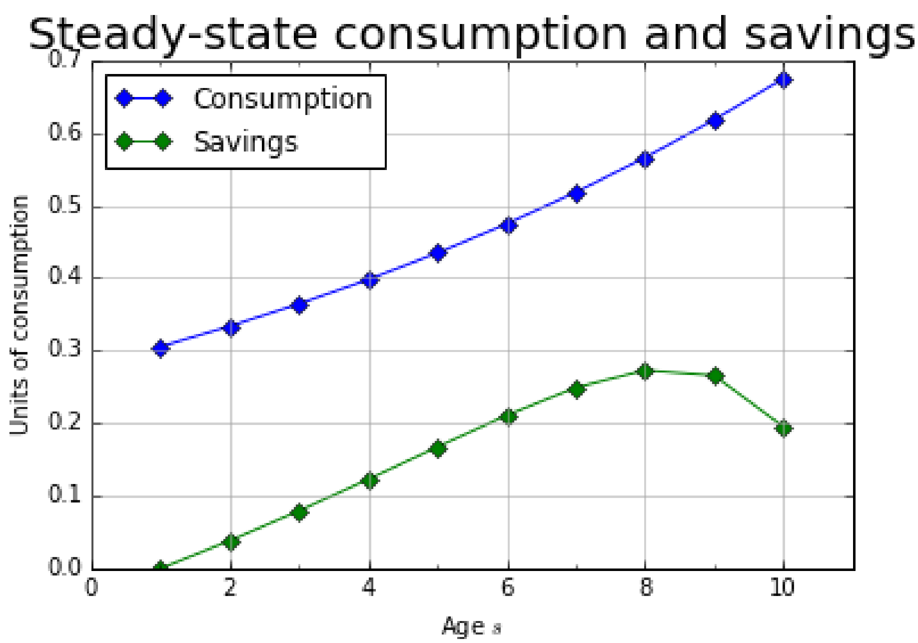
```
ss_output = \
    {'C_ss': 4.6866646273404537,
     'EulErr_ss': array([  0.00000000e+00,  -2.13162821e-14,   2.48689958e-14,
             -2.84217094e-14,   1.77635684e-14,  -1.42108547e-14,
              1.42108547e-14,  -1.15463195e-14,   1.77635684e-15,
             -1.57385216e-12,   3.62376795e-13,  -6.07514039e-13,
              1.29674049e-13,   4.44089210e-14,   6.26165786e-14,
              2.08721929e-14,   1.20348176e-13,   7.99360578e-15,
             -1.06581410e-14]),
     'K_ss': 1.6018986928759162,
     'L_ss': 9.8779380658172506,
     'RCerr_ss': -3.3306690738754696e-16,
     'Y_ss': 5.2258309985072442,
     'b_ss': array([ 0.03834027,  0.07940578,  0.12276813,
 0.16737695,  0.2109947 ,0.24912868,  0.27300846,
 0.26575224,  0.19512349]),
```

```
    'c_ss': array([ 0.30550668,  0.33362127,  0.36432313,
  0.39785036,  0.43446297, 0.4744449 ,  0.51810621,
  0.5657855 ,  0.61785252,  0.67471108]),
    'n_ss': array([ 0.99991426,  0.99982011,  0.99962266,
   0.99920868,  0.9983415 ,0.99652821,
0.99275076,  0.98494243,  0.96905683,  0.93775263]),
    'r_ss': 0.80521601275234878,
    'ss_time': 0.03868644034764657,
    'w_ss': 0.3438764372075131}
```

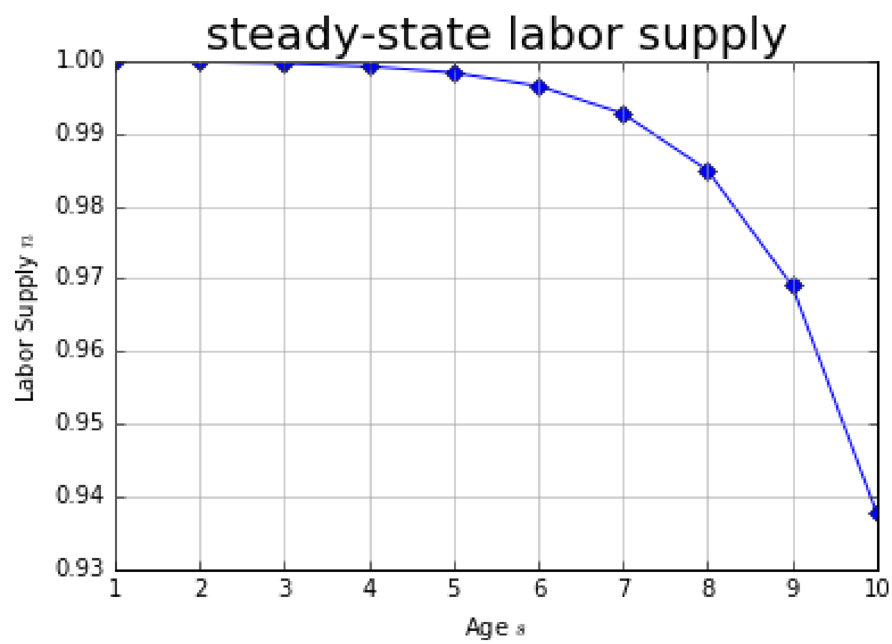
It took 0.039 seconds to get the steady state solution.

2.

The following is the steady state path of consumption and saving distribution:



And the following is the labor supply graph:



3.

The following is the steady state path of consumption and saving distribution for $\alpha = 0.25$:

```
ss_output_alpha0.25 = \
     {'C_ss': 5.7036127386763988,
     'EulErr_ss': array([ -8.88178420e-15,   8.88178420e-15,  -5.32907052e-15,
               8.88178420e-15,  -8.88178420e-16,   5.32907052e-15,
              -5.32907052e-15,   5.77315973e-15,  -4.44089210e-16,
               1.99840144e-13,   1.32338585e-13,  -2.48689958e-14,
              -1.43440815e-13,   1.02140518e-14,  -1.19904087e-14,
              -4.44089210e-15,   2.24265051e-14,  -1.77635684e-15,
               2.44249065e-15]),
     'K_ss': 1.5716136055694354,
     'L_ss': 9.865272629087368,
     'RCerr_ss': 1.1102230246251565e-16,
     'Y_ss': 6.2325857682192014,
     'b_ss': array([ 0.0424765 ,  0.08635831,  0.13084747,  0.17444768,
0.21445154, 0.24606189,  0.26089182,  0.24441559,
 0.17166281]),
     'c_ss': array([ 0.43105166,  0.45726955,  0.48508209,  0.51458627,
0.54588499, 0.57908739,  0.61430927,  0.65167345,
0.69131023,  0.73335784]),
     'n_ss': array([ 0.99936785,  0.99896138,  0.99829411,  0.9971997 ,
0.99540729, 0.99247863,  0.98771182,  0.98000149,
```

```
0.96765461,  0.94819575]),
     'r_ss': 0.65485142713504463,
     'ss_time': 0.02634410654900421,
     'w_ss': 0.47382768849002721}
```

As α decreases, we can see that labor supply goes up; saving goes down; wage goes up; rent goes down.

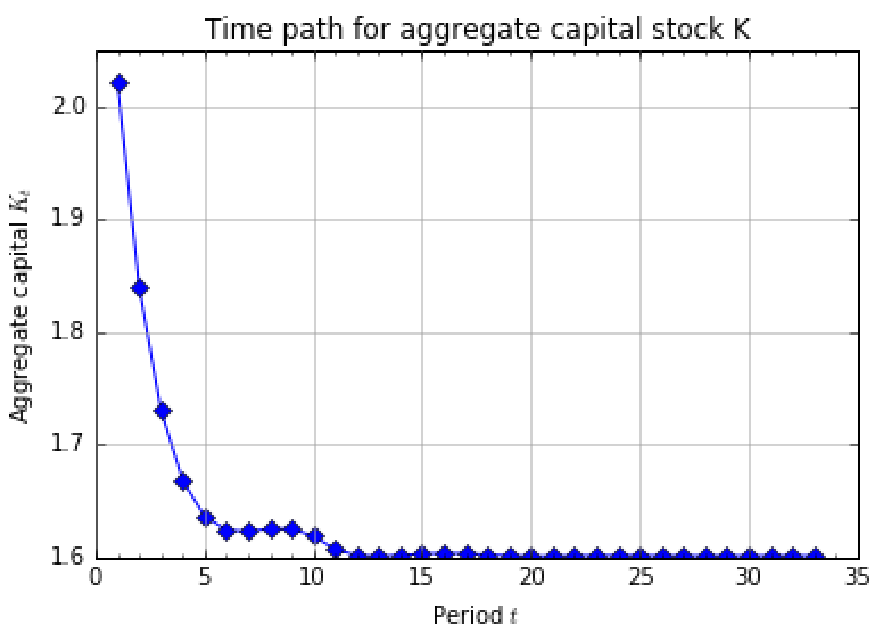
We know that the economy is going to be a more labor intensive econom, where the marginal benefit of adding labor is greater than the marginal benefit of adding one unit of capital.

This is translated into the result where marginal product of labor is higher than the marginal product of captial relatively. Hence wage is going to increase relatively to rent. Hence the agent has the incentive to add more labor supply and dcrease their savings.

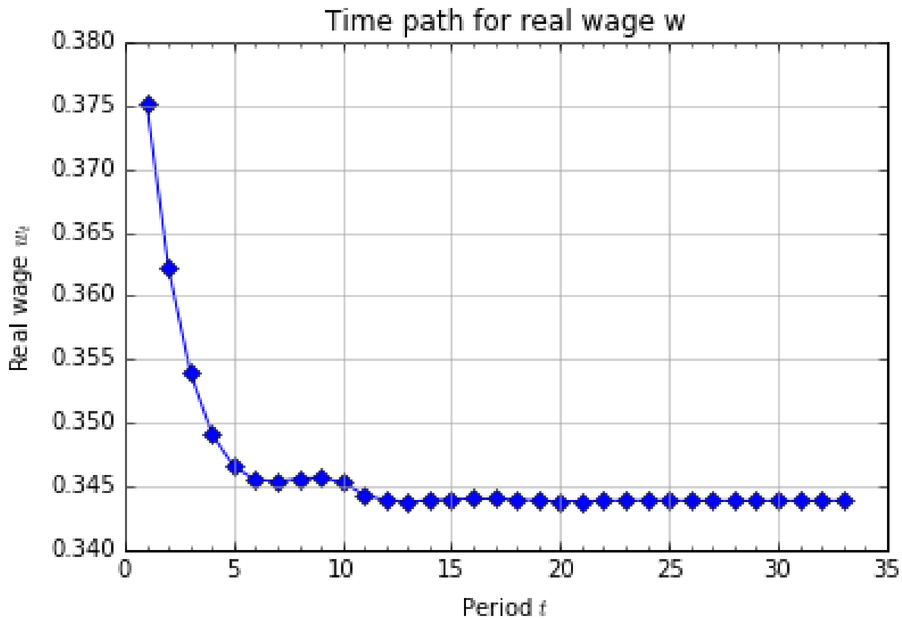
Problem 4

1.

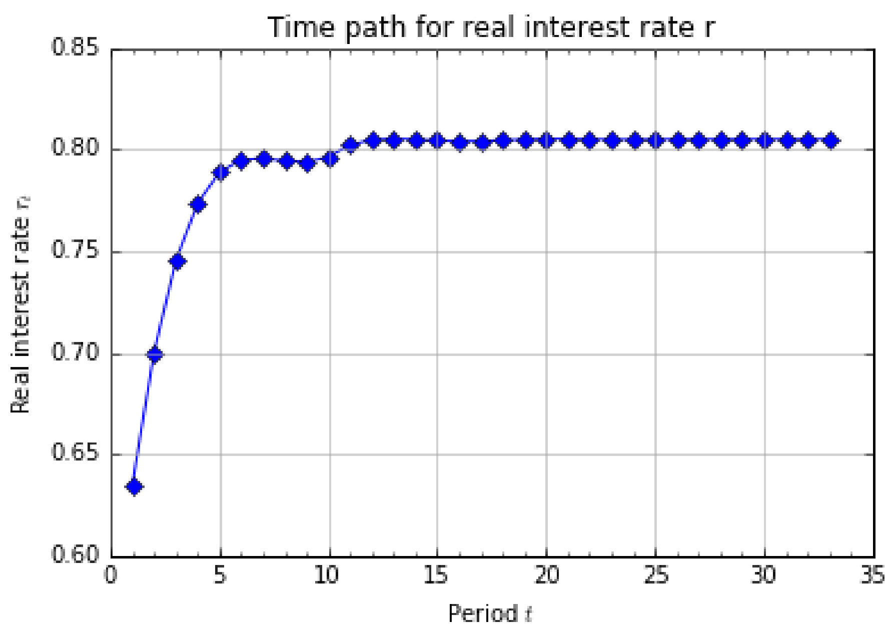
Solving the TPI problem, we get the following graph:
Captial Stock:



Wage path:

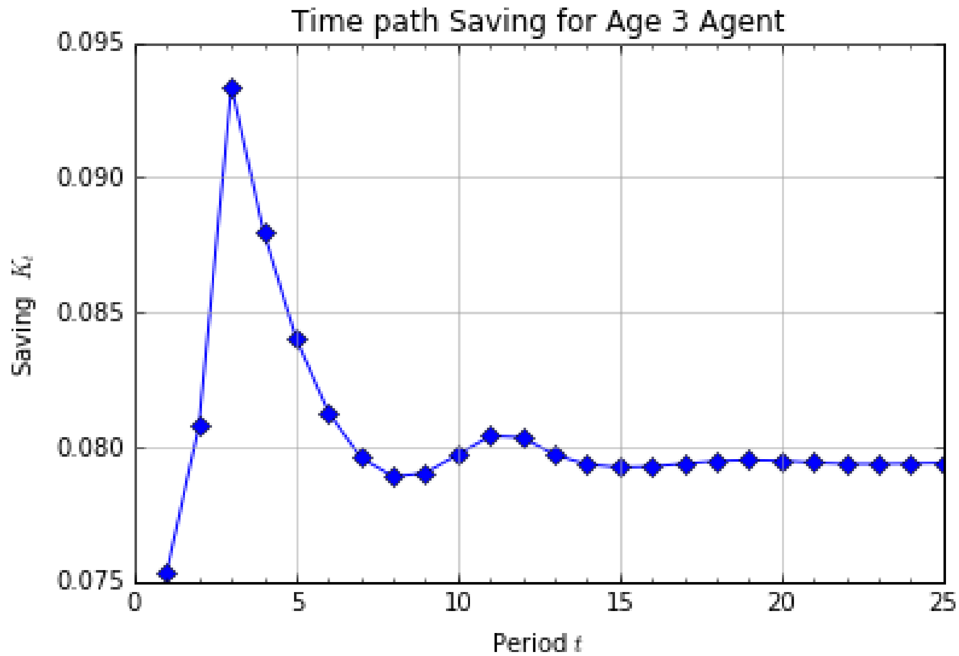


Rent Path:



2.

The following is the time path of Saving for agent of age 3:



The following is the time path of Labor Supply for agent of age 3: