Problem Set 2

MACS 30250

Submitted by - Nipun Thakurele

In [1]:

```
import numpy as np
import scipy.optimize as opt
import scipy.interpolate as intpl
import matplotlib.pyplot as plt
%matplotlib notebook
```

Question 1

In [2]:

```
# Set up the parameters
beta = 0.9
gamma = 2.2
W_min = 0.1
W_max = 10.0
W_size = 30
W_vec = np.linspace(W_min, W_max, W_size)
V_t = np.log(W_vec)

def util_CRRA(W, W_pr, gamma):
    # Define CRRA utility function
    c = W - W_pr
    util = (c ** (1 - gamma) - 1) / (1 - gamma)

    return util

def neg_V_iid(W_pr, *args):
    W, eps, util, Exp_V_t_interp, gamma, beta = args
    Vtp1 = np.exp(eps)*util(W, W_pr, gamma) + beta * Exp_V_t_interp(W_pr)
    neg_Vtp1 = -Vtp1

    return neg_Vtp1
```

Question 1(a)

In [3]:

```
eps_prob = np.array([0.1, 0.2, 0.4, 0.2, 0.1])
eps_vec = np.array([-1.40, -0.55, 0.0, 0.55, 1.4])
eps_size = eps_vec.shape[0]

V_init = np.zeros((W_size, eps_size))
V_new = V_init.copy()

VF_iter = 0
VF_dist = 10
VF_maxiter = 200
VF_mindist = 1e-8

while (VF_iter < VF_maxiter) and (VF_dist > VF_mindist):
    VF_iter += 1
    V_init = V_new.copy()
    V_new = np.zeros((W_size, eps_size))
    psi_mat = np.zeros((W_size, eps_size))

    Exp_V = V_init @ eps_prob.reshape((eps_size, 1))
```

```python
    Exp_V_interp = intpl.interp1d(W_vec, Exp_V.flatten(), kind='cubic',
                                  fill_value='extrapolate')

    for eps_ind in range(eps_size):
        for W_ind in range(W_size):
            W = W_vec[W_ind]
            eps = eps_vec[eps_ind]
            V_args = (W, eps, util_CRRA, Exp_V_interp, gamma, beta)
            results_1 = opt.minimize_scalar(neg_V_iid, bounds=(1e-10, W - 1e-10),
                                            args=V_args, method='bounded')
            V_new[W_ind, eps_ind] = -results_1.fun
            psi_mat[W_ind, eps_ind] = results_1.x

    VF_dist = ((V_init - V_new) ** 2).sum()
    print('VF_iter=', VF_iter, ', VF_dist=', VF_dist)
```

```
VF_iter= 1 , VF_dist= 3494.416552492849
VF_iter= 2 , VF_dist= 3288.9775602179398
VF_iter= 3 , VF_dist= 4368.033199294504
VF_iter= 4 , VF_dist= 5171.123172733101
VF_iter= 5 , VF_dist= 5690.816865389136
VF_iter= 6 , VF_dist= 5962.334129328155
VF_iter= 7 , VF_dist= 6029.410863441401
VF_iter= 8 , VF_dist= 5938.006359170556
VF_iter= 9 , VF_dist= 5728.733568262043
VF_iter= 10 , VF_dist= 5436.089591828773
VF_iter= 11 , VF_dist= 5088.450567375278
VF_iter= 12 , VF_dist= 4708.595711207952
VF_iter= 13 , VF_dist= 4314.425870978352
VF_iter= 14 , VF_dist= 3919.6529865086823
VF_iter= 15 , VF_dist= 3534.453890340005
VF_iter= 16 , VF_dist= 3166.0717799094914
VF_iter= 17 , VF_dist= 2819.3724059115366
VF_iter= 18 , VF_dist= 2497.331372214219
VF_iter= 19 , VF_dist= 2201.4557625475145
VF_iter= 20 , VF_dist= 1932.1399178953566
VF_iter= 21 , VF_dist= 1688.9596638065682
VF_iter= 22 , VF_dist= 1470.9108719157662
VF_iter= 23 , VF_dist= 1276.5993702275612
VF_iter= 24 , VF_dist= 1104.3893159739218
VF_iter= 25 , VF_dist= 952.5168794056659
VF_iter= 26 , VF_dist= 819.1754636558386
VF_iter= 27 , VF_dist= 702.5782302191728
VF_iter= 28 , VF_dist= 601.0007639998626
VF_iter= 29 , VF_dist= 512.8120804049411
VF_iter= 30 , VF_dist= 436.49195996903325
VF_iter= 31 , VF_dist= 370.64094039451504
VF_iter= 32 , VF_dist= 313.9837850772325
VF_iter= 33 , VF_dist= 265.36837905541597
VF_iter= 34 , VF_dist= 223.76142626173007
VF_iter= 35 , VF_dist= 188.24202140834456
VF_iter= 36 , VF_dist= 157.99391929400753
VF_iter= 37 , VF_dist= 132.29712064193444
VF_iter= 38 , VF_dist= 110.51922993676429
VF_iter= 39 , VF_dist= 92.10691083376445
VF_iter= 40 , VF_dist= 76.57766324529271
VF_iter= 41 , VF_dist= 63.512068135772914
VF_iter= 42 , VF_dist= 52.54658698138166
VF_iter= 43 , VF_dist= 43.3669590697024
VF_iter= 44 , VF_dist= 35.702208079202634
VF_iter= 45 , VF_dist= 29.31924705157791
VF_iter= 46 , VF_dist= 24.018055757038024
VF_iter= 47 , VF_dist= 19.62739479941466
VF_iter= 48 , VF_dist= 16.001015214233433
VF_iter= 49 , VF_dist= 13.014319735746323
VF_iter= 50 , VF_dist= 10.56143149616109
VF_iter= 51 , VF_dist= 8.552627061405703
VF_iter= 52 , VF_dist= 6.912092964935729
VF_iter= 53 , VF_dist= 5.575967859534671
VF_iter= 54 , VF_dist= 4.490635846895642
VF_iter= 55 , VF_dist= 3.611240167867626
VF_iter= 56 , VF_dist= 2.9003900724273053
VF_iter= 57 , VF_dist= 2.3270371537171513
VF_iter= 58 , VF_dist= 1.8655005992880302
VF_iter= 59 , VF_dist= 1.4946235941207497
VF_iter= 60 , VF_dist= 1.1970454632792984
```

```
VF_iter= 61 , VF_dist= 0.9585760728643425
VF_iter= 62 , VF_dist= 0.7676605629606206
VF_iter= 63 , VF_dist= 0.6149237330662378
VF_iter= 64 , VF_dist= 0.4927844192420875
VF_iter= 65 , VF_dist= 0.3951310655441188
VF_iter= 66 , VF_dist= 0.31705046239942836
VF_iter= 67 , VF_dist= 0.25460234391745856
VF_iter= 68 , VF_dist= 0.20463323107554093
VF_iter= 69 , VF_dist= 0.16462358821635545
VF_iter= 70 , VF_dist= 0.1325630262736468
VF_iter= 71 , VF_dist= 0.10684893010182255
VF_iter= 72 , VF_dist= 0.08620449950960188
VF_iter= 73 , VF_dist= 0.06961276460117397
VF_iter= 74 , VF_dist= 0.05626365791980316
VF_iter= 75 , VF_dist= 0.045511694260017735
VF_iter= 76 , VF_dist= 0.03684222157750761
VF_iter= 77 , VF_dist= 0.029844564402374686
VF_iter= 78 , VF_dist= 0.024190686856972787
VF_iter= 79 , VF_dist= 0.019618260386259922
VF_iter= 80 , VF_dist= 0.015917236304987616
VF_iter= 81 , VF_dist= 0.012919200660435864
VF_iter= 82 , VF_dist= 0.010488934004746101
VF_iter= 83 , VF_dist= 0.008517716355044384
VF_iter= 84 , VF_dist= 0.00691801246777643
VF_iter= 85 , VF_dist= 0.005619248530286449
VF_iter= 86 , VF_dist= 0.004564451958587326
VF_iter= 87 , VF_dist= 0.0037075740906608633
VF_iter= 88 , VF_dist= 0.0030113536420462708
VF_iter= 89 , VF_dist= 0.0024456088523578355
VF_iter= 90 , VF_dist= 0.001985869936846679
VF_iter= 91 , VF_dist= 0.0016122821021656536
VF_iter= 92 , VF_dist= 0.0013087240486581234
VF_iter= 93 , VF_dist= 0.0010620984074924253
VF_iter= 94 , VF_dist= 0.0008617596247306495
VF_iter= 95 , VF_dist= 0.0006990519398148664
VF_iter= 96 , VF_dist= 0.0005669357206090445
VF_iter= 97 , VF_dist= 0.0004596848484600202
VF_iter= 98 , VF_dist= 0.0003726413433400984
VF_iter= 99 , VF_dist= 0.00030201618674586747
VF_iter= 100 , VF_dist= 0.00024472749354110894
VF_iter= 101 , VF_dist= 0.00019826892637177278
VF_iter= 102 , VF_dist= 0.00016060263224256825
VF_iter= 103 , VF_dist= 0.0001300720891483579
VF_iter= 104 , VF_dist= 0.00010533113515101626
VF_iter= 105 , VF_dist= 8.528616230351807e-05
VF_iter= 106 , VF_dist= 6.904902893045537e-05
VF_iter= 107 , VF_dist= 5.58987032119717e-05
VF_iter= 108 , VF_dist= 4.525002255832318e-05
VF_iter= 109 , VF_dist= 3.662825367946391e-05
VF_iter= 110 , VF_dist= 2.9648381684870055e-05
VF_iter= 111 , VF_dist= 2.39982549749399e-05
VF_iter= 112 , VF_dist= 1.9424873004279235e-05
VF_iter= 113 , VF_dist= 1.5723235673077267e-05
VF_iter= 114 , VF_dist= 1.272727964557564e-05
VF_iter= 115 , VF_dist= 1.0302514478394718e-05
VF_iter= 116 , VF_dist= 8.34004242997388e-06
VF_iter= 117 , VF_dist= 6.751704204548888e-06
VF_iter= 118 , VF_dist= 5.466140377418238e-06
VF_iter= 119 , VF_dist= 4.425597146845334e-06
VF_iter= 120 , VF_dist= 3.5833367766181355e-06
VF_iter= 121 , VF_dist= 2.901539092470968e-06
VF_iter= 122 , VF_dist= 2.3496014077785377e-06
VF_iter= 123 , VF_dist= 1.9027617856574288e-06
VF_iter= 124 , VF_dist= 1.5409843211358737e-06
VF_iter= 125 , VF_dist= 1.2480569000018439e-06
VF_iter= 126 , VF_dist= 1.0108610126543219e-06
VF_iter= 127 , VF_dist= 8.187809298941962e-07
VF_iter= 128 , VF_dist= 6.632257181555538e-07
VF_iter= 129 , VF_dist= 5.372425577246577e-07
VF_iter= 130 , VF_dist= 4.352039711978974e-07
VF_iter= 131 , VF_dist= 3.52554804532906e-07
VF_iter= 132 , VF_dist= 2.856075891862148e-07
VF_iter= 133 , VF_dist= 2.3137700024210983e-07
VF_iter= 134 , VF_dist= 1.8744595280674418e-07
VF_iter= 135 , VF_dist= 1.518572697400391e-07
VF_iter= 136 , VF_dist= 1.2302604725753922e-07
VF_iter= 137 , VF_dist= 9.966874322757488e-08
```

```
VF_iter= 138 , VF_dist= 8.074580018350248e-08
VF_iter= 139 , VF_dist= 6.541520832632279e-08
VF_iter= 140 , VF_dist= 5.299492534475509e-08
VF_iter= 141 , VF_dist= 4.293245385716458e-08
VF_iter= 142 , VF_dist= 3.4780217313545753e-08
VF_iter= 143 , VF_dist= 2.817561810545218e-08
VF_iter= 144 , VF_dist= 2.282489479627886e-08
VF_iter= 145 , VF_dist= 1.8490046714220514e-08
VF_iter= 146 , VF_dist= 1.4978247147702045e-08
VF_iter= 147 , VF_dist= 1.213326600466 7127e-08
VF_iter= 148 , VF_dist= 9.828523946527788e-09
```

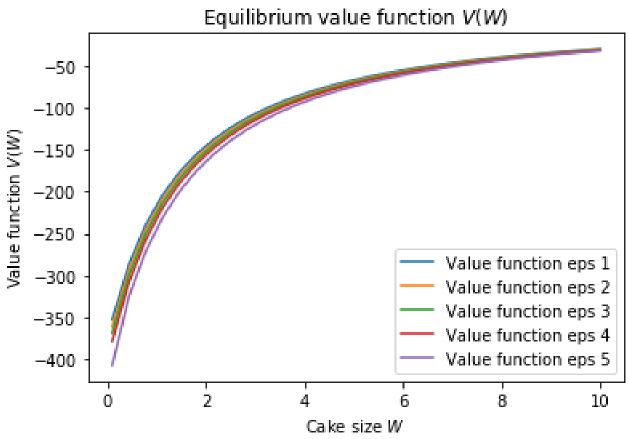
Question 1(b)

In [4]:

```
plt.plot(W_vec, V_new[:, 0], label='Value function eps 1')
plt.plot(W_vec, V_new[:, 1], label='Value function eps 2')
plt.plot(W_vec, V_new[:, 2], label='Value function eps 3')
plt.plot(W_vec, V_new[:, 3], label='Value function eps 4')
plt.plot(W_vec, V_new[:, 4], label='Value function eps 5')
plt.title('Equilibrium value function $V(W)$')
plt.xlabel('Cake size $W$')
plt.ylabel('Value function $V(W)$')
plt.legend()
```

Out[4]:

```
<matplotlib.legend.Legend at 0x117a859b0>
```



Question 1(c)

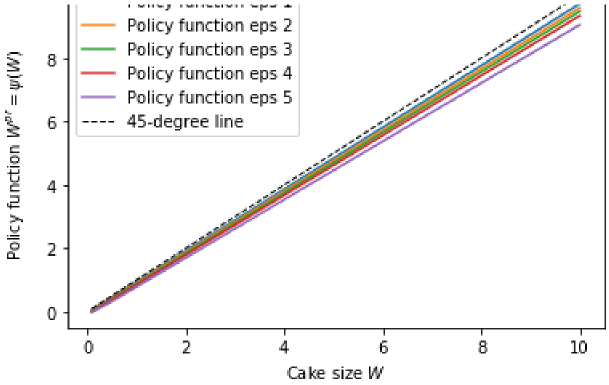
In [5]:

```
plt.plot(W_vec, psi_mat[:, 0], label='Policy function eps 1')
plt.plot(W_vec, psi_mat[:, 1], label='Policy function eps 2')
plt.plot(W_vec, psi_mat[:, 2], label='Policy function eps 3')
plt.plot(W_vec, psi_mat[:, 3], label='Policy function eps 4')
plt.plot(W_vec, psi_mat[:, 4], label='Policy function eps 5')

plt.plot(W_vec, W_vec, color='black', linewidth=1, linestyle='--',
         label='45-degree line')
plt.title('Cake size tomorrow policy function $\psi(W)$')
plt.xlabel('Cake size $W$')
plt.ylabel('Policy function $W^{pr}=\psi(W)$')
plt.legend()
```

Out[5]:

```
<matplotlib.legend.Legend at 0x117bb85c0>
```



Question 2

In [6]:

```
trans_mat = np.array([[0.4,0.28,0.18,0.1,0.04],
                      [0.2, 0.4, 0.2, 0.13, 0.17],
                      [0.1, 0.2, 0.4, 0.2, 0.1],
                      [0.07, 0.13, 0.2, 0.4, 0.2],
                      [0.04, 0.1, 0.18, 0.28, 0.4]])
```

Question 2(a)

In [7]:

```
V_init = np.zeros((W_size, eps_size))
V_new = V_init.copy()

VF_iter = 0
VF_dist = 10
VF_maxiter = 200
VF_mindist = 1e-8

while (VF_iter < VF_maxiter) and (VF_dist > VF_mindist):
    VF_iter += 1
    V_init = V_new.copy()
    V_new = np.zeros((W_size, eps_size))
    psi_mat = np.zeros((W_size, eps_size))

    for eps_ind in range(eps_size):
        trans_mat_ind  = trans_mat[eps_ind,:]
        Exp_V = V_init @ trans_mat_ind.reshape((eps_size, 1))
        Exp_V_interp = intpl.interp1d(W_vec, Exp_V.flatten(), kind='cubic',
                                      fill_value='extrapolate')

        for W_ind in range(W_size):
            W = W_vec[W_ind]
            eps = eps_vec[eps_ind]
            V_args = (W, eps, util_CRRA, Exp_V_interp, gamma, beta)
            results1 = opt.minimize_scalar(neg_V_iid, bounds=(1e-10, W - 1e-10),
                                           args=V_args, method='bounded')
            V_new[W_ind, eps_ind] = -results1.fun
            psi_mat[W_ind, eps_ind] = results1.x

    VF_dist = ((V_init - V_new) ** 2).sum()
    print('VF_iter=', VF_iter, ', VF_dist=', VF_dist)
```

```
VF_iter= 1 , VF_dist= 3494.416552492849
VF_iter= 2 , VF_dist= 5195.909710673112
VF_iter= 3 , VF_dist= 6537.789177725918
VF_iter= 4 , VF_dist= 7782.294929790439
VF_iter= 5 , VF_dist= 8748.293763205218
VF_iter= 6 , VF_dist= 9421.062136450677
VF_iter= 7 , VF_dist= 9830.62784526284
VF_iter= 8 , VF_dist= 10015.136371648436
VF_iter= 9 , VF_dist= 10012.445686363524
VF_iter= 10 , VF_dist= 9857.383167965068
```

```
VF_iter= 11 , VF_dist= 9581.890912598756
VF_iter= 12 , VF_dist= 9214.360736617782
VF_iter= 13 , VF_dist= 8779.460736215107
VF_iter= 14 , VF_dist= 8298.225746363109
VF_iter= 15 , VF_dist= 7788.279219494456
VF_iter= 16 , VF_dist= 7264.1305312291315
VF_iter= 17 , VF_dist= 6737.507687989991
VF_iter= 18 , VF_dist= 6217.69795253823
VF_iter= 19 , VF_dist= 5711.879012757782
VF_iter= 20 , VF_dist= 5225.430082022241
VF_iter= 21 , VF_dist= 4762.216951759559
VF_iter= 22 , VF_dist= 4324.848287404181
VF_iter= 23 , VF_dist= 3914.9025598025523
VF_iter= 24 , VF_dist= 3533.126427748155
VF_iter= 25 , VF_dist= 3179.6061116370283
VF_iter= 26 , VF_dist= 2853.9138194052566
VF_iter= 27 , VF_dist= 2555.2315330466695
VF_iter= 28 , VF_dist= 2282.454537270004
VF_iter= 29 , VF_dist= 2034.2770527571029
VF_iter= 30 , VF_dist= 1809.2622316695986
VF_iter= 31 , VF_dist= 1605.8986287225932
VF_iter= 32 , VF_dist= 1422.645085041609
VF_iter= 33 , VF_dist= 1257.965775812599
VF_iter= 34 , VF_dist= 1110.356988113756
VF_iter= 35 , VF_dist= 978.3670118405364
VF_iter= 36 , VF_dist= 860.610353399984
VF_iter= 37 , VF_dist= 755.777321937308
VF_iter= 38 , VF_dist= 662.6398903667689
VF_iter= 39 , VF_dist= 580.0546002736296
VF_iter= 40 , VF_dist= 506.96316001765763
VF_iter= 41 , VF_dist= 442.39127986697775
VF_iter= 42 , VF_dist= 385.44619440111916
VF_iter= 43 , VF_dist= 335.3132415807365
VF_iter= 44 , VF_dist= 291.251797629058
VF_iter= 45 , VF_dist= 252.59080737764938
VF_iter= 46 , VF_dist= 218.72409876846714
VF_iter= 47 , VF_dist= 189.10562829621603
VF_iter= 48 , VF_dist= 163.24476859649047
VF_iter= 49 , VF_dist= 140.70172093426316
VF_iter= 50 , VF_dist= 121.08311151015789
VF_iter= 51 , VF_dist= 104.03781221387302
VF_iter= 52 , VF_dist= 89.25301110872702
VF_iter= 53 , VF_dist= 76.45054649762139
VF_iter= 54 , VF_dist= 65.38350938908263
VF_iter= 55 , VF_dist= 55.833112368059076
VF_iter= 56 , VF_dist= 47.60581787228346
VF_iter= 57 , VF_dist= 40.530715288786574
VF_iter= 58 , VF_dist= 34.45713381769707
VF_iter= 59 , VF_dist= 29.25247653460704
VF_iter= 60 , VF_dist= 24.80026022766808
VF_iter= 61 , VF_dist= 20.998345344661224
VF_iter= 62 , VF_dist= 17.757340523283176
VF_iter= 63 , VF_dist= 14.99916667939204
VF_iter= 64 , VF_dist= 12.655766358041157
VF_iter= 65 , VF_dist= 10.667944941230665
VF_iter= 66 , VF_dist= 8.984331304073294
VF_iter= 67 , VF_dist= 7.560446553967289
VF_iter= 68 , VF_dist= 6.357870525544135
VF_iter= 69 , VF_dist= 5.343496706547637
VF_iter= 70 , VF_dist= 4.48886720357381
VF_iter= 71 , VF_dist= 3.7695802044486775
VF_iter= 72 , VF_dist= 3.164763141765419
VF_iter= 73 , VF_dist= 2.6566054150245875
VF_iter= 74 , VF_dist= 2.2299450861975085
VF_iter= 75 , VF_dist= 1.8719044392724276
VF_iter= 76 , VF_dist= 1.571569701734922
VF_iter= 77 , VF_dist= 1.3197105790345587
VF_iter= 78 , VF_dist= 1.1085355673814503
VF_iter= 79 , VF_dist= 0.9314792964028445
VF_iter= 80 , VF_dist= 0.7830184228635721
VF_iter= 81 , VF_dist= 0.6585128551485445
VF_iter= 82 , VF_dist= 0.554069340484342
VF_iter= 83 , VF_dist= 0.46642469448124474
VF_iter= 84 , VF_dist= 0.3928461951675637
VF_iter= 85 , VF_dist= 0.33104690048342156
VF_iter= 86 , VF_dist= 0.27911387653124037
VF_iter= 87 , VF_dist= 0.23544754285035896
VF_iter= 88 , VF_dist= 0.19871054643006057
```

```
VF_iter= 88 , VF_dist= 0.19871054643906957
VF_iter= 89 , VF_dist= 0.16778476900689582
VF_iter= 90 , VF_dist= 0.14173524881757954
VF_iter= 91 , VF_dist= 0.11977995981995102
VF_iter= 92 , VF_dist= 0.10126453614892976
VF_iter= 93 , VF_dist= 0.0856411596644798
VF_iter= 94 , VF_dist= 0.07245094279482743
VF_iter= 95 , VF_dist= 0.06130923931675407
VF_iter= 96 , VF_dist= 0.0518934029789999
VF_iter= 97 , VF_dist= 0.043932589214933405
VF_iter= 98 , VF_dist= 0.037199259816773134
VF_iter= 99 , VF_dist= 0.03150210555761637
VF_iter= 100 , VF_dist= 0.026680148541516072
VF_iter= 101 , VF_dist= 0.022597825533365257
VF_iter= 102 , VF_dist= 0.019140886807854952
VF_iter= 103 , VF_dist= 0.01621297290608622
VF_iter= 104 , VF_dist= 0.013732754989233684
VF_iter= 105 , VF_dist= 0.01163154391351098
VF_iter= 106 , VF_dist= 0.009851289324215588
VF_iter= 107 , VF_dist= 0.008342903478254917
VF_iter= 108 , VF_dist= 0.0070648556596155385
VF_iter= 109 , VF_dist= 0.005981992267100913
VF_iter= 110 , VF_dist= 0.00506454529909467
VF_iter= 111 , VF_dist= 0.004287298294294554
VF_iter= 112 , VF_dist= 0.0036288839990503284
VF_iter= 113 , VF_dist= 0.003071192378993692
VF_iter= 114 , VF_dist= 0.002598871163116776
VF_iter= 115 , VF_dist= 0.0021989040841400597
VF_iter= 116 , VF_dist= 0.0018602544297048854
VF_iter= 117 , VF_dist= 0.001573563558800882
VF_iter= 118 , VF_dist= 0.001330895732707843
VF_iter= 119 , VF_dist= 0.0011255220051619257
VF_iter= 120 , VF_dist= 0.0009517370992764942
VF_iter= 121 , VF_dist= 0.0008047041559526714
VF_iter= 122 , VF_dist= 0.0006803230660853189
VF_iter= 123 , VF_dist= 0.0005751187628412419
VF_iter= 124 , VF_dist= 0.0004861464284072534
VF_iter= 125 , VF_dist= 0.00041091103816584974
VF_iter= 126 , VF_dist= 0.0003472990666163479
VF_iter= 127 , VF_dist= 0.0002935205135074363
VF_iter= 128 , VF_dist= 0.0002480596918466168
VF_iter= 129 , VF_dist= 0.00020963345524314093
VF_iter= 130 , VF_dist= 0.0001771557437783312
VF_iter= 131 , VF_dist= 0.00014970749782921146
VF_iter= 132 , VF_dist= 0.00012651113088476217
VF_iter= 133 , VF_dist= 0.0001069088761491525
VF_iter= 134 , VF_dist= 9.034442187956188e-05
VF_iter= 135 , VF_dist= 7.63473418629601e-05
VF_iter= 136 , VF_dist= 6.451989720191396e-05
VF_iter= 137 , VF_dist= 5.4525853025489977e-05
VF_iter= 138 , VF_dist= 4.60810042088902e-05
VF_iter= 139 , VF_dist= 3.8945151232940654e-05
VF_iter= 140 , VF_dist= 3.291530678150012e-05
VF_iter= 141 , VF_dist= 2.781994525600842e-05
VF_iter= 142 , VF_dist= 2.351413652115173e-05
VF_iter= 143 , VF_dist= 1.9875429192709617e-05
VF_iter= 144 , VF_dist= 1.6800368523235177e-05
VF_iter= 145 , VF_dist= 1.4201552109292216e-05
VF_iter= 146 , VF_dist= 1.2005140360091325e-05
VF_iter= 147 , VF_dist= 1.014875234655472e-05
VF_iter= 148 , VF_dist= 8.579687353182364e-06
VF_iter= 149 , VF_dist= 7.253422181298792e-06
VF_iter= 150 , VF_dist= 6.132341440466651e-06
VF_iter= 151 , VF_dist= 5.184664827746173e-06
VF_iter= 152 , VF_dist= 4.383541000793488e-06
VF_iter= 153 , VF_dist= 3.7062822558704594e-06
VF_iter= 154 , VF_dist= 3.133717978595725e-06
VF_iter= 155 , VF_dist= 2.6496486297548373e-06
VF_iter= 156 , VF_dist= 2.2403846119759284e-06
VF_iter= 157 , VF_dist= 1.8943567501376207e-06
VF_iter= 158 , VF_dist= 1.601787326662814e-06
VF_iter= 159 , VF_dist= 1.3544122504855372e-06
VF_iter= 160 , VF_dist= 1.1452463460584366e-06
VF_iter= 161 , VF_dist= 9.68385142628241e-07
VF_iter= 162 , VF_dist= 8.188374213154782e-07
VF_iter= 163 , VF_dist= 6.923837610306188e-07
VF_iter= 164 , VF_dist= 5.854570325534972e-07
```

```
VF_iter= 165 , VF_dist= 4.950414334304928e-07
VF_iter= 166 , VF_dist= 4.185871792220053e-07
VF_iter= 167 , VF_dist= 3.539384077504905e-07
VF_iter= 168 , VF_dist= 2.992722770000032e-07
VF_iter= 169 , VF_dist= 2.53047470666367e-07
VF_iter= 170 , VF_dist= 2.1396069615119078e-07
VF_iter= 171 , VF_dist= 1.809098960820852e-07
VF_iter= 172 , VF_dist= 1.529631598039433e-07
VF_iter= 173 , VF_dist= 1.293324573506388e-07
VF_iter= 174 , VF_dist= 1.0935140785848304e-07
VF_iter= 175 , VF_dist= 9.245649780637081e-08
VF_iter= 176 , VF_dist= 7.817120433872029e-08
VF_iter= 177 , VF_dist= 6.609256775247748e-08
VF_iter= 178 , VF_dist= 5.587982702032924e-08
VF_iter= 179 , VF_dist= 4.72448318872795e-08
VF_iter= 180 , VF_dist= 3.994391170003187e-08
VF_iter= 181 , VF_dist= 3.3771017416790086e-08
VF_iter= 182 , VF_dist= 2.8551914510611586e-08
VF_iter= 183 , VF_dist= 2.4139269419378372e-08
VF_iter= 184 , VF_dist= 2.040849946576962e-08
VF_iter= 185 , VF_dist= 1.7254260912511963e-08
VF_iter= 186 , VF_dist= 1.4587479176401499e-08
VF_iter= 187 , VF_dist= 1.2332837685834466e-08
VF_iter= 188 , VF_dist= 1.0426652239007826e-08
VF_iter= 189 , VF_dist= 8.81507724930518e-09
```

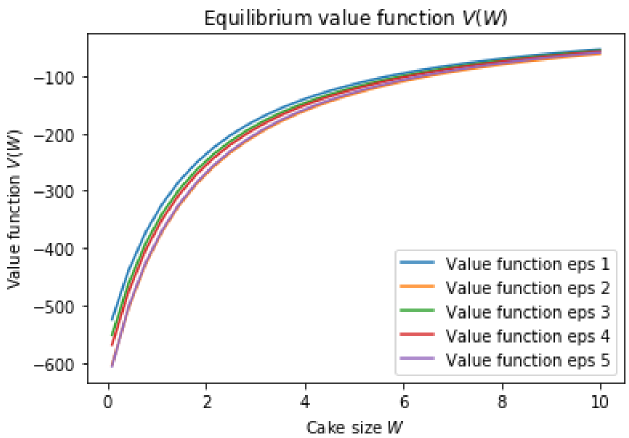
Question 2(b)

In [8]:

```
plt.plot(W_vec, V_new[:, 0], label='Value function eps 1')
plt.plot(W_vec, V_new[:, 1], label='Value function eps 2')
plt.plot(W_vec, V_new[:, 2], label='Value function eps 3')
plt.plot(W_vec, V_new[:, 3], label='Value function eps 4')
plt.plot(W_vec, V_new[:, 4], label='Value function eps 5')
plt.title('Equilibrium value function $V(W)$')
plt.xlabel('Cake size $W$')
plt.ylabel('Value function $V(W)$')
plt.legend()
```

Out[8]:

<matplotlib.legend.Legend at 0x117d11e80>



Question 2(c)

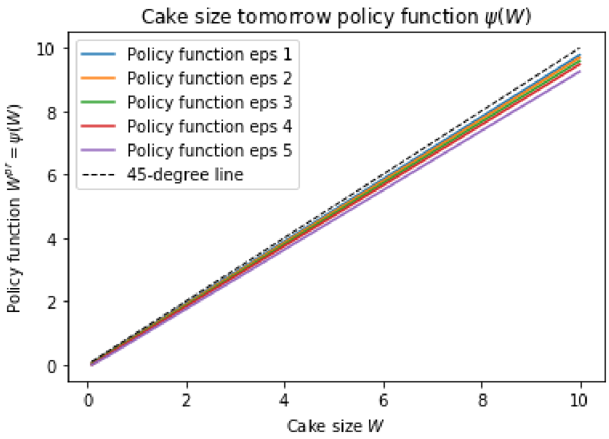
In [9]:

```
plt.plot(W_vec, psi_mat[:, 0], label='Policy function eps 1')
plt.plot(W_vec, psi_mat[:, 1], label='Policy function eps 2')
plt.plot(W_vec, psi_mat[:, 2], label='Policy function eps 3')
plt.plot(W_vec, psi_mat[:, 3], label='Policy function eps 4')
plt.plot(W_vec, psi_mat[:, 4], label='Policy function eps 5')
```

```
plt.plot(W_vec, W_vec, color='black', linewidth=1, linestyle='--',
         label='45-degree line')
plt.title('Cake size tomorrow policy function $\psi(W)$')
plt.xlabel('Cake size $W$')
plt.ylabel('Policy function $W^{pr}=\psi(W)$')
plt.legend()
```

Out[9]:

```
<matplotlib.legend.Legend at 0x117e104e0>
```



In [ ]: