COMPUTATIONAL ECONOMICS HW1

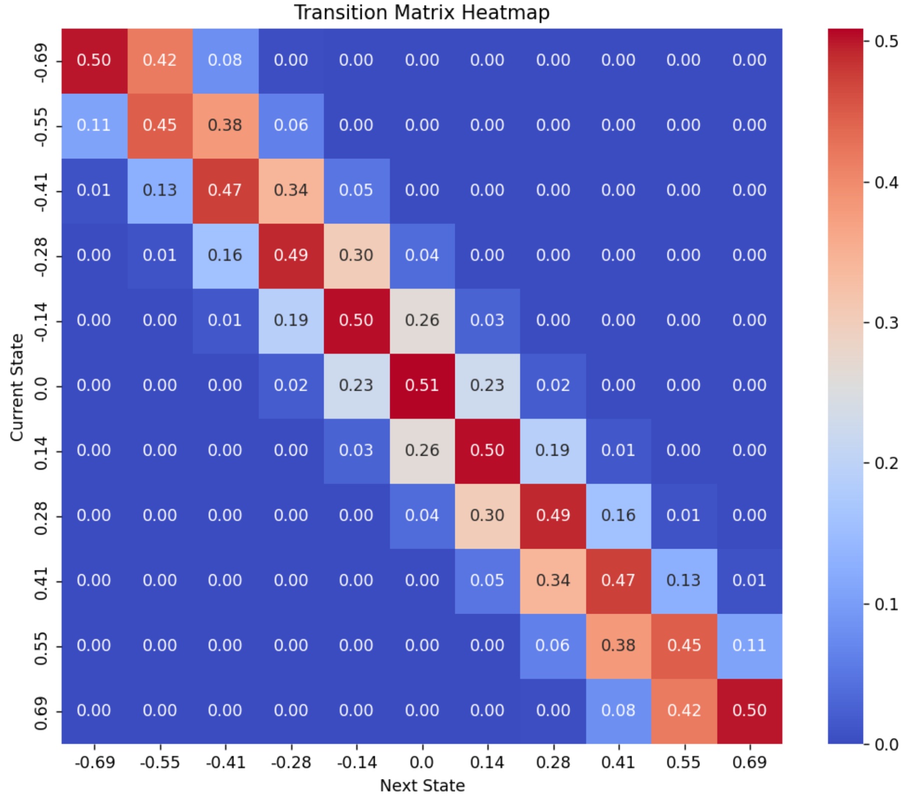
SHANGZE DAI

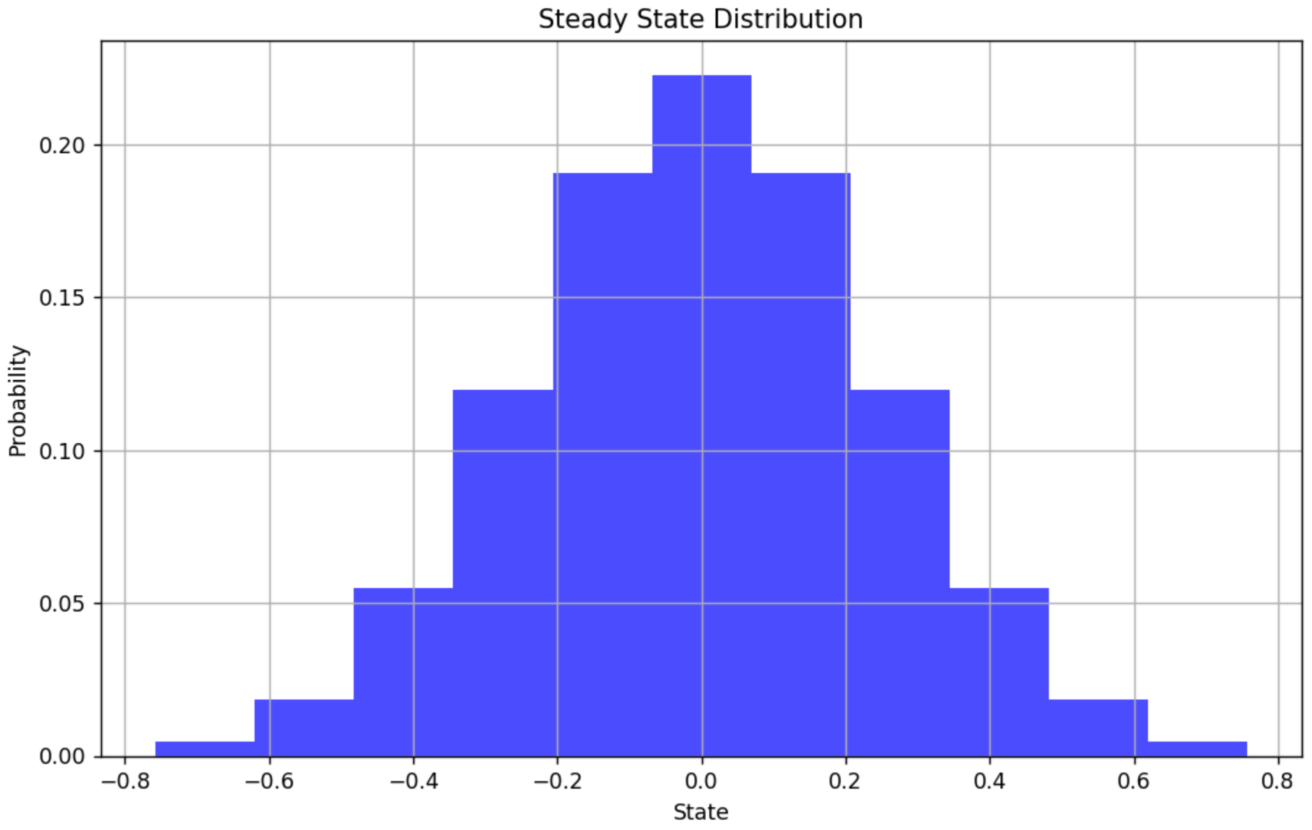
QUESTION

1.1 Solve Tauchen (86) approximation and simulate the stationary distribution $D^T(y)$

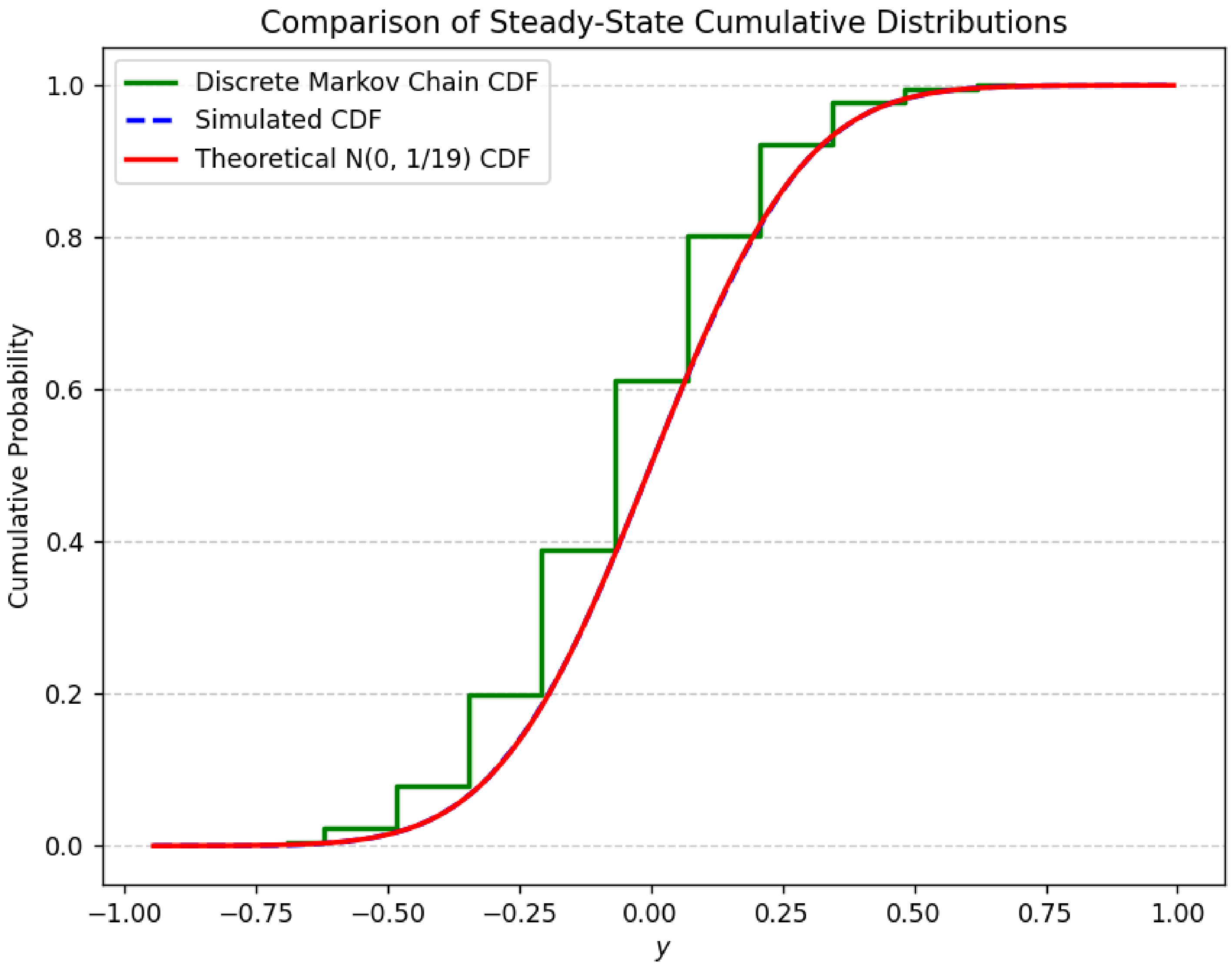
$$y_{t+1} = \rho y_t + u_{t+1} \quad N(0, \sigma_u^2) \quad \rho = 0.9, \sigma_u = 0.1, n = 11, m = 3$$

Solving M*Y=Y





QUESTION 1.2



Comparison of Steady-State Cumulative Distributions
Discrete Markov Chain CDF
Simulated CDF
Theoretical N(0, 1/19) CDF
Cumulative Probability
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
y

QUESTION 1.3&1.4

Base Case: $n = 2$

For two states ($n = 2$), the transition probability matrix is:

$$P_2 = \begin{bmatrix} p & 1-p \\ 1-q & q \end{bmatrix}$$

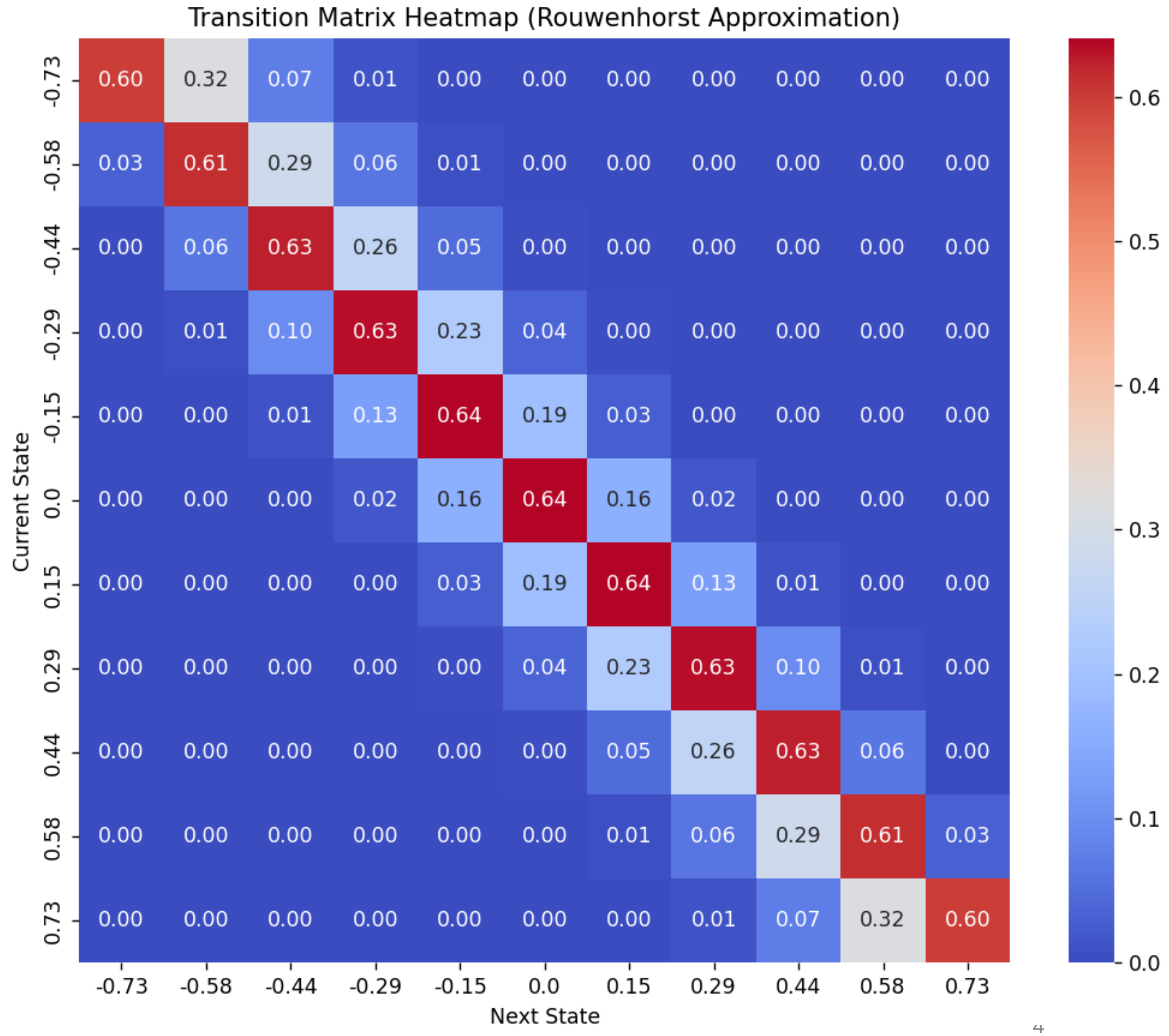
where:

- $p = \frac{1+\rho}{2}$
- $q = p$

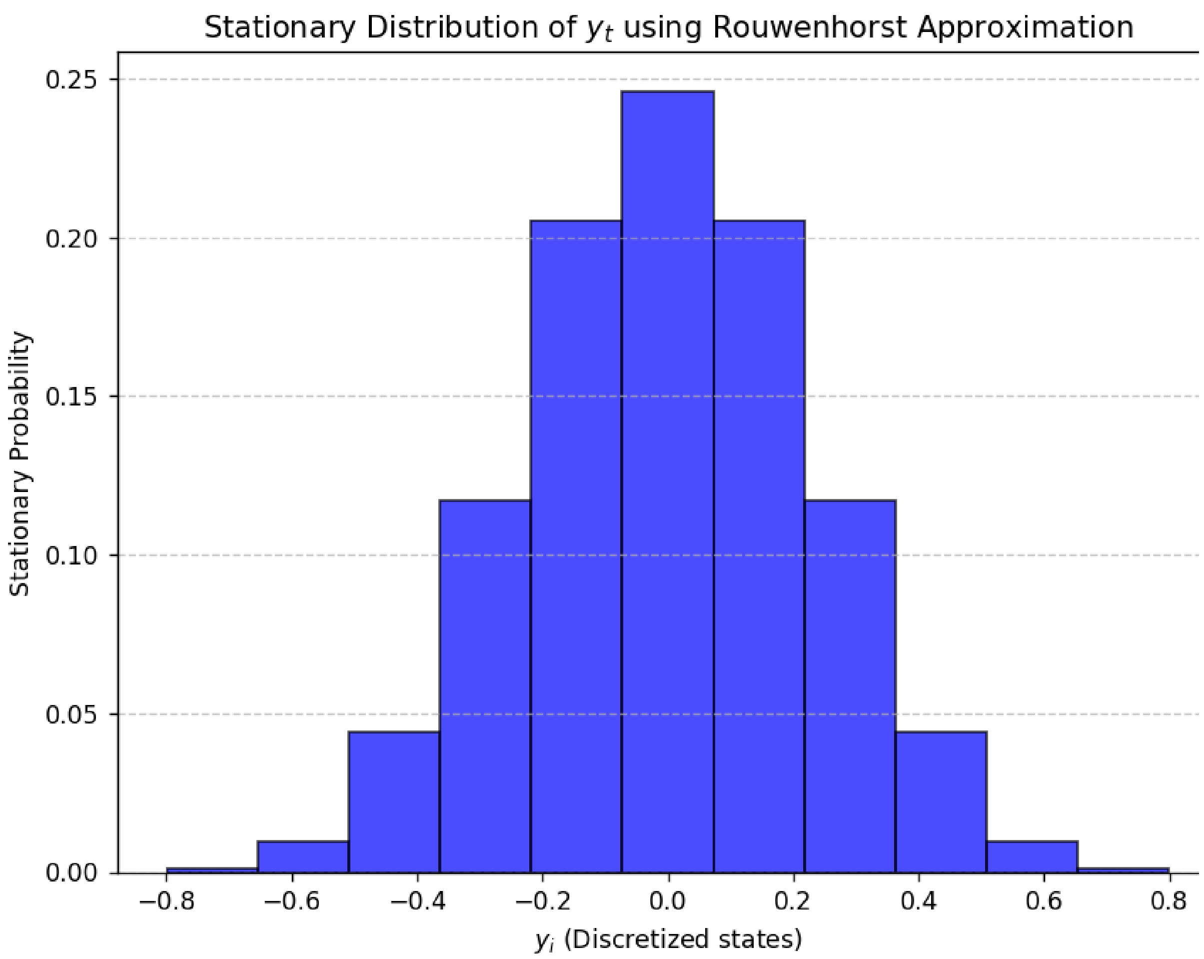
Recursive Step: Expanding to n States

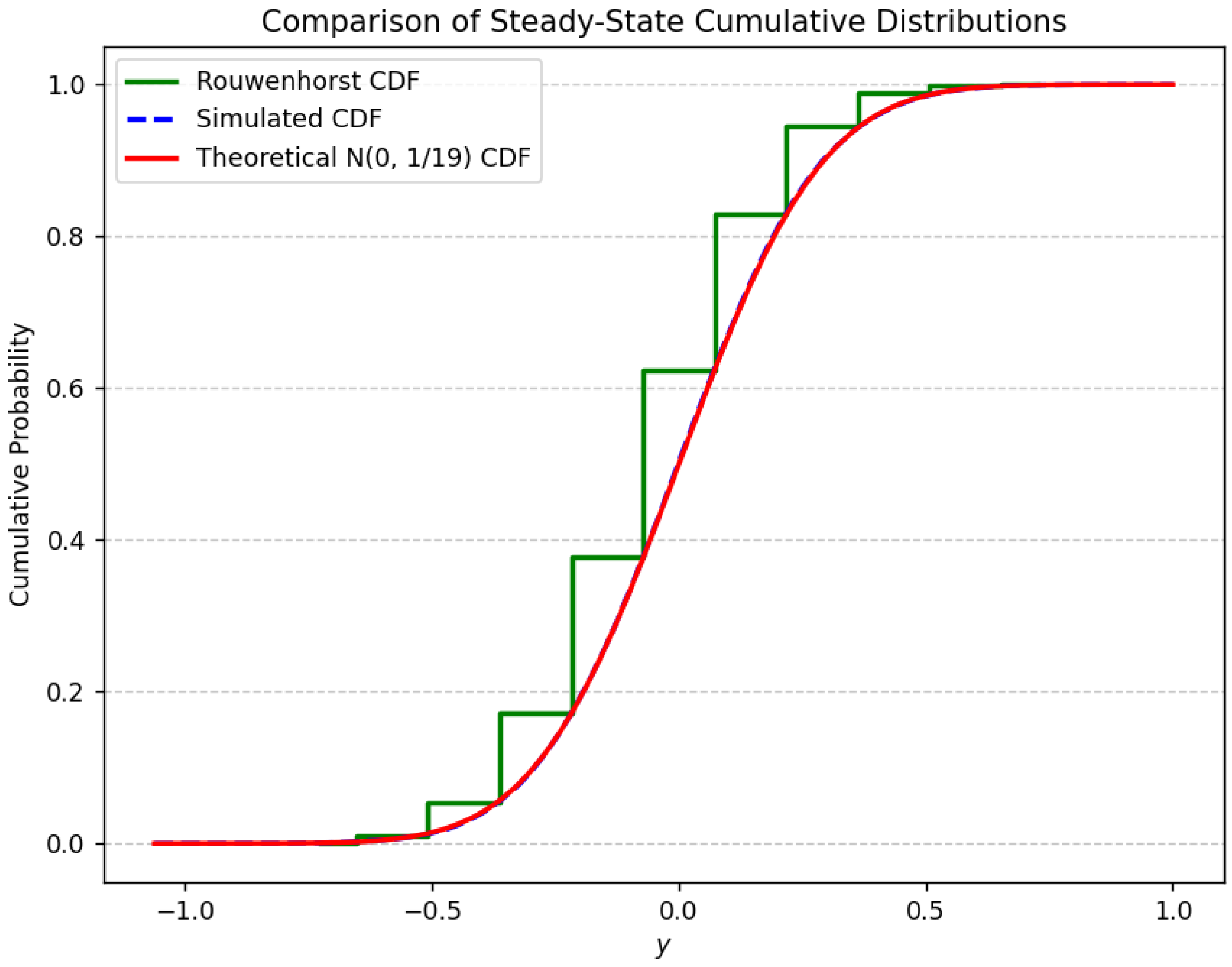
For $n > 2$, we construct P_n recursively from P_{n-1}:

$$P_n = \begin{bmatrix} pP_{n-1} & (1-p)P_{n-1} \\ (1-q)P_{n-1} & qP_{n-1} \end{bmatrix}$$



QUESTION 1.3&1.4

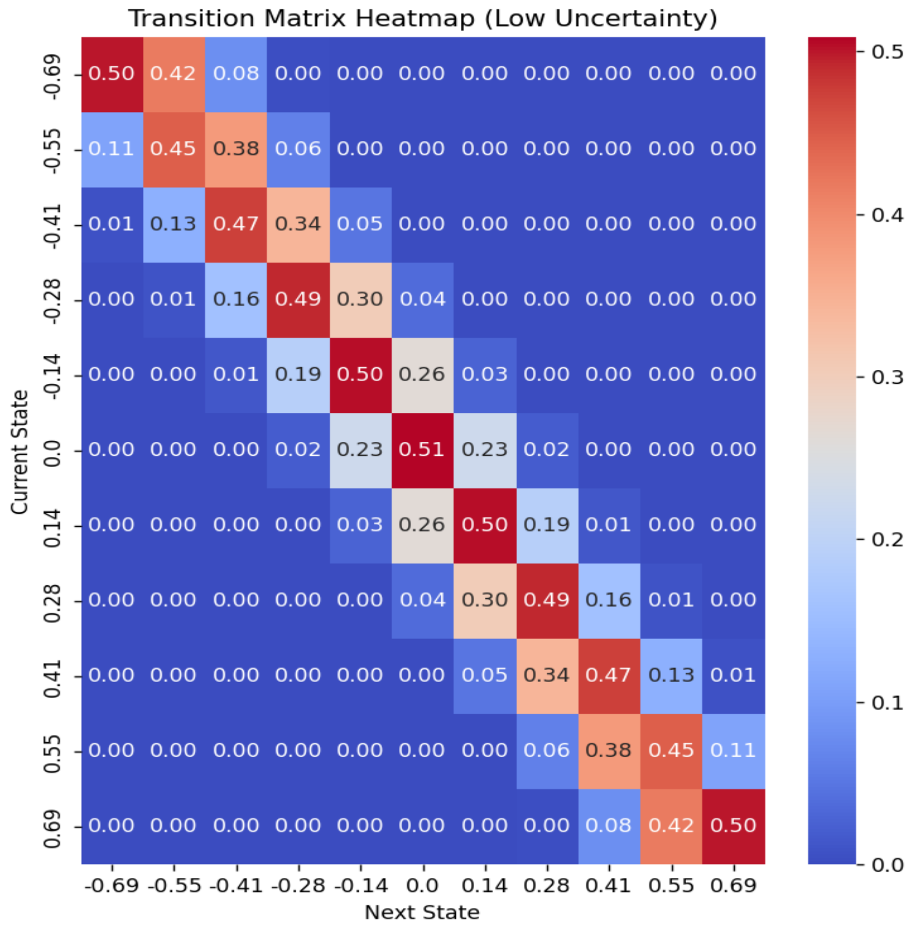


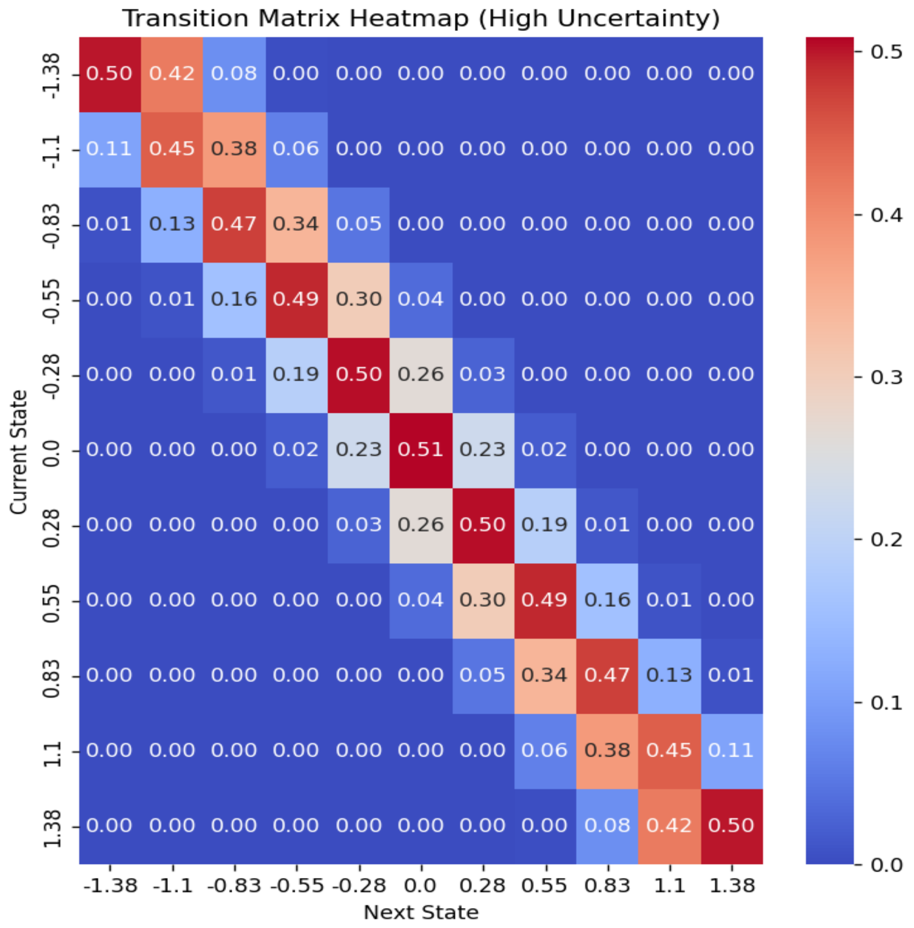


QUESTION 2.1

$$\rho = 0.9, \sigma_u = 0.1, n = 11, m = 3, \hat{\sigma}_u = 0.2$$

$$P(i,j) = \Phi\left(\frac{y_j + \delta/2 - \rho\, y_i}{\sigma}\right) - \Phi\left(\frac{y_j - \delta/2 - \rho\, y_i}{\sigma}\right)$$

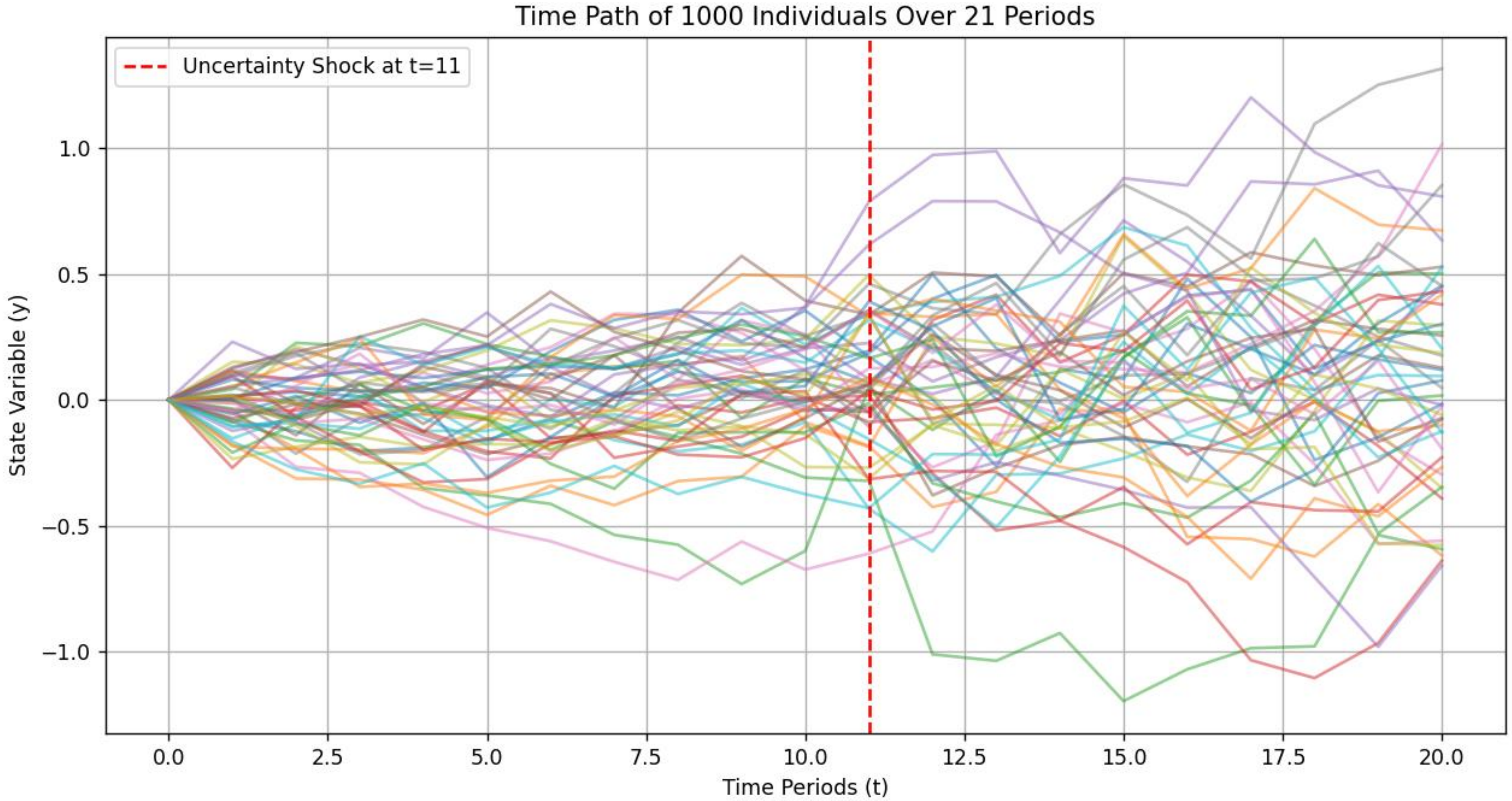




As long as the state grid is constructed according to the unconditional standard deviation (that is, the grid range and step size are scaled with σ), the transition probability matrix must be the same under the normalized scale.

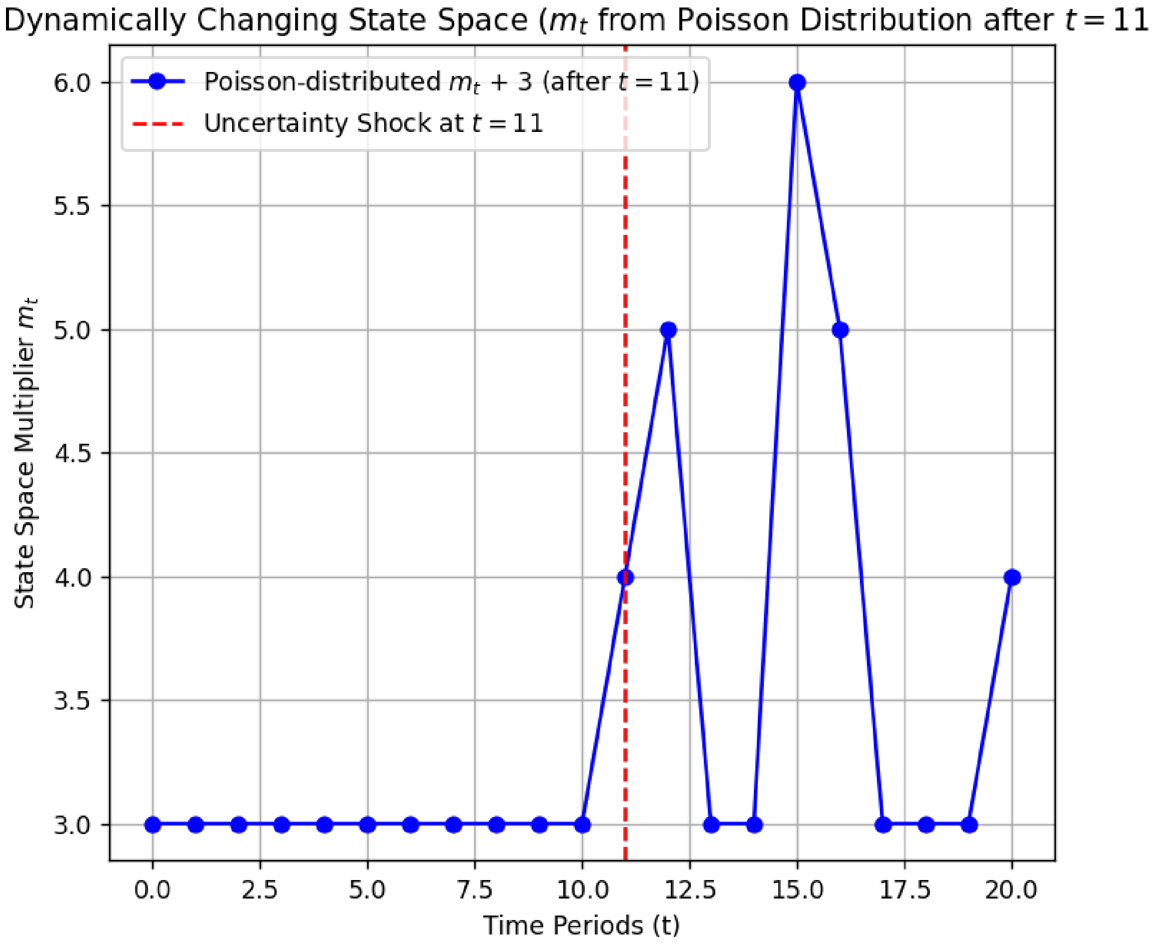
QUESTION 2.2

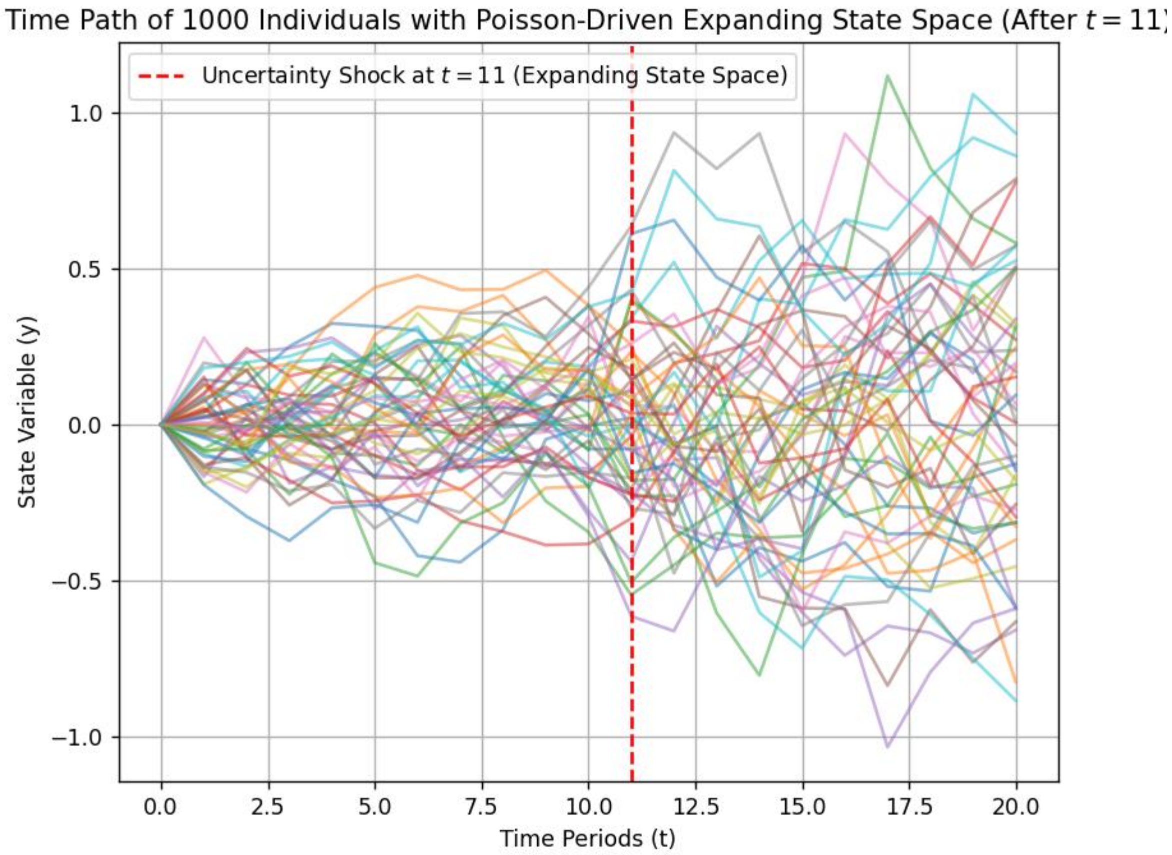
Simulate 1000 individuals indexed by i for 21 periods $t = 0, 1, \ldots, 20$, starting with $y_{i,0} = 0$ that receive an uncertainty shock only at time $t = 11$. Plot the time path of $y_{i,t}$ of all 1000 individuals over t.



QUESTION 2.3

```
m_t[11:] = np.random.poisson(1, num_periods - 11) + 3  # Poisson distribution + 3 after t=11
```





QUESTION 3.1

$$w_{t+1} = (1 + r_{t+1})s(w_t) + y_{t+1}$$

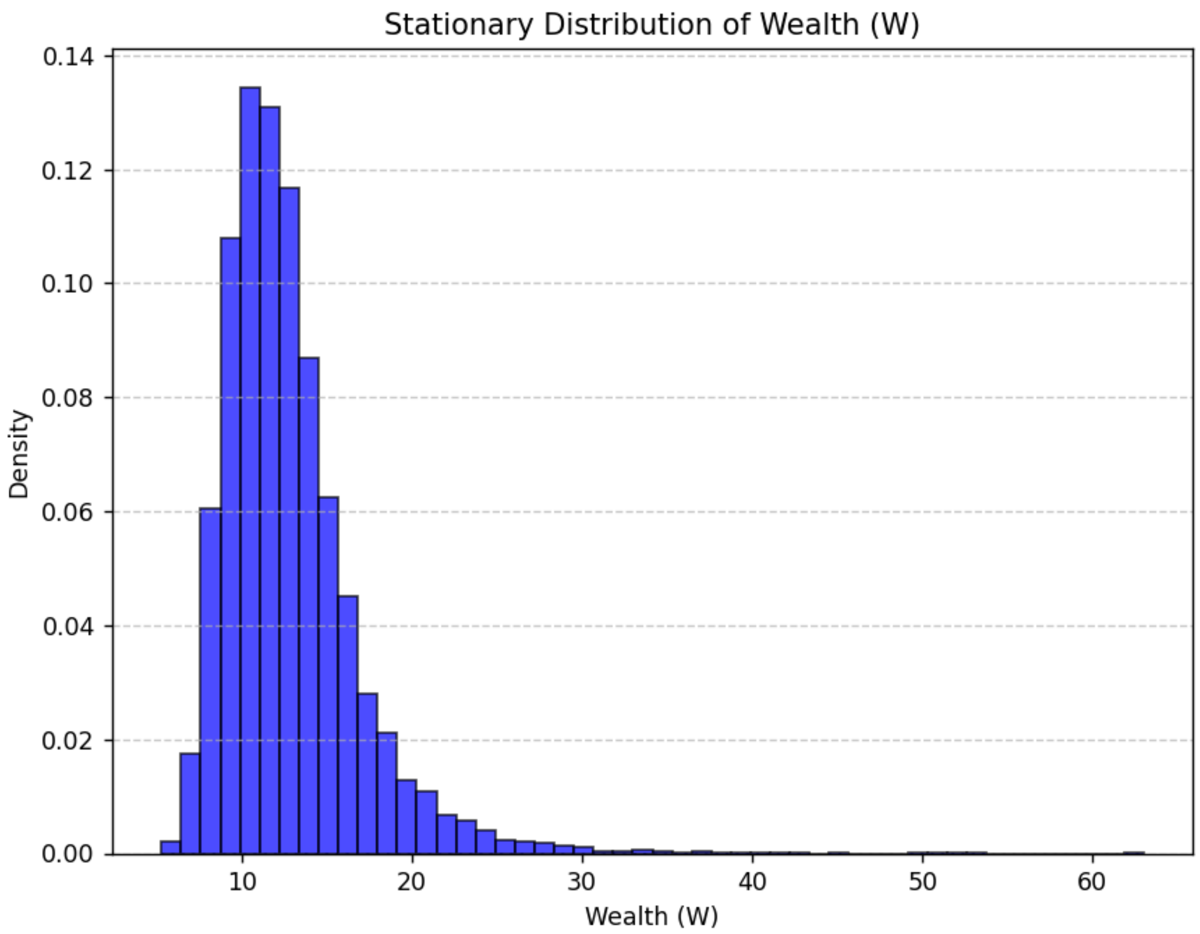
$$z_{t+1} = az_t + b + \sigma_z \epsilon_{t+1}$$

$$1 + r_t = c_r \exp(z_t) + \exp(\mu_r + \sigma_r \xi_t)$$

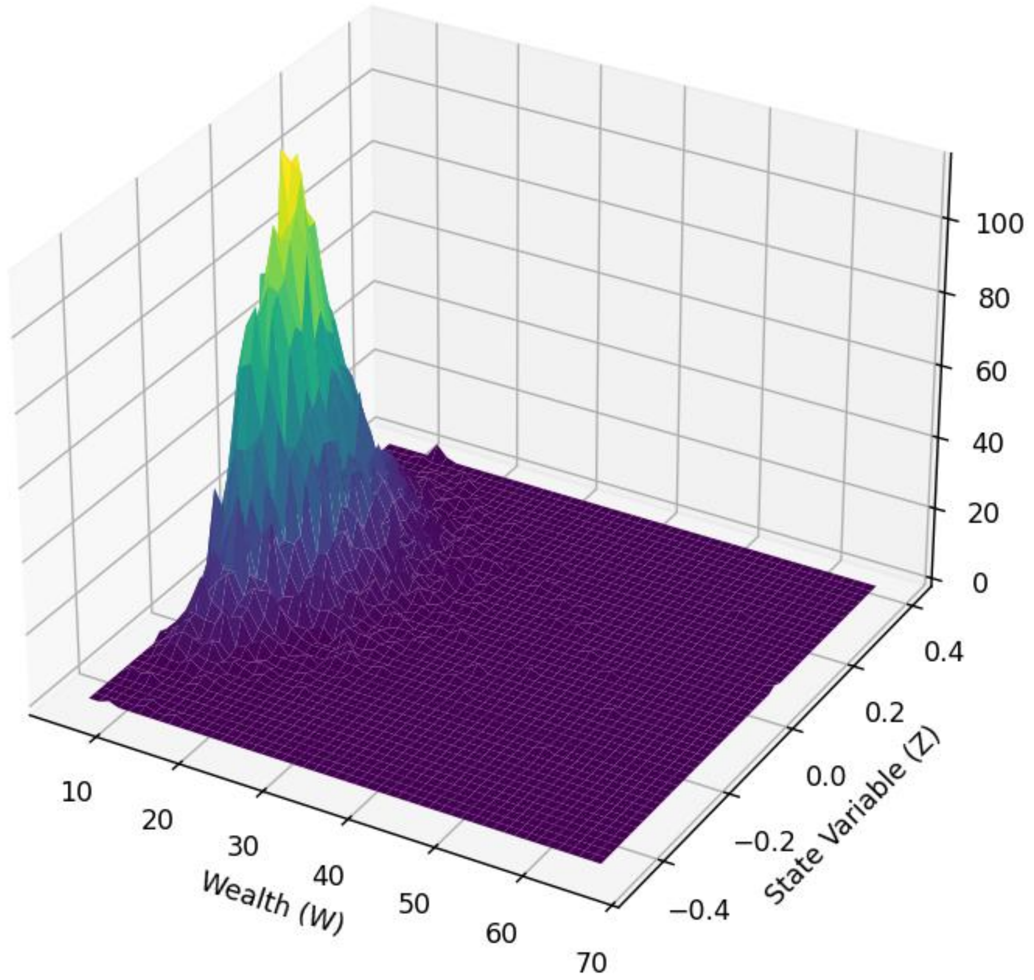
$$y_t = c_y \exp(z_t) + \exp(\mu_y + \sigma_y \zeta_t)$$

$$s(w) = s_0 w \cdot \mathbf{1}\{w \geq \hat{w}\}$$

```
# Define simulation parameters
N = 10000  # Number of individuals
T = 1000    # Number of time steps for convergence
w_0 = np.random.uniform(1.0, 10.0, N)  # Initial wealth values randomly distributed
```



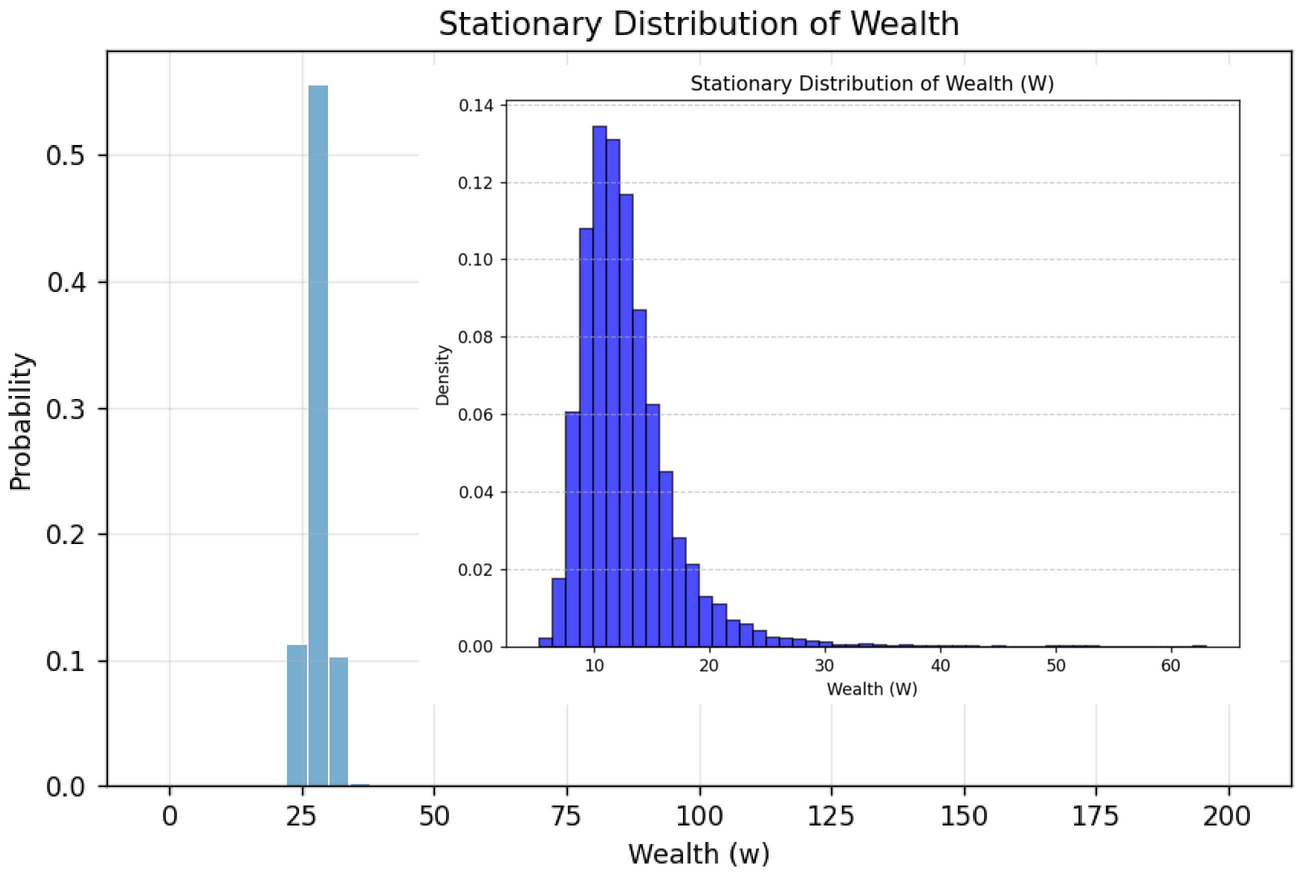
QUESTION 3.2& 3.3

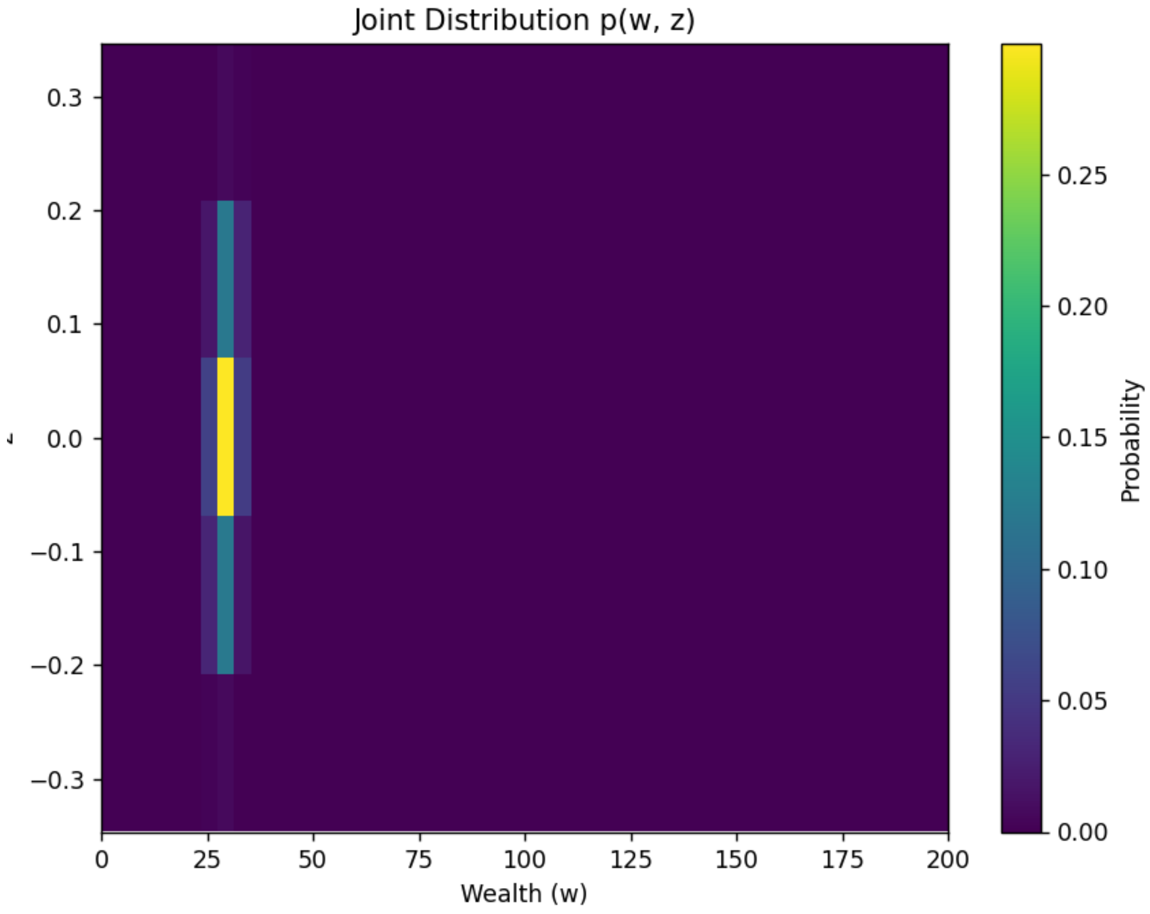


Mean of z: -0.000681
Variance of z: 0.0133
Mean of w: 12.8
Variance of w: 16.9
Correlation(z, w): 0.0665

QUESTION 4.1

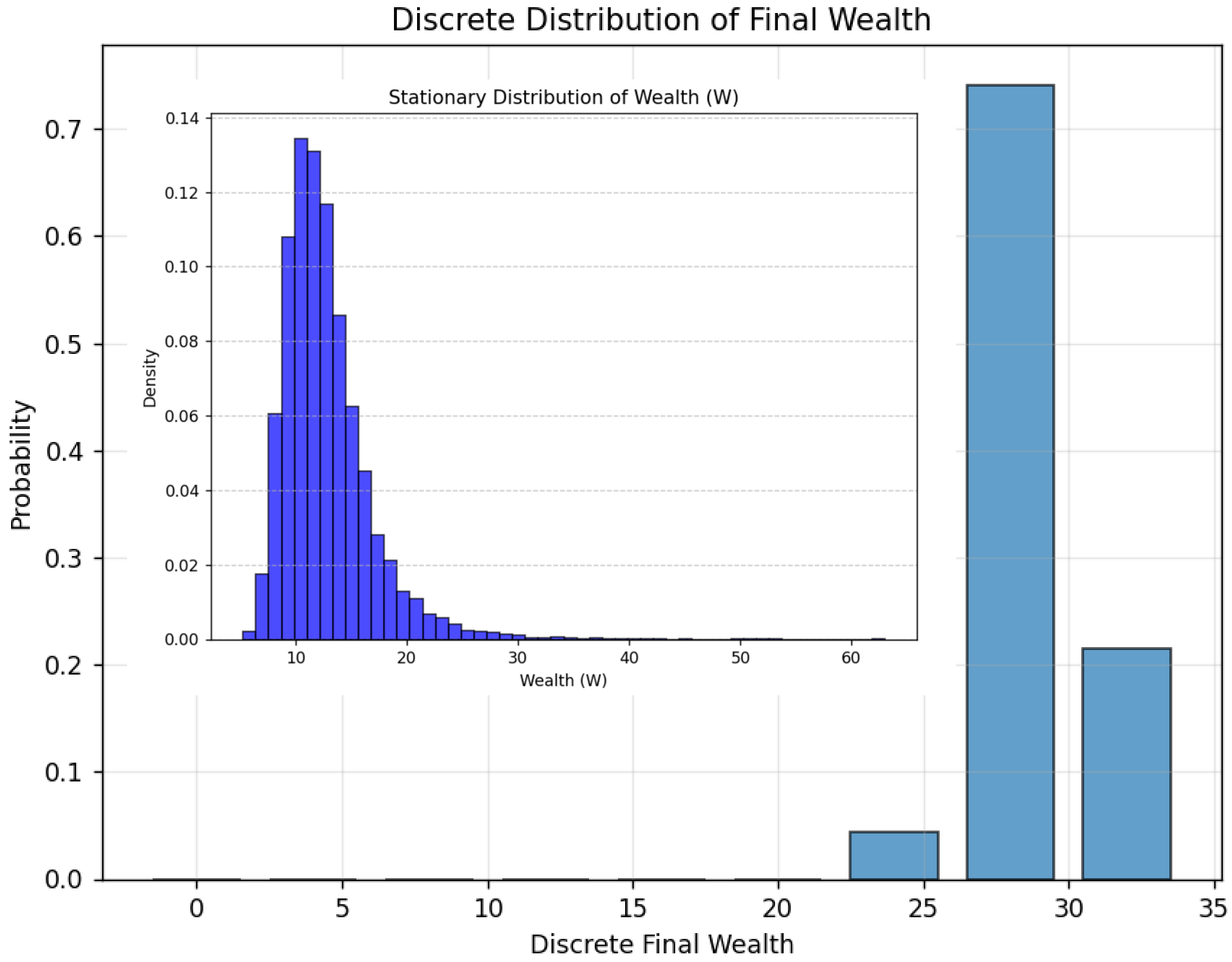
Compare the stochastic simulation using discrete grids to the original stochastic simulation in Task 3. Plot the stationary distribution in one- and two-dimension.

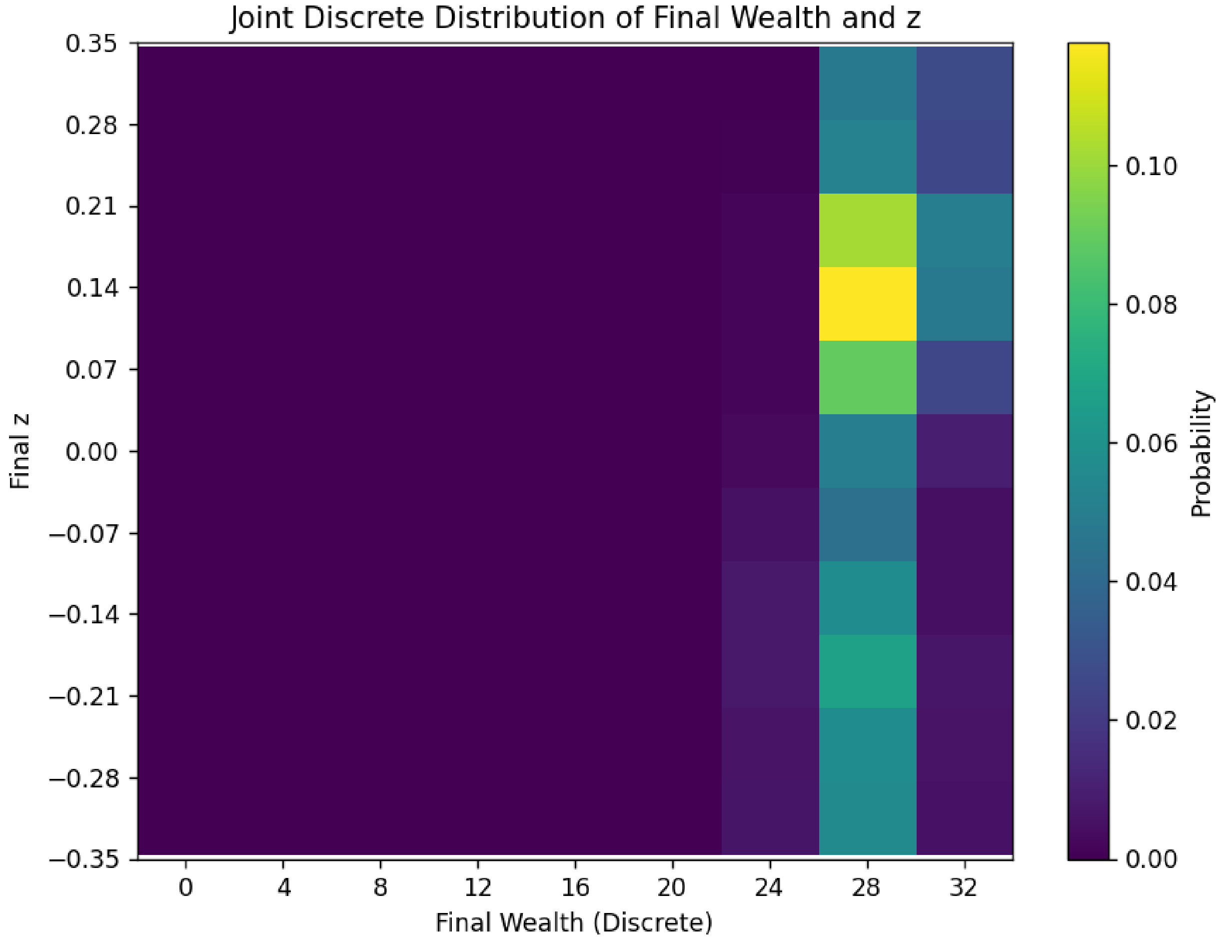




QUESTION 4.2

Compare the non-stochastic simulation to the stochastic simulation using discrete grids. Plot the stationary distribution in one- and two-dimension.





QUESTION 4.3

1.3.2: 43.5s
1.4.1: 4.83s
1.4.2: 10.6s

```python
def run_and_time(script_path):
    """
    Call an external Python script and count its running time (in seconds).
    Returns the running time of the script.
    """
    start = time.time()
    # If necessary, you can change it to "python3" or an absolute path
    subprocess.run(["python", script_path], check=True)
    end = time.time()

    return end - start
```

THANK YOU