ECON 6140

Problem Set 7

Gabe Sekeres

April 21, 2025

Authors: Gabe Sekeres and Finn Ye

NetIDs: gs754 and fy55

n.b. All code is below, in the code section.

1. First, note that there is a production subsidy in place of $\tau = \frac{1}{\varepsilon}$ such that

$$-\frac{U_{n,t}}{U_{c,t}} = \frac{W_t}{P_t} = \frac{W_t}{\mathcal{M}\frac{W_t(1-\tau)}{MPN_t}} = MPN_t$$

Following the notes and substituting in the respective rules into the dynamic IS equation, we get that

$$\tilde{y}_t = \mathbb{E}_t\{\tilde{y}_{t+1}\} - \frac{1}{\sigma}(\rho + \phi_\pi \pi_t + \phi_y \tilde{y}_t - \mathbb{E}_t\{\pi_{t+1}\} - \rho + \sigma(1-\rho_a)\psi_{ya}a_t - (1-\rho_z)z_t)$$

We will conjecture that our two variables of interest take the form

$$\pi_t = \psi_{\pi a}a_t + \psi_{\pi z}z_t$$
$$\tilde{y}_t = \psi_{ya}a_t + \psi_{yz}z_t$$

So under rational expectations, since both a_t and z_t are AR(1) processes and $\mathbb{E}[\varepsilon_t^a] = \mathbb{E}[\varepsilon_t^z] = 0$, we have that

$$\mathbb{E}\{\pi_{t+1}\} = \psi_{\pi a}\rho_a a_t + \psi_{\pi z}\rho_z z_t$$
$$\mathbb{E}\{\tilde{y}_{t+1}\} = \psi_{ya}\rho_a a_t + \psi_{yz}\rho_z z_t$$

Substituting the conjectured solution into our two equations of interest, we get

$$\begin{aligned}
\psi_{\pi a}a_t + \psi_{\pi z}z_t &= \beta\left(\psi_{\pi a}\rho_a a_t + \psi_{\pi z}\rho_z z_t\right) + \kappa\left(\psi_{ya}a_t + \psi_{yz}z_t\right) \\
\psi_{ya}a_t + \psi_{yz}z_t &= \psi_{ya}\rho_a a_t + \psi_{yz}\rho_z z_t - \frac{1}{\sigma}(\phi_\pi(\psi_{\pi a}a_t + \psi_{\pi z}z_t) + \phi_y(\psi_{ya}a_t + \psi_{yz}z_t) \\
&\quad - (\psi_{\pi a}\rho_a a_t + \psi_{\pi z}\rho_z z_t) + \sigma(1-\rho_a)\psi_{ya}a_t - (1-\rho_z)z_t)
\end{aligned}$$

Isolating the shocks, we have that

$$a_t\left(\psi_{\pi a} - \psi_{\pi a}\rho_a - \kappa\psi_{ya}\right) + z_t\left(\psi_{\pi z} - \psi_{\pi z}\rho_z - \kappa\psi_{yz}\right) = 0$$
$$a_t\left(\psi_{ya} - \psi_{ya}\rho_a + \frac{\psi_{\pi a}\phi_\pi + \psi_{ya}\phi_y - \psi_{\pi a}\rho_a}{\sigma} + (1-\rho_a)\psi_{ya}\right) +$$
$$z_t\left(\psi_{yz} - \psi_{yz}\rho_z + \frac{\psi_{\pi z}\phi_\pi + \psi_{yz}\phi_y - \psi_{\pi z}\rho_z - (1-\rho_z)}{\sigma}\right) = 0$$

Since we need these equations to hold for each possible shock in steady state, we will fix everything inside the parentheses to be zero. This will give is four linear equations in our four unknowns. We have:

$$(1-\rho_a)\psi_{\pi a} - \kappa\psi_{ya} = 0$$
$$(1-\rho_z)\psi_{\pi z} - \kappa\psi_{yz} = 0$$
$$\frac{\phi_\pi - \rho_a}{\sigma}\psi_{\pi a} + \left(2(1-\rho_a) + \frac{\phi_y}{\sigma}\right)\psi_{ya} = 0$$
$$\frac{\phi_\pi - \rho_z}{\sigma}\psi_{\pi z} + \left(1-\rho_z + \frac{\phi_y}{\sigma}\right)\psi_{yz} = \frac{1-\rho_z}{\sigma}$$

Putting everything in matrix form, this becomes

$$\underbrace{\begin{bmatrix} 1-\rho_a & -\kappa & 0 & 0 \\ 0 & 0 & 1-\rho_z & -\kappa \\ \frac{\phi_\pi-\rho_a}{\sigma} & 2-2\rho_a+\frac{\phi_y}{\sigma} & 0 & 0 \\ 0 & 0 & \frac{\phi_\pi-\rho_z}{\sigma} & 1-\rho_z+\frac{\phi_y}{\sigma} \end{bmatrix}}_{A} \cdot \begin{bmatrix} \psi_{\pi a} \\ \psi_{ya} \\ \psi_{\pi z} \\ \psi_{yz} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1-\rho_z}{\sigma} \end{bmatrix}$$

So we have that

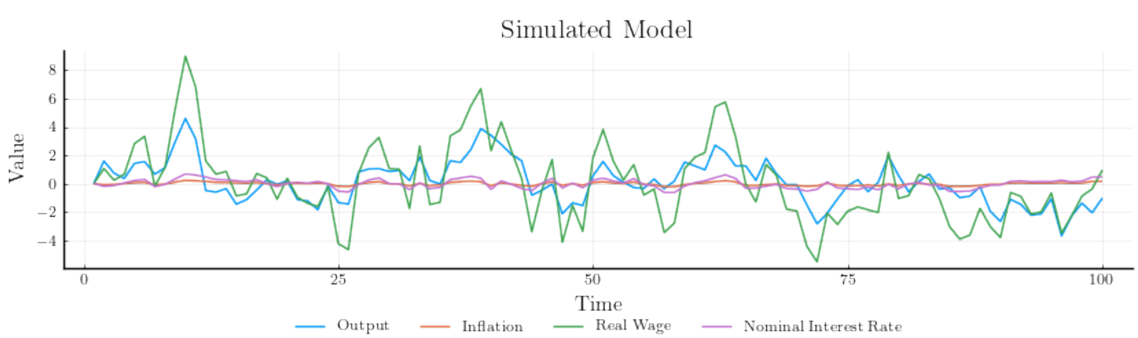
$$\begin{bmatrix} \psi_{\pi a} \\ \psi_{ya} \\ \psi_{\pi z} \\ \psi_{yz} \end{bmatrix} = A^{-1} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1-\rho_z}{\sigma} \end{bmatrix}$$

Putting everything together in Julia, and adding the actual values of the coefficients, recalling from the notes that $\kappa = \left(\sigma + \frac{\varphi+\alpha}{1-\alpha}\right)\frac{(1-\theta)(1-\beta\theta)}{\theta}\frac{1-\alpha}{1-\alpha+\alpha\varepsilon}$, we get that:

$$\begin{bmatrix} \psi_{\pi a} \\ \psi_{ya} \\ \psi_{\pi z} \\ \psi_{yz} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0.09823 \\ 0.26785 \end{bmatrix}$$

The first two being zero is strange compared to what we had in class, but relates to the fact that the natural rate of interest is itself a function of ψ_{ya}. This means that the only way we can hold in steady state is if neither inflation nor output respond to the log productivity shocks.

2. I simulated the model over 100 periods, and got the following time series data:

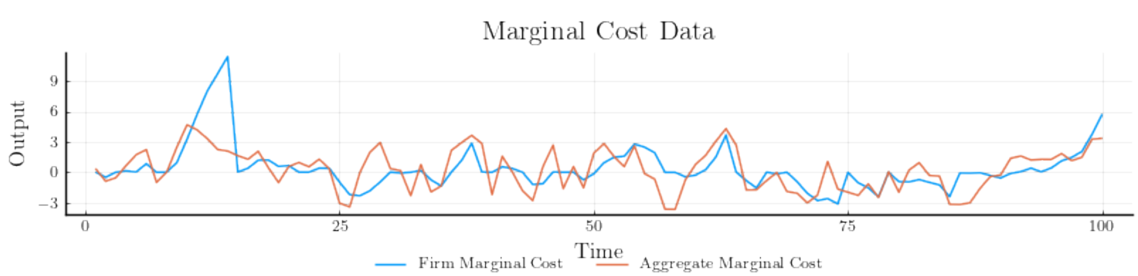


3. I simulated a single firm over 100 periods, only letting them change their price $(1-\theta)$ of the time (using a uniform random variable over the unit interval), and the firm's price was in the following figure:

The firm changed its price 28 times, and the average price duration was 2.3537 periods. This is similar to what we would expect from θ, of changing the price 25 times with an average duration of 3 periods. Plotting the price of firm j with respect to the aggregate price gives us

Plotting the output of the firm with respect to the aggregate output gives us

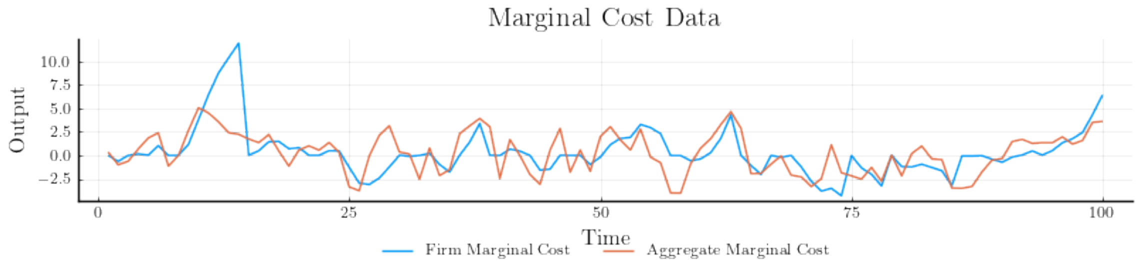
Plotting the marginal cost gives us



We conclude that the output and marginal cost are very correlated, anda are inversely correlated with the price levels.

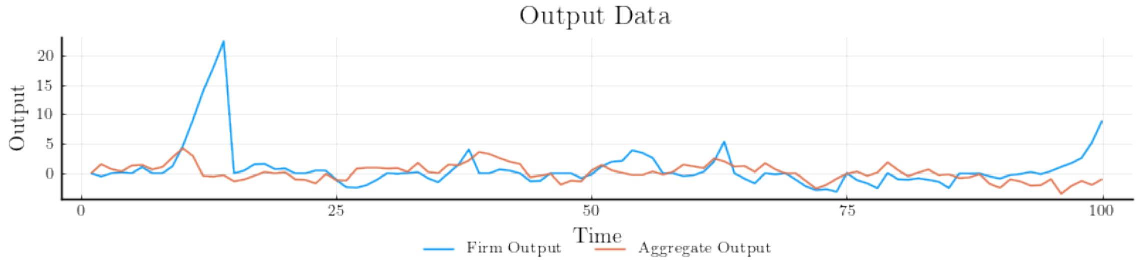
4. I redid the earlier analysis for the three changed parameterizations:

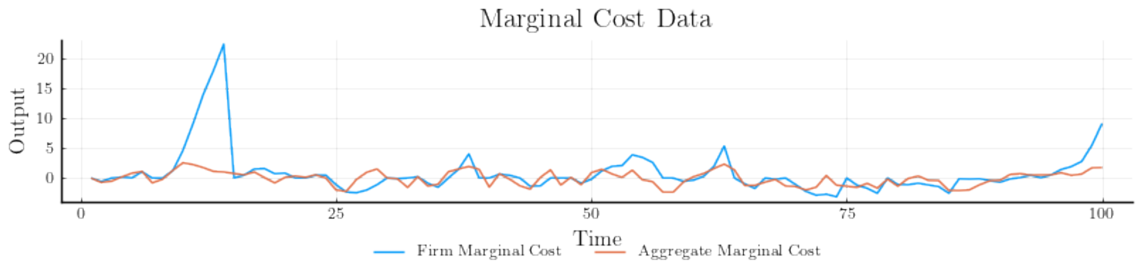
   (a) We set $\varepsilon = 10$. The plots we generated are:



We found that the variance of inflation was lower, the variance of output was slightly higher but basically the same, and that the variance of marginal cost increased substantially. We conclude that increasing ε decreases the change in inflation decreases but that the marginal cost gets rougher period-to-period.

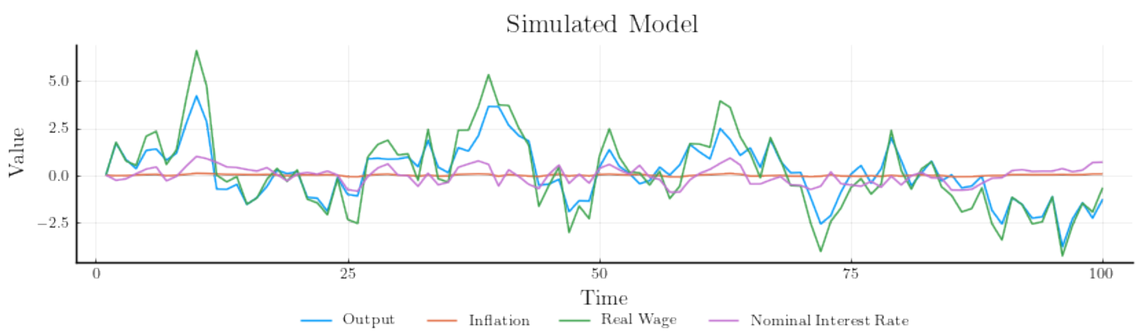
(b) We set $\alpha = 0$. The plots we generated are:

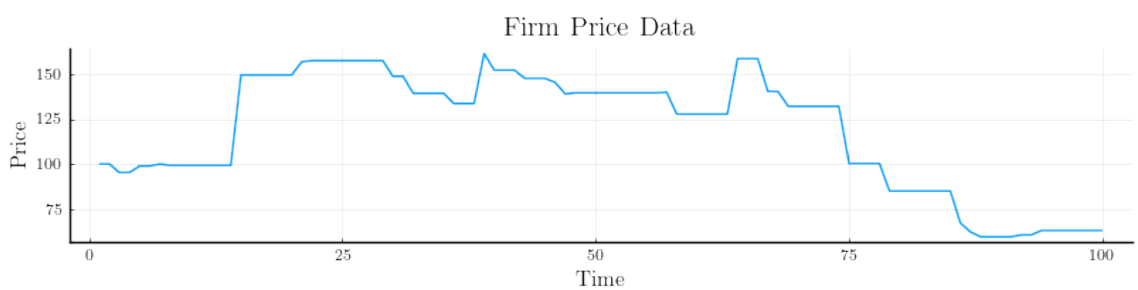


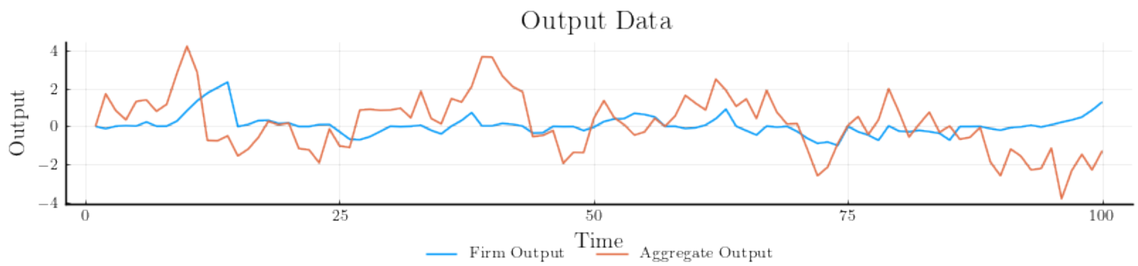


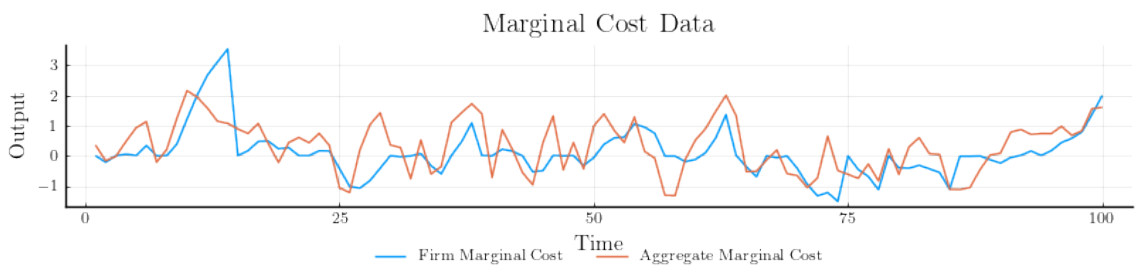
We found that the variance of everything except for output increases, and marginal cost increased massively in variance. We can see from the plots that there is a big spike in each and then a decrease back to relatively stable levels.

(c) We set $\phi_\pi = 10$ and $\phi_y = 0$. The plots we generated are:







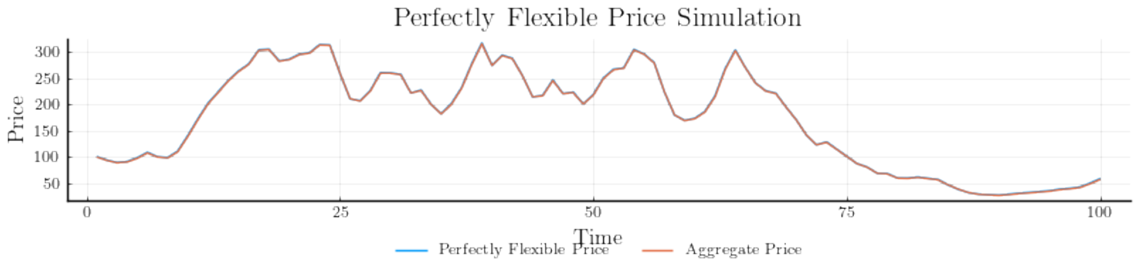


We found that the variance of inflation and marginal cost both massively decreased, but that output variance did not change that much at all. This can be seen in the plots, where it's quite spiky.

5. We calculated the price path using the expression

$$p_t^\star = \frac{1-\alpha}{1-\alpha+\alpha\varepsilon}\left(\sigma + \frac{\varphi+\alpha}{1-\alpha}\tilde{y}_t\right) + \pi_t$$

We got that the perfectly flexible firm would charge the same as exactly the aggregate price levels. This seems wrong, since the question seems to be guiding us towards a "no" answer, but we don't know what's going on here. The prices are:


Perfectly Flexible Price Simulation
Price
300
250
200
150
100
50
0
25
50
75
100
Time
Perfectly Flexible Price
Aggregate Price

Code Section

The code was:

```
mutable struct Parameters
    sigma::Float64
    varphi::Float64
    epsilon::Float64
    alpha::Float64
    beta::Float64
    theta::Float64
    phipi::Float64
    phiy::Float64
    rhoa::Float64
    sigmaa::Float64
    rhoz::Float64
    sigmaz::Float64
    kappa::Float64

    function Parameters(;sigma=2.0, varphi=3.0, epsilon=5.0, alpha=0.3, beta
       =0.99, theta=0.75, phipi=1.5, phiy=0.5, rhoa=0.8, sigmaa=1.0, rhoz
       =0.5, sigmaz=1.0)
        kappa = (sigma + (varphi + alpha) / (1 - alpha)) * (((1 - theta) * (1
           - beta * theta)) / theta) * ((1 - alpha) / (1 - alpha + alpha *
           epsilon))
        return new(sigma, varphi, epsilon, alpha, beta, theta, phipi, phiy,
           rhoa, sigmaa, rhoz, sigmaz, kappa)
    end
end

"""
    undet_coef(p::Parameters) -> Vector{Float64}

Solve for the psis using the method of undetermined coefficients.
"""
function undet_coef(p::Parameters)
    A = [(1 - p.rhoa) (-p.kappa) 0 0;
         0 0 (1-p.rhoz) (-p.kappa);
         ((p.phipi - p.rhoa) / p.sigma) (2 - 2 * p.rhoa + p.phiy / p.sigma) 0
             0;
         0 0 ((p.phipi - p.rhoz) / p.sigma) (1 - p.rhoz + p.phiy / p.sigma)]
    B = [0; 0; 0; (1 - p.rhoz) / p.sigma]
    return inv(A) * B
end

"""
    simulate(p::Parameters, T::Int) -> Vector{Float64}, Vector{Float64},
       Vector{Float64}, Vector{Float64}

Simulate the model for T periods in a certain parameterization, returning
   vectors of
    output, inflation, the real wage, and the nominal interest rate.
```

```
"""
function simulate(p::Parameters, T::Int; seed::Int=12345)
    Random.seed!(seed)
    # Initialize productivity and demand shocks
    a = zeros(T)
    z = zeros(T)
    for t in 2:T
        a[t] = p.rhoa * a[t-1] + randn() * sqrt(p.sigmaa)
        z[t] = p.rhoz * z[t-1] + randn() * sqrt(p.sigmaz)
    end

    psis = undet_coef(p)

    # Initialize output, inflation, real wage, and nominal interest rate
    y_tilde = zeros(T)
    y_n = zeros(T)
    y = zeros(T)
    pi = zeros(T)
    w = zeros(T)
    i = zeros(T)

    for t in 1:T
        y_tilde[t] = psis[2] * a[t] + psis[4] * z[t]
        y_n[t] = ((1 + p.varphi) / (p.sigma * (1 - p.alpha) + p.varphi + p.
           alpha)) * a[t]
        y[t] = y_tilde[t] + y_n[t]
        pi[t] = psis[1] * a[t] + psis[3] * z[t]
        w[t] = p.sigma * y[t] + p.varphi * ((y[t] - a[t]) / (1 - p.alpha))
        i[t] = -log(p.beta) + p.phipi * pi[t] + p.phiy * y_tilde[t]
    end

    return y, pi, w, i, a, z, y_tilde, y_n
end

using Random, Distributions, Plots
include("functions.jl")
# Make the plots look pretty
pyplot()
PyPlot.rc("text", usetex=true)
PyPlot.rc("font", family="serif")
PyPlot.matplotlib.rcParams["mathtext.fontset"] = "cm"
# Initialize parameters
p = Parameters()

# 1: Solve using method of undetermined coefficients
psis = undet_coef(p)
println("psi_pi_a: $(psis[1])")
println("psi_y_a: $(psis[2])")
println("psi_pi_z: $(psis[3])")
println("psi_y_z: $(psis[4])")

# 2: Simulate the model
```

```
T = 100
yInit, piInit, wInit, iInit, aInit, zInit, y_tildeInit, y_nInit = simulate(p,
    T)

plt2 = plot(background=:transparent, legend=:outerbottom, legendcolumns=4,
    xlabel="Time", ylabel="Value", title="Simulated Model", linewidth=2,
    linealpha=0.7, size=(825,250))
plot!(plt2, yInit, label="Output")
plot!(plt2, piInit, label="Inflation")
plot!(plt2, wInit, label="Real Wage")
plot!(plt2, iInit, label="Nominal Interest Rate")

savefig(plt2, "macro_hw7_code/2_simulated_model.png")

# 3: Follow a single firm j through the simulation
price_data = zeros(T)
price_data[1] = 100
for t in 2:T
    price_data[t] = price_data[t-1] * exp(piInit[t])
end

price_firm = zeros(T)
price_firm[1] = 100
for t in 2:T
    if rand() < p.theta # Firm cannot change price
        price_firm[t] = price_firm[t-1]
    else # Firm can change price
        price_firm[t] = price_data[t]
    end
end
output_firm = (yInit .+ (-p.epsilon) * (price_firm - price_data)) ./
    price_firm
mc_aggregate = ((p.sigma + (p.varphi + p.alpha) / (1 - p.alpha))) .* yInit .-
    log(1 - p.alpha) .- (1 + p.varphi) / (1 - p.alpha) .* aInit
mc_firm = (mc_aggregate .- (p.alpha + p.epsilon) / (1 - p.alpha) * (price_firm
     - price_data)) ./ price_firm


plt3a = plot(background=:transparent, legend=false, xlabel="Time", ylabel="
    Price", title="Firm Price Data", linewidth=2, linealpha=0.7, size
    =(800,200))
plot!(plt3a, price_firm, label="Firm Price")

savefig(plt3a, "macro_hw7_code/3a_firm_price_data.png")

plt3b = plot(background=:transparent, legend=false, xlabel="Time", ylabel="
    Price", title="Price Data", linewidth=2, linealpha=0.7, size=(800,200))
plot!(plt3b, price_firm, label="Firm Price")
plot!(plt3b, price_data, label="Aggregate Price")

savefig(plt3b, "macro_hw7_code/3b_price_data.png")
```

```
plt3c = plot(background=:transparent, legend=:outerbottom, legendcolumns=2,
   xlabel="Time", ylabel="Output", title="Output Data", linewidth=2,
   linealpha=0.7, size=(800,200))
plot!(plt3c, output_firm, label="Firm Output")
plot!(plt3c, yInit, label="Aggregate Output")

savefig(plt3c, "macro_hw7_code/3c_output_data.png")

plt3d = plot(background=:transparent, legend=:outerbottom, legendcolumns=2,
   xlabel="Time", ylabel="Output", title="Marginal Cost Data", linewidth=2,
   linealpha=0.7, size=(800,200))
plot!(plt3d, mc_firm, label="Firm Marginal Cost")
plot!(plt3d, mc_aggregate, label="Aggregate Marginal Cost")

savefig(plt3d, "macro_hw7_code/3d_marginal_cost_data.png")


# 4a: Simulate the model with epsilon = 10
p_epsilon = Parameters(epsilon=10.0)
psis_epsilon = undet_coef(p_epsilon)

y_epsilon, pi_epsilon, w_epsilon, i_epsilon, a_epsilon, z_epsilon = simulate(
   p_epsilon, T)
plt4a2 = plot(background=:transparent, legend=:outerbottom, legendcolumns=4,
   xlabel="Time", ylabel="Value", title="Simulated Model", linewidth=2,
   linealpha=0.7, size=(825,250))
plot!(plt4a2, y_epsilon, label="Output")
plot!(plt4a2, pi_epsilon, label="Inflation")
plot!(plt4a2, w_epsilon, label="Real Wage")
plot!(plt4a2, i_epsilon, label="Nominal Interest Rate")

savefig(plt4a2, "macro_hw7_code/4a2_simulated_model.png")

price_data_epsilon = zeros(T)
price_data_epsilon[1] = 100
for t in 2:T
    price_data_epsilon[t] = price_data_epsilon[t-1] * exp(pi_epsilon[t])
end

price_firm_epsilon = zeros(T)
price_firm_epsilon[1] = 100
for t in 2:T
    if rand() < p_epsilon.theta # Firm cannot change price
        price_firm_epsilon[t] = price_firm_epsilon[t-1]
    else # Firm can change price
        price_firm_epsilon[t] = price_data_epsilon[t]
    end
end
output_firm_epsilon = (y_epsilon .+ (-p_epsilon.epsilon) * (price_firm_epsilon
     - price_data_epsilon)) ./ price_firm_epsilon
```

```
mc_aggregate_epsilon = ((p_epsilon.sigma + (p_epsilon.varphi + p_epsilon.alpha
    ) / (1 - p_epsilon.alpha))) .* y_epsilon .- log(1 - p_epsilon.alpha) .- (1
     + p_epsilon.varphi) / (1 - p_epsilon.alpha) .* a_epsilon
mc_firm_epsilon = (mc_aggregate_epsilon .- (p_epsilon.alpha + p_epsilon.
    epsilon) / (1 - p_epsilon.alpha) * (price_firm_epsilon -
    price_data_epsilon)) ./ price_firm_epsilon


plt4a3a = plot(background=:transparent, legend=false, xlabel="Time", ylabel="
    Price", title="Firm Price Data", linewidth=2, linealpha=0.7, size
    =(800,200))
plot!(plt4a3a, price_firm_epsilon, label="Firm Price")

savefig(plt4a3a, "macro_hw7_code/4a3a_firm_price_data.png")

plt4a3b = plot(background=:transparent, legend=false, xlabel="Time", ylabel="
    Price", title="Price Data", linewidth=2, linealpha=0.7, size=(800,200))
plot!(plt4a3b, price_firm_epsilon, label="Firm Price")
plot!(plt4a3b, price_data_epsilon, label="Aggregate Price")

savefig(plt4a3b, "macro_hw7_code/4a3b_price_data.png")

plt4a3c = plot(background=:transparent, legend=:outerbottom, legendcolumns=2,
    xlabel="Time", ylabel="Output", title="Output Data", linewidth=2,
    linealpha=0.7, size=(800,200))
plot!(plt4a3c, output_firm_epsilon, label="Firm Output")
plot!(plt4a3c, y_epsilon, label="Aggregate Output")

savefig(plt4a3c, "macro_hw7_code/4a3c_output_data.png")

plt4a3d = plot(background=:transparent, legend=:outerbottom, legendcolumns=2,
    xlabel="Time", ylabel="Output", title="Marginal Cost Data", linewidth=2,
    linealpha=0.7, size=(800,200))
plot!(plt4a3d, mc_firm_epsilon, label="Firm Marginal Cost")
plot!(plt4a3d, mc_aggregate_epsilon, label="Aggregate Marginal Cost")

savefig(plt4a3d, "macro_hw7_code/4a3d_marginal_cost_data.png")

println("With epsilon = 10:")
println("Variance of inflation: $(var(pi_epsilon)), old variance: $(var(piInit
    ))")
println("Variance of output: $(var(y_epsilon)), old variance: $(var(yInit))")
println("Variance of marginal cost: $(var(mc_firm_epsilon)), old variance: $(
    var(mc_firm))")

# 4b: Simulate the model with alpha = 0
p_alpha = Parameters(alpha=0.0)
psis_alpha = undet_coef(p_alpha)

y_alpha, pi_alpha, w_alpha, i_alpha, a_alpha, z_alpha = simulate(p_alpha, T)
plt4b2 = plot(background=:transparent, legend=:outerbottom, legendcolumns=4,
```

```
    xlabel="Time", ylabel="Value", title="Simulated Model", linewidth=2,
    linealpha=0.7, size=(825,250))
plot!(plt4b2, y_alpha, label="Output")
plot!(plt4b2, pi_alpha, label="Inflation")
plot!(plt4b2, w_alpha, label="Real Wage")
plot!(plt4b2, i_alpha, label="Nominal Interest Rate")

savefig(plt4b2, "macro_hw7_code/4b2_simulated_model.png")

price_data_alpha = zeros(T)
price_data_alpha[1] = 100
for t in 2:T
    price_data_alpha[t] = price_data_alpha[t-1] * exp(pi_alpha[t])
end

price_firm_alpha = zeros(T)
price_firm_alpha[1] = 100
for t in 2:T
    if rand() < p_alpha.theta # Firm cannot change price
        price_firm_alpha[t] = price_firm_alpha[t-1]
    else # Firm can change price
        price_firm_alpha[t] = price_data_alpha[t]
    end
end
output_firm_alpha = (y_alpha .+ (-p_alpha.epsilon) * (price_firm_alpha -
    price_data_alpha)) ./ price_firm_alpha
mc_aggregate_alpha = ((p_alpha.sigma + (p_alpha.varphi + p_alpha.alpha) / (1 -
     p_alpha.alpha))) .* y_alpha .- log(1 - p_alpha.alpha) .- (1 + p_alpha.
    varphi) / (1 - p_alpha.alpha) .* a_alpha
mc_firm_alpha = (mc_aggregate_alpha .- (p_alpha.alpha + p_alpha.epsilon) / (1
    - p_alpha.alpha) * (price_firm_alpha - price_data_alpha)) ./
    price_firm_alpha


plt4b3a = plot(background=:transparent, legend=false, xlabel="Time", ylabel="
    Price", title="Firm Price Data", linewidth=2, linealpha=0.7, size
    =(800,200))
plot!(plt4b3a, price_firm_alpha, label="Firm Price")

savefig(plt4b3a, "macro_hw7_code/4b3a_firm_price_data.png")

plt4b3b = plot(background=:transparent, legend=false, xlabel="Time", ylabel="
    Price", title="Price Data", linewidth=2, linealpha=0.7, size=(800,200))
plot!(plt4b3b, price_firm_alpha, label="Firm Price")
plot!(plt4b3b, price_data_alpha, label="Aggregate Price")

savefig(plt4b3b, "macro_hw7_code/4b3b_price_data.png")

plt4b3c = plot(background=:transparent, legend=:outerbottom, legendcolumns=2,
    xlabel="Time", ylabel="Output", title="Output Data", linewidth=2,
    linealpha=0.7, size=(800,200))
```

```
plot!(plt4b3c, output_firm_alpha, label="Firm Output")
plot!(plt4b3c, y_alpha, label="Aggregate Output")

savefig(plt4b3c, "macro_hw7_code/4b3c_output_data.png")

plt4b3d = plot(background=:transparent, legend=:outerbottom, legendcolumns=2,
    xlabel="Time", ylabel="Output", title="Marginal Cost Data", linewidth=2,
    linealpha=0.7, size=(800,200))
plot!(plt4b3d, mc_firm_alpha, label="Firm Marginal Cost")
plot!(plt4b3d, mc_aggregate_alpha, label="Aggregate Marginal Cost")

savefig(plt4b3d, "macro_hw7_code/4b3d_marginal_cost_data.png")

println("With alpha = 0:")
println("Variance of inflation: $(var(pi_alpha)), old variance: $(var(piInit))
    ")
println("Variance of output: $(var(y_alpha)), old variance: $(var(yInit))")
println("Variance of marginal cost: $(var(mc_firm_alpha)), old variance: $(var
    (mc_firm))")

# 4c: Simulate the model with phipi = 10 and phiy = 0
p_phipi = Parameters(phipi=10.0, phiy=0.0)
psis_phipi = undet_coef(p_phipi)

y_phipi, pi_phipi, w_phipi, i_phipi, a_phipi, z_phipi = simulate(p_phipi, T)
plt4c2 = plot(background=:transparent, legend=:outerbottom, legendcolumns=4,
    xlabel="Time", ylabel="Value", title="Simulated Model", linewidth=2,
    linealpha=0.7, size=(825,250))
plot!(plt4c2, y_phipi, label="Output")
plot!(plt4c2, pi_phipi, label="Inflation")
plot!(plt4c2, w_phipi, label="Real Wage")
plot!(plt4c2, i_phipi, label="Nominal Interest Rate")

savefig(plt4c2, "macro_hw7_code/4c2_simulated_model.png")

price_data_phipi = zeros(T)
price_data_phipi[1] = 100
for t in 2:T
    price_data_phipi[t] = price_data_phipi[t-1] * exp(pi_phipi[t])
end

price_firm_phipi = zeros(T)
price_firm_phipi[1] = 100
for t in 2:T
    if rand() < p_phipi.theta # Firm cannot change price
        price_firm_phipi[t] = price_firm_phipi[t-1]
    else # Firm can change price
        price_firm_phipi[t] = price_data_phipi[t]
    end
end
output_firm_phipi = (y_phipi .+ (-p_phipi.epsilon) * (price_firm_phipi -
```

```
    price_data_phipi)) ./ price_firm_phipi
mc_aggregate_phipi = ((p_phipi.sigma + (p_phipi.varphi + p_phipi.alpha) / (1 -
     p_phipi.alpha))) .* y_phipi .- log(1 - p_phipi.alpha) .- (1 + p_phipi.
    varphi) / (1 - p_phipi.alpha) .* a_phipi
mc_firm_phipi = (mc_aggregate_phipi .- (p_phipi.alpha + p_phipi.epsilon) / (1
    - p_phipi.alpha) * (price_firm_phipi - price_data_phipi)) ./
    price_firm_phipi


plt4c3a = plot(background=:transparent, legend=false, xlabel="Time", ylabel="
    Price", title="Firm Price Data", linewidth=2, linealpha=0.7, size
    =(800,200))
plot!(plt4c3a, price_firm_phipi, label="Firm Price")

savefig(plt4c3a, "macro_hw7_code/4c3a_firm_price_data.png")

plt4c3b = plot(background=:transparent, legend=false, xlabel="Time", ylabel="
    Price", title="Price Data", linewidth=2, linealpha=0.7, size=(800,200))
plot!(plt4c3b, price_firm_phipi, label="Firm Price")
plot!(plt4c3b, price_data_phipi, label="Aggregate Price")

savefig(plt4c3b, "macro_hw7_code/4c3b_price_data.png")

plt4c3c = plot(background=:transparent, legend=:outerbottom, legendcolumns=2,
    xlabel="Time", ylabel="Output", title="Output Data", linewidth=2,
    linealpha=0.7, size=(800,200))
plot!(plt4c3c, output_firm_phipi, label="Firm Output")
plot!(plt4c3c, y_phipi, label="Aggregate Output")

savefig(plt4c3c, "macro_hw7_code/4c3c_output_data.png")

plt4c3d = plot(background=:transparent, legend=:outerbottom, legendcolumns=2,
    xlabel="Time", ylabel="Output", title="Marginal Cost Data", linewidth=2,
    linealpha=0.7, size=(800,200))
plot!(plt4c3d, mc_firm_phipi, label="Firm Marginal Cost")
plot!(plt4c3d, mc_aggregate_phipi, label="Aggregate Marginal Cost")

savefig(plt4c3d, "macro_hw7_code/4c3d_marginal_cost_data.png")

println("With phipi = 10 and phiy = 0:")
println("Variance of inflation: $(var(pi_phipi)), old variance: $(var(piInit))
    ")
println("Variance of output: $(var(y_phipi)), old variance: $(var(yInit))")
println("Variance of marginal cost: $(var(mc_firm_phipi)), old variance: $(var
    (mc_firm))")


# 5: Perfectly flexible price simulation
price_perfect = (1 - p.alpha) / (1 - p.alpha + p.alpha * p.epsilon) * (p.sigma
     .+ (p.varphi + p.alpha) / (1 - p.alpha) * (y_tildeInit)) .+ price_data
```

```
plt5 = plot(background=:transparent, legend=:outerbottom, legendcolumns=2,
    xlabel="Time", ylabel="Price", title="Perfectly Flexible Price Simulation"
    , linewidth=2, linealpha=0.7, size=(800,200))
plot!(plt5, price_perfect, label="Perfectly Flexible Price")
plot!(plt5, price_data, label="Aggregate Price")

savefig(plt5, "macro_hw7_code/5_perfectly_flexible_price_simulation.png")
```