ECO862 - International Trade
Lecture 8a: A primer on solving inventory models

Recap

Studied how inventory models can help understand

- effects of large devaluation
  - sharp quantity reduction
  - gradual price adjustment (markup movement)
- mismeasurement in trade elasticity
  - anticipatory trade reversal before policy change
  - laboratory to test correction applied to data

Inventory Models

- Inventory Management is an important part of International Trade
- The sensitivity of trade to various shocks depend on how inventories are held
- Understanding inventories is, thus, understanding a large part of international trade
- Certain features of trade facilitate inventory holding

Why hold Inventories?

- Holding inventories is basically negative-return investment due to depreciation
- Fixed cost of shipment: introduces non-convexities in costs creating action/inaction regions i.e. (s,S) policy
- Uncertain demand: holding buffer stocks
- Delivery Lags: inventories to meet the demand in transition of goods
- Production adjustment cost: inventories to meet fluctuations in demand when production is costly to adjust
  - Generates counterfactual business cycle implication that $\sigma_{prod} < \sigma_{sales}$, see Ramey and West (1999)
  - Khan and Thomas (2007) show that a macro model with (s,S) policies achieve $\sigma_{prod} > \sigma_{sales}$ as in the data

This lecture

- Take a look at analytical solution of inventory models (mappable to other discrete choice problems)
- Discuss the solution method of these types of models
- Look at transitions in the case of announced and unannounced shocks
- Consider model with trade policy uncertainty in the form of non-stationary tariff changes

Model

- Alessandria, Kaboski and Midrigan's (2010) inventory model
- Small Open economy with a continuum of monopolistically competitive importers that differ in i.i.d. demand shock and stock level
- Importers face fixed cost of importing, demand uncertainty and delivery lag
- This leads to a (s,S) ordering policy
- Solve for ergodic distribution of importer's inventory holdings

Model Description

- Let state of the world in period-t be η_t and history η^t
- CES Demand faced by the importer-j is

$$y_j(\eta^t) = e^{\nu_j(\eta^t)} p_j(\eta^t)^{-\sigma}$$

- constant-across-all-importers unit cost of imports ω (shocked later)
- Fixed ordering cost of f
- Delivery lags means

$$q_j(\eta^t) = min(e^{\nu_j(\eta^t)} p_j(\eta^t)^{-\sigma}, s_j(\eta^t))$$

  i.e. can't sell goods just ordered
- Goods in warehouses and in transit depreciate at the rate of δ

Importer's Problem

- Importer Decides between Importing or not importing

$$V(s,\nu) = \max[V^a(s,\nu), V^n(s,\nu)]$$

$$V^a(s,\nu) = \max_{p,z>0} q(p,s,\nu)p - \omega z - f + \beta EV(s',\nu')$$

$$V^n(s,\nu) = \max_{p>0} q(p,s,\nu)p + \beta EV(s',\nu')$$

Subject to

$$q(p,s,\nu) = min(e^{\nu} p^{-\sigma}, s)$$

$$s' = \begin{cases} (1-\delta)[s - q(p,s,\nu) + z] & \text{if import} \\ (1-\delta)[s - q(p,s,\nu)] & \text{o/w} \end{cases}$$

Optimal Policies

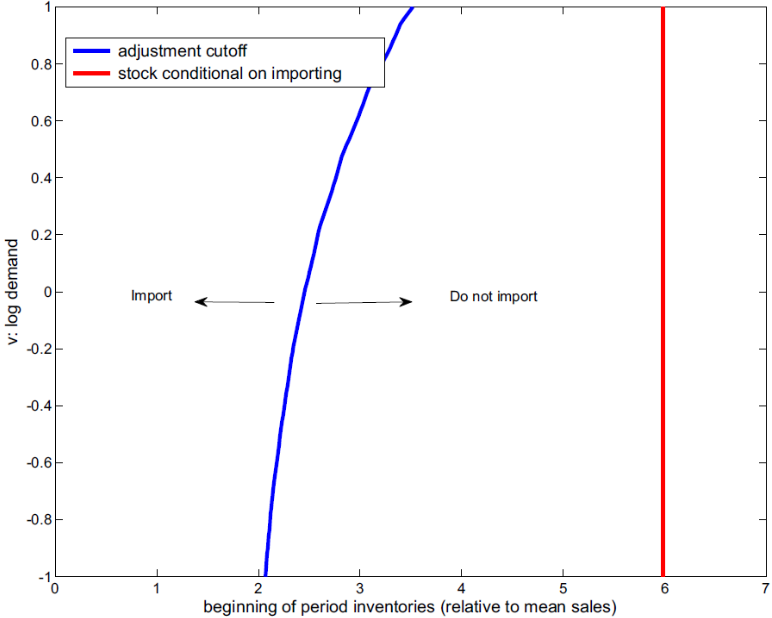
- Characterize pricing decision conditional on ordering $\{p^a(s,\nu), p^n(s,\nu)\}$, orders $i(s,\nu)$ and binary ordering decision $\phi(s,\nu)$
- Optimal import level comes from

$$\omega = \beta(1-\delta)EV_s(s',\nu')$$

- Firm's cutoff inventory level changes with the demand shock and the level of inventory
  - high demand $\Rightarrow$ more inventory depleted $\Rightarrow$ order sooner
  - low inventory $\Rightarrow$ higher chance of stocking-out $\Rightarrow$ order sooner

Ordering Rule

Figure 2: Optimal Import Rules



Pricing Rule

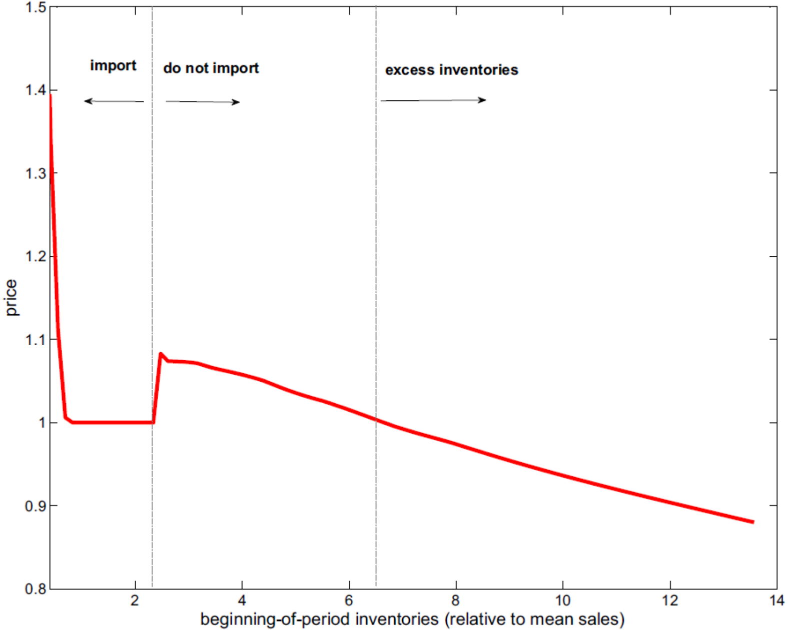
- CES demand structure: price would be a markup over marginal cost
- Here marginal cost is not purchase cost ω, but firm's marginal value of additional inventory

$$p = \frac{\sigma}{\sigma - 1} V_s(s, \nu)$$

- Marginal value depends on the stock level i.e. $\omega \gtreqless V_s(s, \nu)$
  - $V_s(s, \nu) = \omega$ when firm orders
  - $V_s(s, \nu) > \omega$ when firm does not order because it is saving fixed importing cost
  - $V_s(s, \nu) < \omega$ in case of excess inventories
- In case of stockout: $\nu p^{-\sigma} = s$

Pricing Rule

Figure 3: Optimal Price Functions



Analytical example

Assume, $f = 0$, $\nu = 0$ but 1 period lag

$$V(s) = \max_{q,z} (q^{\frac{\sigma-1}{\sigma}} - \omega z) + \beta V((1-\delta)(s - q + z))$$

s.t.:

$$q \leq s \,, \; z \geq 0$$

Note: Discrete-time makes characterizing a full solution a pain. Try continuous time?

Intuitively, what should happen when $f = 0$ and no uncertainty?

Analytical example — langrangian

$$V(s) = \max(q^{\frac{\sigma-1}{\sigma}} - \omega z) + \beta V((1-\delta)(s-q+z)) + \lambda(s-q) + \mu z$$

Notice:

- $\mu = 0$ when $z > 0$
- $\lambda = 0$ when $q < s$
- $\lambda \uparrow$ as inventories $\downarrow$

Analytical example — FOCs

$$V(s) = \max(q^{\frac{\sigma-1}{\sigma}} - \omega z) + \beta V((1-\delta)(s-q+z)) + \lambda(s-q) + \mu z$$

FOCs:

$$\begin{aligned}
q :& \frac{\sigma-1}{\sigma} q^{-\frac{1}{\sigma}} = \frac{\sigma-1}{\sigma} p = \beta(1-\delta)\frac{\partial V(s')}{\partial s'} + \lambda \\
z :& \omega - \mu = \beta(1-\delta)\frac{\partial V(s')}{\partial s'} \\
s :& \frac{\partial V(s)}{\partial s} = \beta(1-\delta)\frac{\partial V(s')}{\partial s'} + \lambda
\end{aligned}$$

Analytical example — price rule

Rearranging these terms notice that: $p = \frac{\sigma}{\sigma-1}\frac{\partial V(s)}{\partial s} = \frac{\sigma}{\sigma-1}(\omega - \mu + \lambda)$

$q < s$ & $z > 0 \implies \lambda, \mu = 0 \implies$ price is markup over purchase price, ω

Implication 1: price depends on replacement cost, not the actual purchase price

Can bring goods into next period by ordering or not selling which reduces need to order

Rearranging these terms notice that: $p' = \frac{\sigma}{\sigma-1}\frac{\partial V(s')}{\partial s'} = \frac{\sigma}{\sigma-1}\frac{\omega-\mu}{\beta(1-\delta)}$

$q < s$ & $z > 0 \implies \lambda, \mu = 0 \implies$ price is markup over marginal cost which includes interest and deprciation

Implication 2: Full pass-through if they import

Research idea: how do prices change following stockout?

Analytical example — price rule

If $q < s$ then $\lambda = 0 \rightarrow \frac{p'}{p} = \frac{V(s')}{V(s)} = \frac{1}{\beta(1-\delta)}$

Implication 3: Prices rise smoothly as you run down stock (increases with R & δ)

Analytical example w/ fixed cost

Now let us consider the case with fixed cost f and a delivery lag

But will fix demand to 1 & no depreciation (need to simplify somehow)

$$V(s) = \begin{cases} ps + \max\{\beta V(0), \max_z [-f - \omega z + \beta V(z)]\} & s \in [0,1] \\ p + \max\{\beta V(s-1), \max_z [-f - \omega z + \beta V(s-1+z)]\} & s \in (1,\infty) \end{cases}$$

Analytical example w/ fixed cost

Begin with the case where $s = 0$

Firm orders n units immediately and then sell 1 for next n periods

$$\begin{aligned} \text{NPV} \Rightarrow r_n &= -f - \omega z + (\beta + \beta^2 + \ldots + \beta^n)p \\ &= -f - \omega z + \beta \frac{1-\beta^n}{1-\beta} p \end{aligned}$$

Repeat this whenever $s = 1$ i.e. at $t = n, 2n, ...,$ value of this (Y_n)

$$\begin{aligned} Y_n &= r_n(1 + \beta^n + \beta^{2n} + \ldots) = \frac{r_n}{1-\beta^n} \\ &= -\frac{f + \omega n}{1-\beta^n} + \frac{\beta}{1-\beta} p, \qquad n = 1, 2, \ldots \end{aligned}$$

Analytical example w/ fixed cost

Guess that $V(0) = Y_N$ is optimal for $s = 0$

For $s \in [0, 1]$, sell existing stock at p and order N units

$$V(s) = ps + Y_N, \qquad s \in [0, 1]$$

For $s \in (1, 2]$, 2 options: sell 1 and order $N - (s - 1)$
or sell 1 and order nothing, let A be the inflection point

$$V(s) = \begin{cases} p + \omega(s-1) + Y_N, & s \in (1, 1+A) \\ p + \beta V(s-1), & s \in [1+A, 2] \end{cases}$$

Must be indifferent at $s = 1 + A$: $p + \omega A + Y_N = p + \beta V(A)$

Since $A \in [0, 1]$, $V(A)$ given above

$$A = \frac{(1-\beta) Y_N}{\beta p - \omega}$$

Analytical example w/ fixed cost

For the case when $s \in (2, \infty)$, guess the firm sells 1 every period until $s \in (1, 2]$

$$V(s) = p + \beta V(s-1), \qquad s \in (2, \infty)$$

$$V(s) = \begin{cases} ps + Y_N, & 0 \leq s \leq 1 \\ p\frac{1-\beta^{\nu(s)}}{1-\beta} + \beta^{\nu(s)-1}(\omega\alpha(s) + Y_N), & s > 1 \ \&\ \alpha(s) \in [0, A] \\ p\frac{1-\beta^{\nu}(s)}{1-\beta} + \beta^{\nu(s)}(p\alpha(s) + Y_N), & s > 1 \ \&\ \alpha(s) \in [A, 1) \end{cases}$$

where $\nu(s)$ is largest integer $< s$ and $\alpha(s) = s - \nu(s) \in [0, 1)$

Takeaways:
1. N depends on β, f, p, and ω
2. (s, S) policy: $s = 1 + A$ and $S = N$ (s depends on demand, but not S)
3. Highly non-linear solution

Inventory dynamics

- Even among dynamic models, inventory model is unusual
- Decision today affects a continuous state variable tomorrow
- Not true for most dynamic models (sunk exporting costs, etc)

Recall: importer's Problem

- Importer Decides between Importing or not importing

$$V(s,\nu) = \max[V^a(s,\nu), V^n(s,\nu)]$$

$$V^a(s,\nu) = \max_{p,z>0} q(p,s,\nu)p - \omega z - f + \beta EV(s',\nu')$$

$$V^n(s,\nu) = \max_{p>0} q(p,s,\nu)p + \beta EV(s',\nu')$$

Subject to

$$q(p,s,\nu) = min(e^{\nu} p^{-\sigma}, s)$$

$$s' = \begin{cases} (1-\delta)[s - q(p,s,\nu) + z] & \text{if import} \\ (1-\delta)[s - q(p,s,\nu)] & \text{o/w} \end{cases}$$

Value Function Iteration

- Stockey et al. (1989): Define Bellman Operator $B(V)$, then
  1. V^* such that $B(V^*) = V^*$ exists and is unique.
  2. $V^* = \lim_{t\to\infty} B^t(V^0)$ for any continuous function V^0. In addition $B^t(V^0)$ converges to V^* monotonically.
- Value function iteration is the solution method and can be performed using three alternative Global Methods - solution is valid not only around some point in the state space:
  1. Discretization
  2. Finite Element Method
  3. Weighted-residual or Chebyshev regression method
- We here discuss the first two.
- The model here presented is solved using the finite element method with spline interpolation.

Value Function Iteration - Discretization

- Computers cannot deal with continuous state space, so we define $V(x)$ over $\underline{x}, \bar{x}$ and discretize the state space over n_x points. Then we can represent the value function as a set of n_x points $\{V_i\}_{i=1}^{n_x}$, where $V_i = V(x_i)$.

1. Set $n_x, \underline{x}, \bar{x}, \varepsilon$, where ε is tolerance error, and grid points $\{x_1, x_2, ..., x_{n_x}\}$.
2. Initial guess: $V^0 = \{V_i^0\}_{i=1}^{n_x}$.
3. Update value function obtaining $V^1 = \{V_i^1\}_{i=1}^{n_x}$ from:
   - $\forall x_j \in [\underline{x}, \bar{x}]$, $V_{i,j}^1 = u(x_i, x_j) + \beta V_j^0$, where $u(x_i, x_j)$ is instant return from being at x_i and choosing x_j and V_j^0 is guess of value of being at x_j in the next period.
   - Choose j which gives the highest value among $\{V_{i,j}^1\}$ and call it V_i^1. Store optimal decision as $j = g_i \in \{1, ..., n_x\}$
   - Do this for i, so that $V^1 = \{V_i^1\}_{i=1}^{n_x}$.
4. If $d = \max_{i \in \{1,\ldots,n_x\}} |V_i^0 - V_i^1| < \varepsilon$ then done. Otherwise, go back to previous step using V^1 as updated guess.
5. Robustness checks: bounds in state space not binding, sensitivity to ε and n_x.

Value Function Iteration - Finite Element Method

- Instead of discretizing the state space, this method approximates the value function by using interpolation across discrete points, such as piecewise-linear interpolation or cubic splines.

- Here we will use piecewise-linear interpolation, where:
  $\tilde{V}(x) = V_i + \frac{V_{i+1} - V_i}{x_{i+1} - x_i}(x - x_i)$

1. Repeat steps 1 and 2 of discretization (set grid and guess) and call the value function implied by piecewise-linear interpolation with $\{V_i^0\}_{i=1}^{n_x}$ as $\tilde{V}^0(x)$ and the optimal decision rule as $\tilde{g}(x)$.

2. $\forall i = 1, ..., n_x \ g_i = argmax_{x' \in [\underline{x}, \bar{x}]}\{u(x_i, x') + \beta \tilde{V}^0(x')\}$
To find an optimum use a one-dimension optimization algorithm - Golden Search or Newton's Method explained later and used in the code that solves the here explained model.

3. Once g_i is obtained update value function: $V_i^1 = u(x_i, g_i) + \beta \tilde{V}^0(g_i)$

4. Then do steps 4 (tolerance check) and 5 (robustness check) of discretization.

One Dimensional Optimization - Golden Search

- Golden Search method is used to find extremum of a function on a given interval
- Let a^0 and d^0 be the endpoints, and define 2 points within the interval, b^0 and c^0, where $b^0 = a^0 + (d^0 - a^0)(1 - R)$ and $c^0 = a^0 + R(d^0 - a^0)$
- R here is the Golden ratio $= (\sqrt{5} - 1)/2$
- Using the golden ratio makes sure that $\frac{d-a}{c-a} = \frac{c-a}{b-a}$
- This will be used to reduce the number of function evaluations

One Dimensional Optimization - Golden Search

- If $f(b^0) > f(c^0)$,
  - set $a^1 = a^0$ and $d^1 = c^0$
  - The benefit of using golden ratio is that we only need to evaluate function at 1 more point since $c^1 = b^0$ (by the formula) and we already have $f(b^0) = f(c^1)$
  - Define $b^1 = a^1 + (d^1 - a^1)(1 - R)$ and evaluate function here
  - Compare $f(b^1)$ and $f(c^1)$ and repeat
- If $f(b^0) \leq f(c^0)$,
  - set $d^1 = d^0$ and $a^1 = b^0$
  - Evaluate function at 1 more point since $b^1 = c^0$ (by the formula) and we already evaluated $f(c^0) = f(b^1)$
  - Define $c^1 = a^1 + R(d^1 - a^1)$ and evaluate function here
  - Compare $f(b^1)$ and $f(c^1)$ and repeat

One Dimensional Optimization - Newton's Method

1. Set a guess x_0.
   - This guess would preferably be the one obtained using the value function after iteration using golden search.

2. Compute first and second derivatives of $f(x)$ at x_0. If $f'(x_0) < \varepsilon$, done.

3. Apply first order Taylor series expansion around x_0 to obtain $\tilde{f}(x) = f(x_0) + f'(x_0)(x - x_0)$.

4. Solving for $\tilde{f}(x_1) = 0$ yields $x_1 = x_0 - [f'(x_0)]^{-1} f(x_0)$.

5. Update $x_0 = x_1$ and back to step 3.

Recall: importer's Problem

- Importer Decides between Importing or not importing

$$V(s,\nu) = \max[V^a(s,\nu), V^n(s,\nu)]$$

$$V^a(s,\nu) = \max_{p,z>0} q(p,s,\nu)p - \omega z - f + \beta EV(s',\nu')$$

$$V^n(s,\nu) = \max_{p>0} q(p,s,\nu)p + \beta EV(s',\nu')$$

Subject to

$$q(p,s,\nu) = min(e^{\nu} p^{-\sigma}, s)$$

$$s' = \begin{cases} (1-\delta)[s - q(p,s,\nu) + z] & \text{if import} \\ (1-\delta)[s - q(p,s,\nu)] & \text{o/w} \end{cases}$$

Solving for the Steady State

- Functions from CompEcon package by Miranda and Fackler would be used
- Begin by setting the parameter values
- 2 states - beginning stock level and the demand shock (log-normal distribution)
- Discretize normal distribution for demand shocks using qnwnorm() function, gives relevant grid points and their probabilities
- Set beginning and end points for stock state (check robustness for this later)
- define a function space using function fundef() from CompEcon, this family contains grid points for states and the method of approximation (splines, linear, etc)
- This function family will be later used to fit value function on states

Value function iteration

- There will be three value functions: Ordering $V^a(s, \nu)$, not ordering $V^n(s, \nu)$ and expected max value $E_\nu[V(s, \nu)]$
- Guess value functions/policies first and then solve for policy functions using golden search
- Ordering
  - For each state grid point, take a given price and calculate the optimal order associated with it (golden search)
  - Then for each state grid point, choose the price that maximizes value of ordering, get optimal prices and orders
- Not Ordering
  - Solve for the price for each state point that maximizes value of not ordering, optimal prices
- Repeat until value functions converge

Value function iteration

1. Let $V1(s,\nu) = V^a(s,\nu)$, $V2(s,\nu) = V^n(s,\nu)$, $V3(s) = E_\nu[V(s,\nu)]$
2. Guess $V1^0$, $V2^0$, $V3^0$
3. Ordering
   - For each (s,ν), for each grid point of price p, get optimal order $i^*(s,\nu|p)$ and $V1^1(s,\nu|p,i^*(s,\nu;p))$
   - Get $V1^1(s,\nu) = max_p V1^1(s,\nu;p,i^*(s,\nu;p))$
4. Not Ordering
   - For each (s,ν), for each grid point of price p, get $V2^1(s,\nu|p)$
   - set $V2^1(s,\nu) = max_p V2^1(s,\nu|p)$
5. Obtain $V3^1(s) = E_\nu\{max[V1^1(s,\nu), V2^1(s,\nu)]\}$
6. Repeat until the three functions converge

Simulate the model

- Using the optimal policies, simulate K firms for T periods to get relevant statistics
- Draw demand shocks for these firms from a normal distribution over all time periods
- Start with a uniform distribution over stock levels and simulate using their optimal policies
- Use law of motion of stock level to determine next period's beginning stock
- Drop first few periods and calculate required statistics (HH index, I/P ratio, I/S ratio)

Unanticipated Shock

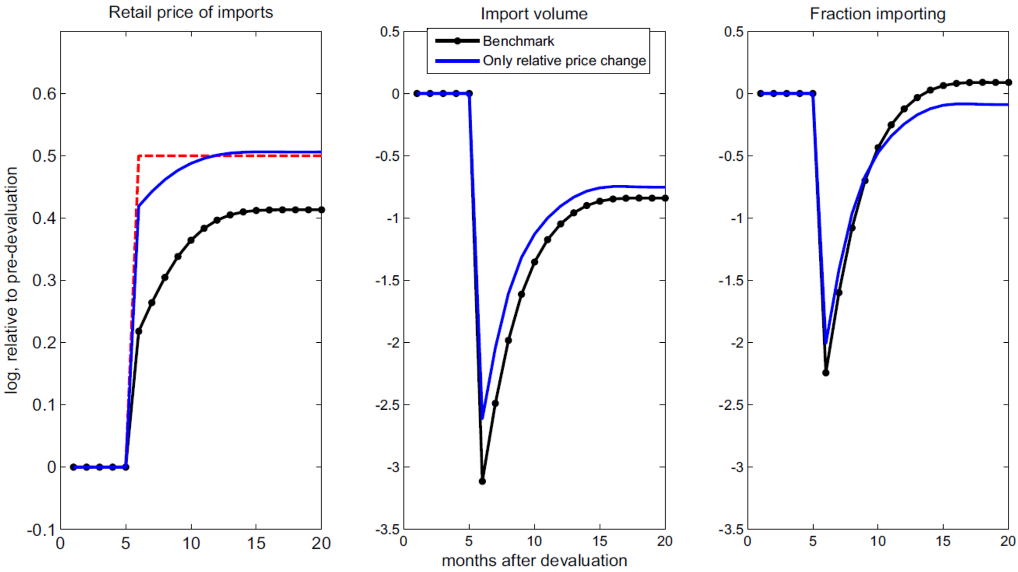
- Here we look at the unanticipated rise in ω as a result of devaluation
- Since this is a partial eq'm model, we can feed in the sequence $\{\omega_t\}$ and let agents respond to it over time
- The key here would be to start with a stationary distribution and keep track of distribution at every period
- Here demand shocks will allow us to aggregate the ordering decisions to get total imports, even though not every one is ordering every period
- Since the shock is unanticipated, we move forward by letting agents know as the path unfolds

Calculating Transition

- Simulate the model with optimal decision rules for T period so that the distribution over stock converges to a stationary level
- Then solve SS after shock and use its policy rules
- Roll the model forward for 18 periods to reach new stationary distribution
- Aggregate price index is calculated every period using the transitional distributions
- The idea is that every producer responded differently to the shock depending on the beginning stock levels and demand shocks

Unanticipated Devaluation

Figure 6: Response of model economy to devaluation



Gradual Shock

- Using the model, we can also study the response of price index and imports a gradual shock
- Basically, at time period T_A agents see a slight devaluation but then see the whole path of future changes
- This ability to see the path of $\{\omega_t\}$ will cause an immediate increase in imports because of the intertemporal trade-off (imports more expensive in future)

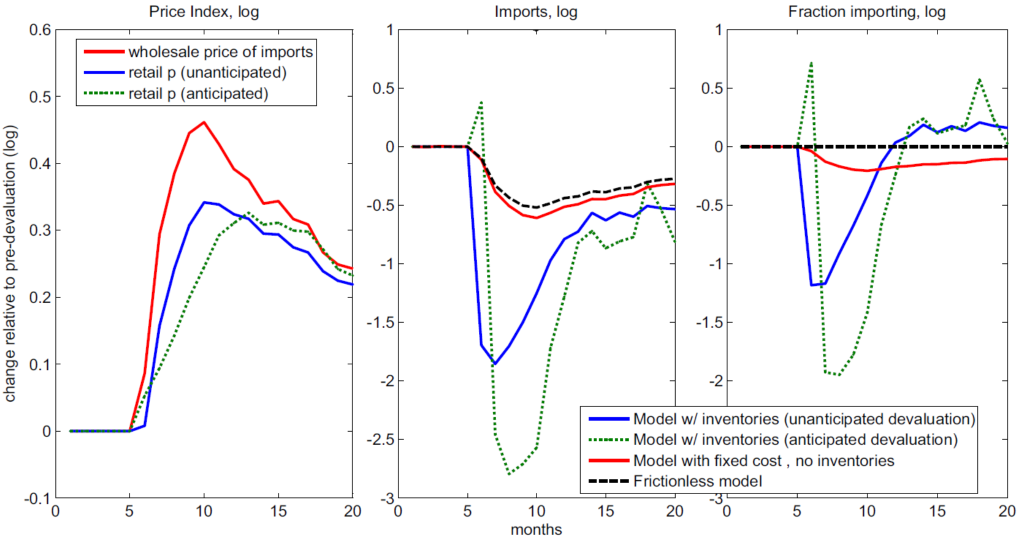
Shooting Algorithm Summary

- The only difference here is that we solve backwards using the third value function as expected max value and get policy functions for each time period
- After solving for the very last period, we obtain 3 value functions for the new steady state at period T
- Then at period T-1 we solve for the value functions for that period using future expected value as the third value function of period T
- Once we reach to t=1, we start solving forwards with the decision rules using the old SS stationary distribution as the initial one, recording the distributions at each period of time
- we have to move backwards before moving forwards because while calculating the distribution we must consider the value functions that have fully anticipated the path forward

"Life can only be understood backwards; but it must be lived forwards."
Soren Kierkegard

Gradual Anticipated Devaluation

Figure 8: Response to a gradual devaluation



Several periods of Anticipation

- We can feed in the sequence of any desired shock
- Suppose you want to put a permanent shock that is anticipated 2 periods in advance
- Feed the sequence of Shocks $\omega =$ [1 , 1 , 1.05 , ... , 1.05]
- This will again calculated the decision rules moving backwards and then iterate it forward to get the distribution with changes

Possible amendments to the model

- There are lots of different cases that can be studied under this setup
  - Reduction in consumption before devaluation (as in AKM (2010a))
  - Foreseeing a permanent or transitory shock few periods in advance (Khan & Khederlarian 2021)
  - News shock that does not realize when the expected time comes
  - Short-term import supply congestion term (by additional loop over price of imports)
  - Multiple anticipated shocks

Possible Experiments

- Aggregate uncertainty shock that does not realize
- Idiosyncratic uncertainty shock i.e. uncertainty for firms with a specific τ
- Anticipation of multiple shocks conditional on realization (tradewar)
- Uncertainty over timing of the shock e.g. if the shock didn't happen in pd t_0, it happens in pd t_1 with certain probability