

The EK-framework and Simulation, by Mike Waugh

One of the overlooked properties of the EK framework (Eaton and Kortum (2002), Bernard et al. (2003), or Eaton et al. (2011)) is how, via simulation, the model can be solved, calibrated, and analyzed.

Let's focus on a specific example. The Bernard et al. (2003) paper wants to match and understand the micro-behavior (firms, productivity dispersion, markups) of the economy in response to globalization-like shocks. How do markups respond to a reduction in trade costs and for whom? Who expands or not or exits? These questions are what they designed the paper / model to speak to.

To answer these kinds of question one needs to be able to generate a dataset from the model that mimics the kind of data set that one might see in a Census RDC.¹ This is where the explicit distributional assumptions of the EK framework help a lot. First, the distributional assumption provides away to aggregate and connect the model with aggregate outcomes like trade flows (this is what Eaton and Kortum (2002) emphasized). Second, they provide a way to construct outcomes at the micro-level through simulation. I'm going to focus on this latter point in the notes below.

1. Simulation — The Basics

A key idea in simulation based approaches is what is called the **Inverse CDF Method**. This method essentially allows us to generate random variables from any distribution as long as we can compute the inverse CDF of the distribution. Then once we have random variables from distribution that we want, we can use **Monte Carlo Integration** to approximate integrals numerical that may be difficult / impossible to compute analytically. This section briefly discusses these two approaches.

1.1. The Inverse CDF Method

The idea is this: suppose we want to draw random samples from a continuous random variable X with CDF $F_X(x) = \Pr(X \leq x)$. In the EK context, this could be the technology draws and (as will show below) prices directly. The next step is that if we can compute F_X^{-1} , the inverse CDF method (also called the quantile transform) gives us an exact sampling algorithm.

Algorithm.

1. Draw $U \sim \text{Uniform}(0, 1)$.
2. Set $X = F_X^{-1}(U)$.

¹This is something one hears Ellen talking about — the idea that one should treat model output just like data. I think its a very important concept to understand.

Then $X \sim F_X$.

Why does this work? The key result is that if $U \sim \text{Uniform}(0, 1)$, then the random variable $F_X^{-1}(U)$ has distribution F_X . To see this, note that

$$\Pr(F_X^{-1}(U) \leq x) = \Pr(U \leq F_X(x)) = F_X(x), \quad (1)$$

where the first equality uses the fact that F_X is monotone increasing (so applying F_X to both sides preserves the inequality), and the second equality uses the fact that U is uniform on $(0, 1)$.

1.2. Monte Carlo Integration

The inverse CDF method becomes especially powerful when combined with simulation to approximate integrals that may be difficult or impossible to evaluate analytically. In the EK context, it's true that with the Fréchet distribution the integrals we care about are computable by hand. However, there are instances where this integral is not easy, e.g., what is the variance of TFPR in BEJK? This is where Monte Carlo integration comes into play.

Setup. Suppose we want to compute

$$I = \int g(x) f(x) dx = \mathbb{E}_{X \sim f}[g(X)], \quad (2)$$

where f is a probability density and g is some function of x . The law of large numbers tells us that if X_1, X_2, \dots, X_N are i.i.d. draws from f , then the sample mean

$$\hat{I}_N = \frac{1}{N} \sum_{i=1}^N g(X_i) \xrightarrow{a.s.} I \quad \text{as } N \rightarrow \infty. \quad (3)$$

This is the **Monte Carlo estimator** of I . Below I sketch out the procedure to construct this estimator.

Procedure.

1. Use the inverse CDF method (or any valid sampler) to draw $X_1, \dots, X_N \stackrel{iid}{\sim} f$.
2. Compute $\hat{I}_N = \frac{1}{N} \sum_{i=1}^N g(X_i)$.

The question that always arises with this method is how accurate is it? How many draws do I need? If $\text{Var}(g(X)) = \sigma^2 < \infty$, the central limit theorem gives

$$\sqrt{N} (\hat{I}_N - I) \xrightarrow{d} \mathcal{N}(0, \sigma^2), \quad (4)$$

so the standard error of the Monte Carlo estimator is σ/\sqrt{N} . So to halve the variance requires

quadrupling the number of draws, that is the rate of convergence is $O(N^{-1/2})$ regardless of dimension. At first, the $O(N^{-1/2})$ rate may look slow compared to methods like quadrature. However, the Monte Carlo rate *does not depend on the dimension of the integral*. Deterministic methods suffer from the curse of dimensionality—their convergence deteriorates rapidly as the number of dimensions grows — while Monte Carlo maintains $O(N^{-1/2})$ whether you are integrating over \mathbb{R} or \mathbb{R}^{100} .

In the EK context, this is why simulation methods are the way to go. Recall that in our EK notes, one can see that integration is increasing in the number of countries. Note as well, in the supply chain discussion, the options to consider might increase, but this argument says that Monte Carlo maintains $O(N^{-1/2})$.

2. Example: EK 2002

Let's put this approach to use. First, let's focus on the competitive EK environment.

Recall that what we want to compute is the expenditure share:

$$\pi_{nj} = \frac{\int_0^1 x_{nj}(\omega) p_{nj}(\omega)^{1-\sigma} d\omega}{\int_0^1 \sum_k x_{nk}(\omega) p_{nk}(\omega)^{1-\sigma} d\omega}, \quad (5)$$

or what share of expenditure does the consume in n purchase from j . And then the pieces that go into this are the x function which is

$$x_{nj}(\omega) = \begin{cases} 1, & \text{if } p_{nj}(\omega) \leq \min_{j'} \left\{ p_{nj'}(\omega) \right\} \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

and we have prices are set competitively where

$$p_{ni}(z) = \frac{w_i d_{ni}}{z_i(\omega)}. \quad (7)$$

2.1. Fréchet Inverse CDF

Recall that for each good ω , productivity $z_j(\omega)$ in country j is an independent draw from a Fréchet (Type II extreme value) distribution with CDF:

$$F_j(z) = \exp \left\{ -T_j z^{-\theta} \right\} = \text{Prob} \left\{ z_j(\omega) \leq z \right\} \quad (8)$$

Then we can use the same logic as before where

$$F_j^{-1}(U) = \left(\frac{-\log U}{T_j} \right)^{\frac{-1}{\theta}} \quad (9)$$

and then from (9) we can sample the productivity terms z as we wish.

2.2. Monte Carlo Integration of Demand

The focus here is that we want to compute the integrals on the top and bottom of (5). Here is a step by step way I think about this.

1. I want to think of discrete points on the continuum to understand the map from the integral to draws in some economic sense. These discrete points are the number of draws used to construct (3). I'll label these points as integers $\omega = 1 \dots N^d$ where N^d is the number of discrete points / draws employed.
2. Then for each good ω , we use (9) to draw a vector $\bar{z}(\omega) = \{z_1(\omega), z_2(\omega), \dots, z_J(\omega)\}$. This vector of technology draws is of length J = the number of countries. Note that each country has their own specific distribution for which $z_j(\omega)$ would come from and we can do this independently because that is the assumption.
3. The estimator of the numerator of (5) is

$$\frac{1}{N^d} \sum_{\omega=1}^{N^d} x_{nj}(\bar{z}(\omega)) \cdot p_{nj}(z_j(\omega))^{1-\sigma}, \quad (10)$$

where notice how the function $x_{nj}(\bar{z}(\omega))$ depends upon the entire vector of technologies associated with point ω because we need to be making comparisons as to which country is the low cost supplier. Then the price **only** depends upon the j specific technology draw.

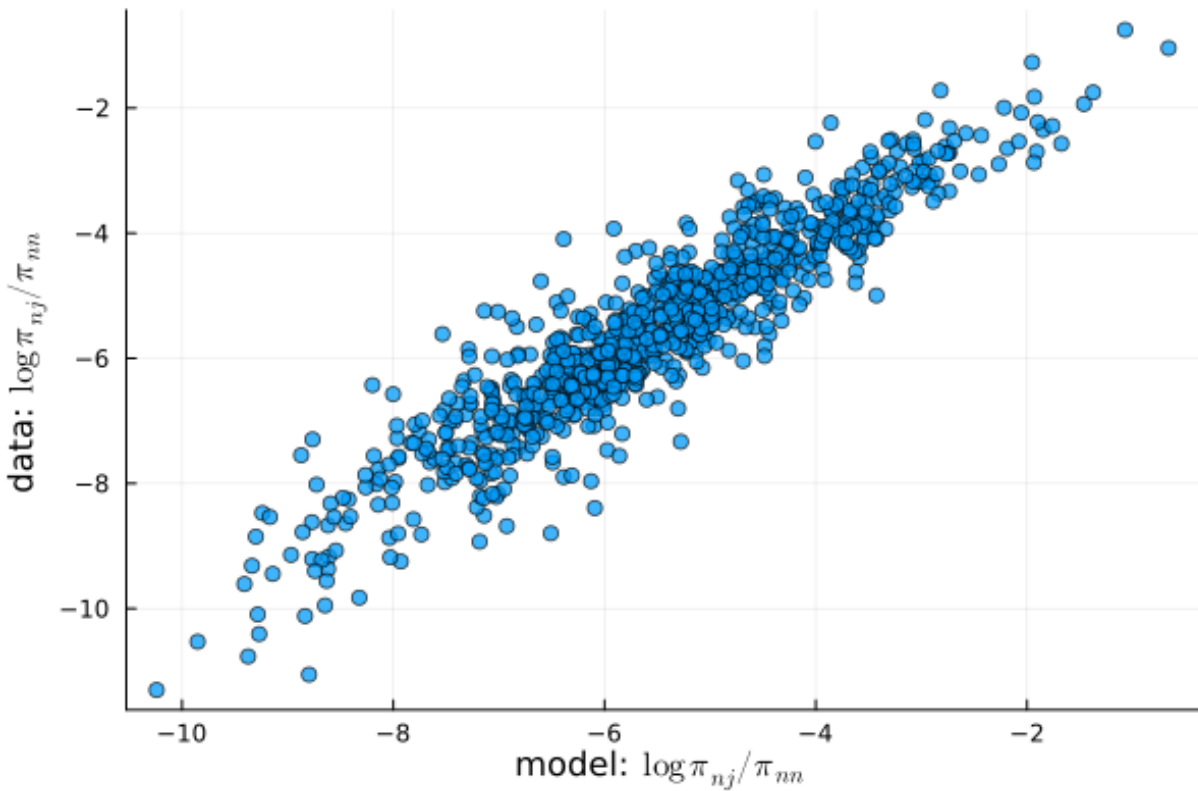
4. Then the estimator of the denominator of (5) simply is the sum of (10) across all possible source countries

$$\frac{1}{N^d} \sum_{\omega=1}^{N^d} \sum_k \left\{ x_{nk}(\bar{z}(\omega)) \cdot p_{nk}(z_k(\omega))^{1-\sigma} \right\}, \quad (11)$$

5. Then the expenditure share can be combining the two so

$$\pi_{nj} \approx \frac{\sum_{\omega=1}^{N^d} x_{nj}(\bar{z}(\omega)) \cdot p_{nj}(z_j(\omega))^{1-\sigma}}{\sum_{\omega=1}^{N^d} \sum_k \left\{ x_{nk}(\bar{z}(\omega)) \cdot p_{nk}(z_k(\omega))^{1-\sigma} \right\}}. \quad (12)$$

Figure 1: Simulated Trade Flows



where notice that we can dispense with $\frac{1}{N^{\alpha}}$ scaling when computing this fraction.

In some ways this is very straightforward. In the back of these notes I have the algorithm (Algorithm 1) from the code that I've been working with.

2.3. Does it work well?

Yes. And fast.

The figure in Figure 1, plots model versus the data. Note that it should not be perfect because the gravity regression does not fit perfectly. This, however, is a very good fit.

3. Connection with Gravity and Prices

There is one more detail that makes simulation in the EK framework very continent. Stepping back when one looks through the equations above, one might say well I need to know a lot to do all of this. I need to know trade costs, wages, the technology parameters T s, in addition to a θ and a σ . It turns out that one can infer a lot of these objects from trade data and do so without even having to compute and equilibrium.

So the way this is done is to estimate the parameters for the country-specific productivity distri-

butions and trade costs (scaled by θ) from bilateral trade-flow data. Relative expenditure shares in the EK model take the form

$$\log \left(\frac{\pi_{nj}}{\pi_{nn}} \right) = S_j - S_n - \theta \log \tau_{nj}, \quad (13)$$

where S_i is defined as $\log [T_i w_i^{-\theta}]$. The goal is to estimate the objects S_i for all $i = 1, \dots, N$ and $\theta \log \tau_{ni}$ for all country pairs n and i such that $n \neq i$. The previous equation gives rise to an empirical gravity equation that corresponds to the theoretical expression in (13). It is given by

$$\log \left(\frac{\pi_{nj}}{\pi_{nn}} \right) = \hat{S}_j - \hat{S}_n - \hat{\theta} \log \hat{\tau}_{nj} + \nu_{nj}. \quad (14)$$

\hat{S}_i 's are recovered as the coefficients on country-specific dummy variables given the restrictions on how trade costs can covary across countries. Now notice how we have $N^2 - N$ observations on the left-hand side and we have $N + N^2 - N$ parameters to estimate. To get anywhere, we need to make a functional form assumption on the trade costs. The simplest way to deal with this is to assume that they are a function of distance. In Waugh (2010), I used an expression where trade costs take the following functional form

$$\log \hat{\tau}_{ni} = d_k + b_{ni} + ex_i. \quad (15)$$

Here, trade costs are a logarithmic function of distance, where d_k with $k = 1, 2, \dots, 6$ is the effect of distance between country i and n lying in the k -th distance interval. b_{ni} is the effect of a shared border in which $b_{ni} = 1$ if country i and n share a border and zero otherwise.

Now the key observation is that the parameter estimates obtained from the gravity regression (14) are sufficient to simulate trade flows and micro-level prices up to a constant, $\hat{\theta}$.

The connection to simulation is direct. Notice from (7) that for any good ω , the delivered price from j to n is $p_{nj}(\omega) = d_{nj} \cdot w_j / z_j(\omega)$. Thus, rather than simulating productivities $z_j(\omega)$ separately, it is sufficient to simulate the *inverse of marginal costs*, defined as

$$u_j(\omega) \equiv \frac{z_j(\omega)}{w_j}, \quad (16)$$

so that delivered prices are simply $p_{nj}(\omega) = d_{nj} / u_j(\omega)$. Since $z_j(\omega)$ is Fréchet distributed, u_j inherits a distribution with CDF

$$\Pr(u_j(\omega) \leq u) = \exp \left(-\tilde{S}_j u^{-\theta} \right), \quad \text{with} \quad \tilde{S}_j = \exp(S_j) = T_j w_j^{-\theta}. \quad (17)$$

This is the important connection: the S_j 's recovered as country fixed effects from the gravity

equation (14) are *exactly* the parameters that govern the distribution from which we sample marginal costs. Applying the inverse CDF method to (17) gives

$$u_j(\omega) = F_j^{-1}(U) = \left(\frac{-\log U_{j\omega}}{\tilde{S}_j} \right)^{-1/\theta}, \quad U_{j\omega} \sim \text{Uniform}(0, 1), \quad (18)$$

which is the same inverse Fréchet formula as (9), but now parameterized directly with the gravity estimates \hat{S}_j . Thus, having obtained the coefficients \hat{S}_j from the first-stage gravity regression and estimates of trade costs $\hat{\tau}_{ni}$, we can simulate the inverse of marginal costs and prices without ever needing to separately identify T_j or w_j .

4. Example: BEJK 2003

Now consider the Bertrand competition environment of Bernard et al. (2003). The key difference from the competitive EK model is in how prices are determined. Rather than pricing at marginal cost, the low-cost producer in each market sets its price strategically—just low enough to keep the next-best competitor out. This means that simulation now requires us to track not only the lowest-cost supplier, but also the relevant second-best alternative.

4.1. Two Draws per Good

Recall that in the BEJK setup, each country j draws *two* independent productivity levels for each good ω . The first draw $z_j^{(1)}(\omega)$ is the most efficient producer in country j ; the second draw $z_j^{(2)}(\omega)$ is the second-most efficient.

Sampling the first draw works exactly as in the EK case: apply the inverse CDF method to the Fréchet distribution parameterized by \tilde{S}_j (which would come from the gravity regression above). And I'll work in terms of marginal costs $c_j(\omega) = w_j/z_j(\omega)$, the first draw is

$$c_j^{(1)}(\omega) = \left(\frac{-\log U_{j\omega}^{(1)}}{\tilde{S}_j} \right)^{1/\theta}, \quad U_{j\omega}^{(1)} \sim \text{Uniform}(0, 1). \quad (19)$$

The second draw $c_j^{(2)}(\omega)$ is the second-order statistic: it must satisfy $c_j^{(2)}(\omega) \geq c_j^{(1)}(\omega)$ (i.e., a weakly higher marginal cost). BEJK show that the conditional distribution of the second-best productivity, given the first, has CDF

$$F\left(c_j^{(2)} \mid c_j^{(1)}\right) = 1 - \exp\left\{-\tilde{S}_j \left(c_j^{(2)}\right)^{-\theta} + \tilde{S}_j \left(c_j^{(1)}\right)^{-\theta}\right\}. \quad (20)$$

Note the structure: at $c_j^{(2)} = c_j^{(1)}$, the exponential term is $e^0 = 1$, so the CDF equals zero—confirming that the second-best cost is (almost surely) strictly above the first. As $c_j^{(2)} \rightarrow \infty$, the

CDF approaches one.

Applying the inverse CDF method to (20) gives the sampling formula

$$c_j^{(2)}(\omega) = \left(\frac{-\log V_{j\omega}}{\tilde{S}_j} + \left(c_j^{(1)}(\omega) \right)^{-\theta} \right)^{-1/\theta}, \quad V_{j\omega} \sim \text{Uniform}(0, 1), \quad (21)$$

where $V_{j\omega}$ is an independent uniform draw. The expression has an intuitive structure: it takes the first draw's contribution $\left(c_j^{(1)} \right)^{-\theta}$, adds the new randomness $-\log V/\tilde{S}_j$, and inverts. Since we are adding a positive quantity inside the parentheses relative to the first draw formula (19), the result is guaranteed to satisfy $c_j^{(2)}(\omega) \geq c_j^{(1)}(\omega)$ —the second-best producer always has weakly higher cost.

4.2. Bertrand Pricing

In BEJK, the low-cost supplier has market power and the price it charges is constrained by three objects:

1. **The second-lowest international cost.** Across all countries $j = 1, \dots, J$, find the lowest delivered cost and let j^* denote the low-cost supplier. So $c_{nj^*}^{(1)}(\omega)$ and the second-lowest delivered cost $c_{nj}^{(1)}(\omega)$ using each country's *first* draw.
2. **The second domestic draw of the winner.** This is the tricky one. That country, j^* own second-best producer has delivered cost $c_{nj^*}^{(2)}(\omega)$. If this is lower than any $c_{nj}^{(1)}(\omega)$ for $j \neq j^*$, then the binding competitive threat comes from within the winning country.
3. **The monopoly markup.** Under CES demand with elasticity σ , the unconstrained monopoly price is $\frac{\sigma}{\sigma-1} \cdot c_{nj^*}^{(1)}(\omega)$. The firm never charges above this even if the competitive threats are weak.

The price actually charged is the minimum of these three:

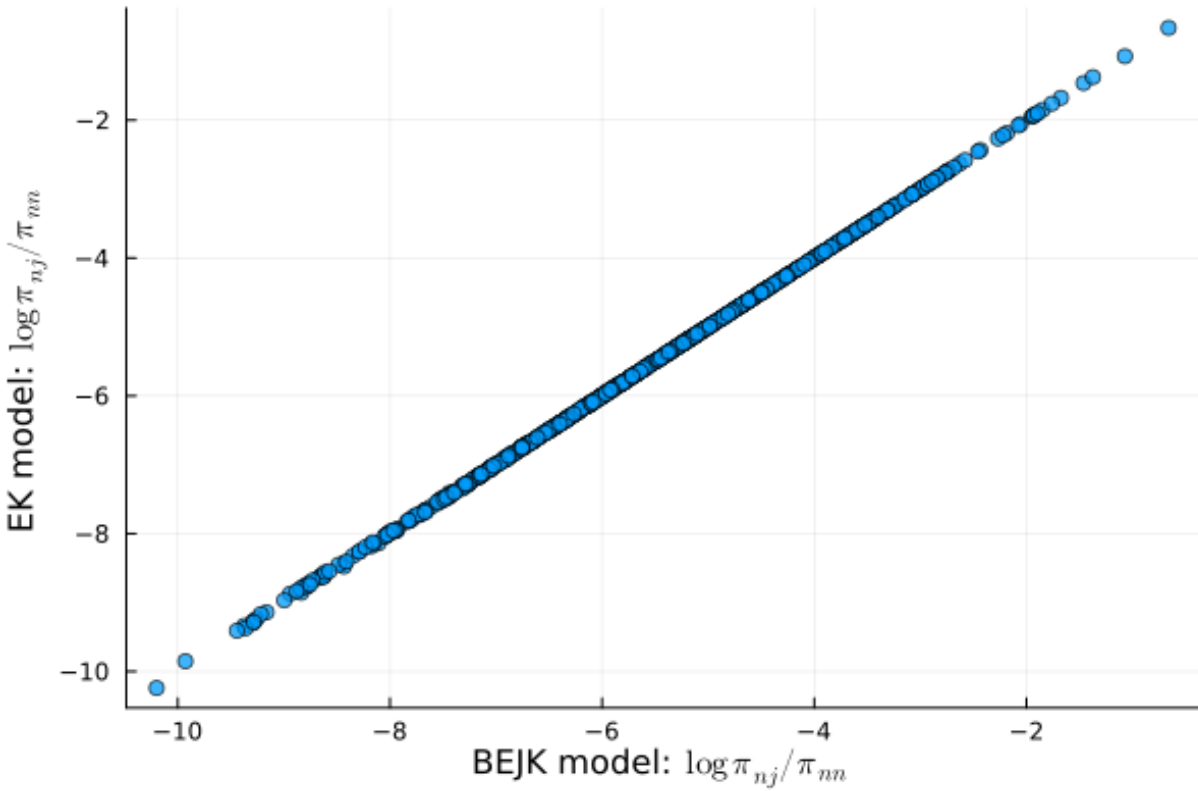
$$p_{nj^*}(\omega) = \min \left\{ c_{nj^*}^{(2)}(\omega), \quad \left\{ c_{nj}^{(1)}(\omega), \forall j \neq j^* \right\}, \quad \frac{\sigma}{\sigma-1} \cdot c_{nj^*}^{(1)}(\omega) \right\}. \quad (22)$$

Which of the three binds determines the markup that the low-cost supplier earns on good ω in market n .

4.3. Monte Carlo Integration of Demand

With pricing determined by (22), the construction of trade shares proceeds in the same spirit as Section 2.2, but with $p_{nj^*}(\omega)$ replacing the competitive price.

Figure 2: Simulated Trade Flows



The important point is that nothing changes about the Monte Carlo mechanics. We are still averaging over i.i.d. draws of goods. What changes is the *integrand*: instead of evaluating a competitive price, we now evaluate the Bertrand price, which depends on richer information about the cost distribution (the identity of the winner, their second draw, and the runner-up's cost).

4.4. Does it work well?

The figure in Figure 2, plots BEJK model output vs. EK model output. We know theoretically that these two, in aggregate, should be the same. And, well it looks pretty good to me.

Algorithm 1 Simulate Trade Patterns — Competitive EK Model

Require: $\mathbf{S} = \{S_1, \dots, S_J\}$ (gravity parameters, sufficient to simulate marginal costs), \mathbf{d} ($J \times J$ trade cost matrix), θ (Fréchet shape), σ (elasticity of substitution), N^d (number of goods/draws)
Ensure: π ($J \times J$ trade share matrix), \mathbf{p}^{\min} ($J \times N^d$ matrix of lowest delivered prices)

% Step 1: Draw productivities and compute marginal costs

```
1: for  $j = 1, \dots, J$  do                                ▷ Loop over countries
2:   for  $\omega = 1, \dots, N^d$  do                            ▷ Loop over goods
3:     Draw  $U_{j\omega} \sim \text{Uniform}(0, 1)$ 
4:      $p_j(\omega) \leftarrow \left( \frac{-\log U_{j\omega}}{S_j} \right)^{-1/\theta}$     ▷ Inverse Fréchet CDF
5:   end for
6: end for
```

% Step 2: Find lowest-cost supplier for each good in each destination

```
7: Initialize  $m_{nj} \leftarrow 0$  for all  $n, j$                 ▷ Accumulator for trade flows
8: for  $\omega = 1, \dots, N^d$  do                                ▷ Loop over goods
9:   for  $n = 1, \dots, J$  do                                ▷ Loop over importing countries
10:     $p^{\min} \leftarrow p_n(\omega); j^* \leftarrow n$           ▷ Initialize: home supplier
11:    for  $j = 1, \dots, J$  do                                ▷ Loop over potential exporters
12:       $p_{nj}^{cif}(\omega) \leftarrow d_{nj} \cdot p_j(\omega)$       ▷ Delivered price
13:      if  $p_{nj}^{cif}(\omega) < p^{\min}$  then
14:         $p^{\min} \leftarrow p_{nj}^{cif}(\omega); j^* \leftarrow j$     ▷ Update low-cost supplier
15:      end if
16:    end for
17:     $m_{n, j^*} \leftarrow m_{n, j^*} + (p^{\min})^{1-\sigma}$         ▷ Accumulate
18:    Record  $p_n^{\min}(\omega) \leftarrow p^{\min}$ 
19:  end for
20: end for
```

% Step 3: Normalize to obtain trade shares

```
21: for  $n = 1, \dots, J$  do
22:    $G_n \leftarrow \frac{1}{N^d} \sum_{j=1}^J m_{nj}$                 ▷ Monte Carlo estimate of price index term
23:   for  $j = 1, \dots, J$  do
24:      $\pi_{nj} \leftarrow \frac{m_{nj}/N^d}{G_n}$                 ▷ Trade share
25:   end for
26: end for
return  $\pi, \mathbf{p}^{\min}$ 
```

References

- BERNARD, A. B., J. EATON, J. B. JENSEN, AND S. KORTUM (2003): "Plants and productivity in international trade," *American economic review*, 93, 1268–1290.
- EATON, J. AND S. KORTUM (2002): "Technology, geography, and trade," *Econometrica*, 70, 1741–1779.
- EATON, J., S. KORTUM, AND F. KRAMARZ (2011): "An anatomy of international trade: Evidence from French firms," *Econometrica*, 79, 1453–1498.
- WAUGH, M. E. (2010): "International Trade and Income Differences," *The American Economic Review*, 100, 2093–2124.