

A Web Appendix

This is a Web Appendix to

“The Dynamic Spillovers of Entry: An Application to the Generic Drug Industry,”

A.1 Solving the Model

When the state space can only take on a finite set of values, Theorem 3.1 of Dutta and Sundaram (1998) implies that the game described in Section 3 has a stationary Markov perfect equilibrium in mixed strategies. An intermediate step of the proof shows that the Bellman equation (12) has a fixed point. This result motivates our computational strategy because it implies that a solution to the game can be found by value function iteration. Solution methods that rely on computing a value function are discussed in Rust (1996).

Parthasarathy (1973) showed that Theorem 3.1 of Dutta and Sundaram (1998) holds for a countably discrete state space. Theorem 5.1 of Dutta and Sundaram (1998) is the extension to a continuous, bounded, state space. The equilibrium strategy profiles provided by Theorem 5.1 may depend on periods t and $t - 1$ of the state vector. That does not exclude the possibility of a solution that depends only on the state at period t but does, in principle, obligate one to look over a larger domain if one cannot be found.

We could modify our problem to meet the requirements of Theorem 3.1 that the state space be finite and countable. However we rely on Theorem 5.1 instead as we do not have trouble computing pure strategy equilibria for the problem as posed with a continuous state space by iterating the Bellman equation (12) using a locally affine, step function to approximate the value function. This type of approximation can approximate any L_2 function to within arbitrary accuracy by taking the grid fine enough.

Strictly speaking, we do not have a finite state space that Theorem 3.1 of Dutta and Sundaram (1998) requires but we have never failed to compute an equilibrium due to nonconvergence. This is because the consequence of failure is that a particle gets killed. Based on the properties of the sequential importance sampler that particle is replenished at the resampling step of the particle filter algorithm described in Section 4. It is extremely rare that any of the N particles used to integrate the likelihood are killed. When it happens, there are no more than a few others, if any. We have never encountered a case where all N particles are killed. For a discussion of similar issues related to the complexity of guaranteeing existence and uniqueness of equilibrium in static games see Berry, Levinsohn, and Pakes (1995).⁴¹

⁴¹On p. 853, footnote 12 they state, “We assume that a Nash equilibrium to this pricing game exists, and that the equilibrium prices are in the interior of the firms’ strategy sets (the positive orthant). While Caplin and Nalebuff (1991) provide a set of conditions for the existence of equilibrium for related models of single product firms, their theorems do not easily generalize to the multiproduct case. However, we are able to check numerically whether our final estimates are consistent with the existence of an equilibrium. Note that none of the properties of the estimates require uniqueness of equilibrium, although without uniqueness it is not clear how to use our estimates to examine the effects of policy and environmental changes.”

Let the entry decisions of all $i = 1, \dots, I$ firms for a market opening at time t , i.e., the strategy profile of the dynamic game, be denoted by

$$A_t = (A_{1t}, \dots, A_{It}). \quad (22)$$

As discussed in Section 3, the strategy profile A_t at time t of the dynamic game is a function of the current period state variables (C_{1t}, \dots, C_{It}) and R_t . The vector of the log of the state variables at time t is

$$s_t = (c_{1t}, \dots, c_{It}, r_t). \quad (23)$$

In particular, equations (9) and (12) can be expressed in terms of s_t using $C_{it} = \exp(s_{it})$ for $i = 1, \dots, I$ and $R_t = \exp(s_{I+1,t})$. We describe the solution algorithm for a given parameter vector and a given state s_t at time t .

We begin by defining a grid on the state space which determines a set of $(I + 1)$ -dimensional hyper-cubes. The grid increments are chosen to be (fractional) powers of two.⁴² The centroid of the hyper-cube that contains a state vector s can be computed, element by element, by dividing by the increment, rounding to an integer, multiplying by the increment, and dividing by two. We use the centroid of each hyper-cube as the index K to the affine function whose domain is that hyper-cube. The rounding rules of the machine resolve which centroid a state on a grid boundary gets mapped to, although lying on a boundary is a probability zero event in principle. The entire grid itself is never computed because all we require is the mapping $s \mapsto K$, which is determined by the increments.

Let the vector $V_K(s_t)$ have as its elements the ex ante value functions $V_{i,K}(s_t)$, i.e., $V_K(s_t) = (V_{1,K}(s_t), \dots, V_{I,K}(s_t))$ (see equations (11) and (12)). To each K associate a vector b_K of length I and a matrix B_K of dimension I by $I + 1$. A given state point s_t is mapped to its centroid K and the value function at state s_t is represented by the affine function $V_K(s_t) = b_K + (B_K)s_t$. A value function $V_K(s_t)$ whose elements satisfy equation (12) is denoted $V_K^*(s_t) = b_K^* + (B_K^*)s_t$.

The game is solved as follows:

1. Given a state point s , get the centroid K that corresponds to it (we suppress the subscript t for notational convenience).⁴³
2. Check whether the fixed point $V_K^*(s)$ of the Bellman equations (12) at this centroid has already been computed, i.e., whether the (b_K^*, B_K^*) for the K that corresponds to s has been computed. If not, then use the following steps to compute it.
3. Start with an initial guess of the ex ante value function $V_K^{(0)}(s)$. An initial guess of the value function is represented by the coefficients $(b_K^{(0)}, B_K^{(0)})$ being set to 0.

⁴²As described below, this is done so that the centroid has an exact machine representation. This facilitates efficient computation through compact storage of objects indexed by the centroid.

⁴³In fact, because it is a stationary game, the subscript t does not really matter

4. Obtain a set of points s_j , $j = 1, \dots, J$, that are centered around K . The objective now is to obtain the ex ante value functions associated with these points to use in a regression to recompute (or update) the coefficients $(b_K^{(0)}, B_K^{(0)})$.
5. Ex ante value functions are evaluated at best response strategies. In order to compute these we must, for each s_j , compute the choice specific value function (9) at as many strategy profiles A as are required to determine whether or not the equilibrium condition in equation (10) is satisfied. In this process we need to take expectations to compute the continuation value $\beta \mathcal{E} \left[V_{K,i}^{(0)}(s_{t+1}) \mid A_{i,t}, A_{-i,t}, C_{i,t}, C_{-i,t}, R_t \right]$ that appears in equation (9), where we have used equation (11) to express equation (9) in terms of $V_K^{(0)}(s)$. To compute expectations over the conditional distribution of the random components of next period state variables, we use Gauss-Hermite quadrature. To do this, we obtain another set of points centered around each s_j , i.e., $s_{j,l}$, $l = 1, \dots, L$. These points are the abscissae of the Gauss-Hermite quadrature rule which are located relative to s_j but shifted by the actions A under consideration to account for the dynamic effects of current actions on future costs (see equation (5)). Expectations are computed using a weighted sum of the value function evaluated at the abscissae (more details are provided below).
6. We can now compute the continuation value at s_j for each candidate strategy A . We compute the best response strategy profile A_j^E corresponding to s_j by checking the Nash equilibrium condition (equation 10). As just described, the choice specific value function evaluated at (A_j^E, s_j) is computed using $V_K^{(0)}(s)$ and equation (9), and denoted by $V_K^{(1)}(A^E, s_j) = (V_{1,K}^{(1)}(A^E, s_j), \dots, V_{I,K}^{(1)}(A^E, s_j))$.
7. Next we use the “data” $(V_K^{(1)}(A^E, s_j), s_j)_{j=1}^J$ to update the ex ante value function to $V_K^{(1)}(s_j)$. This is done by updating the coefficients of its affine representation to $(b_K^{(1)}, B_K^{(1)})$ via a multivariate regression on this “data” (as described in detail below).⁴⁴
8. We iterate (go back to step 5) over the ex ante value functions $V_{i,K}^{(0)}(s), V_{i,K}^{(1)}(s), \dots$ by finding a new equilibrium strategy profile A^E for each s_j until convergence is achieved for the coefficients $(b_K^{(0)}, B_K^{(0)}), (b_K^{(1)}, B_K^{(1)}), \dots, (b_K^{(*)}, B_K^{(*)})$. This gives us $V_K^*(s) = b_K^* + (B_K^*)s$ for every s that maps to centroid K .

To summarize, the process of solving for the equilibrium begins with a conjecture $(b_K^{(l)} = 0, B_K^{(l)} = 0)$ for the linear approximation of the value functions at a given state at iteration $l = 0$. These guesses are then used in computing the choice specific value functions at iteration $l + 1$ using equation (9). This computation involves taking expectations over the conditional distribution of the future state variables, which is accomplished using Gaussian-Hermite quadrature. Once we have the choice specific value functions we compute the equilibrium strategy profile at iteration

⁴⁴ $V_K^{(1)}(A^E, s_j)$ will not equal $V_K^{(1)}(s_j)$ because the former is “data” and the later is a regression prediction.

$l + 1$ using equation (10). The best response strategy profile at iteration $l + 1$ is then used to compute the iteration $l + 1$ ex ante value functions via a regression that can be viewed as iterating equation (12). The iteration $l + 1$ ex ante value functions are then used to compute the iteration $l + 2$ choice specific value functions using equation (9), and the entire procedure is repeated till a fixed point of equation (12) is obtained. This iterative procedure solves the dynamic game. We next provide additional details about the steps of the algorithm described above to solve the model.

To describe the Gauss-Hermite quadrature procedure used in Step 5, note that if one conditions upon s_t and A_t , then the elements of s_{t+1} are independently normally distributed with means $\mu_i = \mu_c + \rho_c(c_{it} - \mu_c) - \kappa_c A_{it}$ for the first I elements (see equation 2), mean $\mu_{I+1} = \mu_R$ for the last element (see equation 6), standard deviations $\sigma_i = \sigma_c$ for the first I elements, and standard deviation $\sigma_{I+1} = \sigma_R$ for the last. Computing a conditional expectation of functions of the form $f(s_{t+1})$ given (A_t, s_t) such as appear in equations (9) and (12) is now a matter of integrating with respect to a normal distribution with these means and variances which can be done by a Gauss-Hermite quadrature rule that has been subjected to location and scale transformations. The weights w_j and abscissae x_j for Gauss-Hermite quadrature may be obtained from tables such as Abramowitz and Stegun (1964) or by direct computation using algorithms such as Golub and Welsch (1969) as updated in Golub (1973). To integrate with respect to $s_{j,t+1}$ conditional upon A_t and s_t the abscissae are transformed to $\tilde{s}_{t+1,j} = \mu_j + \sqrt{2}\sigma_j x_j$, and the weights are transformed to $\tilde{w}_j = w_j/\sqrt{\pi}$, where $\pi = 3.142$.⁴⁵ Then, using a $2L + 1$ rule,

$$\mathcal{E}[f(s_{t+1}) | A_t, s_t] \approx \sum_{j_1=-L}^L \cdots \sum_{j_I=-L}^L \sum_{j_{I+1}=-L}^L f(\tilde{s}_{t+1,j_1}, \dots, \tilde{s}_{t+1,j_I}, \tilde{s}_{t+1,j_{I+1}}) \tilde{w}_{j_1} \cdots \tilde{w}_{j_I} \tilde{w}_{j_{I+1}}. \quad (24)$$

If, for example, there are three firms and a three point quadrature rule is used, then

$$\mathcal{E}[f(s_{t+1}) | A_t, s_t] \approx \sum_{i=-1}^1 \sum_{j=-1}^1 \sum_{k=-1}^1 \sum_{l=-1}^1 f(\tilde{s}_i, \tilde{s}_j, \tilde{s}_k, \tilde{s}_l) \tilde{w}_i \tilde{w}_j \tilde{w}_k \tilde{w}_l.$$

We use three point rules throughout. A three point rule will integrate a polynomial in s_{t+1} up to degree five exactly.⁴⁶

Step 7 involves updating the ex ante value function using a regression. We next describe how we do this. As stated above, we have a grid over the state space whose boundaries are fractional powers of two over the state space.⁴⁷ We approximate the value function $V(s_t)$ by a locally indexed affine

⁴⁵These transformations arise because a Hermite rule integrates $\int_{-\infty}^{\infty} f(x) \exp(-x^2) dx$. Hence we need to do a change of variables to get our integral $\int_{-\infty}^{\infty} g(\sigma z + \mu) (1/\sqrt{2\pi}) \exp(-0.5z^2) dz$ to be of that form. A change of variables puts the equation in the line above in the form $\int_{-\infty}^{\infty} g(\sqrt{2}\sigma x + \mu) (1/\sqrt{\pi}) \exp(-x^2) dx$, which is where the expressions for $\tilde{s}_{t+1,i}$ and \tilde{w}_i come from.

⁴⁶If the \tilde{s}_{t+1} cross a grid boundary when computing (9) in Step 5, we do not recompute K because this would create an impossible circularity due to the fact that the value function at the new K may not yet be available. Our grid increments are large relative to the scatter of abscissae of the quadrature rule so that crossing a boundary will be a rare event, if it happens at all.

⁴⁷Recall that grid increments are chosen to be fractional powers of two so that the centroid has an exact machine representation. This facilitates efficient computation through compact storage of objects indexed by the centroid.

representation as described above. For the grid increments that determine the index of hyper-cubes we tried a range of values from 4 to 16 times the standard deviation of the state variables rounded to a nearby fractional power of two to scale the grid appropriately. The results are effectively the same. Hence in estimating the model we set the grid increments at 16 times the standard deviation of the state variables.⁴⁸ We compute the coefficients b_K and B_K as follows. They are first initialized to zero. We then generate a set of abscissae $\{s_j\}$ clustered about K and solve the game with payoffs (9) to get corresponding equilibria $\{A_j^E\}$. We substitute the (A_j^E, s_j) pairs into equation (9) to get $\{V(A_j^E, s_j)\}_{j=1}^J$. Using the pairs $\{(V(A_j^E, s_j), s_j)\}$ as data, we compute b_K and B_K by multivariate least squares. We repeat until the b_K and B_K stabilize. We have found that approximately twenty iterations suffice for three firms and thirty for four firms.⁴⁹ The easiest way to get a cluster of points $\{s_j\}$ about a centroid is to use abscissae from the quadrature rule described above with s set to K and A set to zero. However, one must jiggle the points so that no two firms have exactly the same cost. Of importance in reducing computational effort is to avoid recomputing the payoff (equation (9)) when checking equilibrium condition (10). Our strategy is to (temporarily) store payoff vectors indexed by A and check for previously computed payoffs before computing new ones in checking condition (10).

A.2 Estimation Details

In this subsection we provide additional details about the estimation procedure that were not provided in the main paper. We begin by showing the log likelihood surface plotted on a fine grid in Figure 4 and on a coarse grid in Figure 5 for the three firm model. Figures 6 and 7 are for the four firm model. The endpoints of the horizontal axes are tenth of a standard deviation to the left and right of the maximum in Figure 4 and 24 standard deviations to the right and left in Figure 5. For Figures 6 and 7 they are a tenth and 48. These are profile likelihoods; i.e., in each panel the indicated parameter is moved and all others are fixed at the values that maximize the likelihood.

As seen from Figures 4 and 6, the surface is, basically, a step function so that curvature at the maximum will not provide a reliable basis for inference. The reason, of course, is that small changes in the parameters do not cause the decisions of the firms to change. In this situation, accurate frequentist inference would be difficult and would be prohibitively computationally intensive if bootstrapping were involved. On the other hand, Bayesian inference in this situation is conceptually straightforward and computationally feasible.

Figure 4 about here

⁴⁸The set of centroids that actually get visited in any MCMC repetition is about the same for grid increments ranging from 4 to 16 times the standard deviation of the state variables in our data. For a three firm game the number of hyper-cubes that actually are visited in any one repetition is about six.

⁴⁹An alternative is to apply a modified Howard acceleration strategy as described in Kuhn (2006); see also Rust (1996) and Howard (1960). The idea is simple: The solution $\{A_t^E\}$ of the game with payoffs (9) will not change much, if at all, for small changes in the value function $V(s)$. Therefore, rather than recompute the solution at every step of the (b_K, B_k) iterations, one can reuse a solution for a few steps.

Figure 5 about here

Figure 6 about here

Figure 7 about here

However, Figures 5 and 7 do suggest that implementing a Bayesian strategy that explores the surface well will be a challenge. They also suggest that the standard deviations of the posterior will be extremely tight. The horizontal line is at three orders of magnitude below the maximum. If an MCMC chain is near the maximum, the chance that it will move to a point below the horizontal line is less than 0.001.

An MCMC chain that uses a move-one-at-a-time random walk proposal density will usually do a good job of exploring surfaces such as seen in Figures 5 and 7; see Gamerman and Lopes (2006). However this comes at a cost because an MCMC chain that uses a move-one-at-a-time random walk proposal strategy is usually inefficient relative to those that use other proposal strategies. The vertical lines in Figures 5 and 7 indicate the range of the MCMC chain's excursions after the transient elements of the chain have died out.

We implement our computational algorithm using code that is in the public domain and available at <http://www.aronaldg.org/webfiles/emm/>. This code is based on Chernozhukov and Hong (2003). Full details regarding the proposal density and other conventions are in the User's Guide distributed with the code. One needs enough draws to accurately compute averages such as standard deviations, histograms, and other characteristics of the posterior distribution. Our chains are highly correlated so that very long chains with a stride (sampling rate) of 375 are required to break the dependence. As explained in the User's Guide, computations can be accelerated if the values of θ visited by the chain are restricted to (fractional) powers of two. We impose this restriction on the chain.

Histograms of the marginal posterior distributions are displayed in Figures 8 and 9 for the three and four firm cases, respectively.

Figure 8 about here

Figure 9 about here

Figures 10, 11, and 12 replicate Figures 1, 2, and 3 for the fully rational model, respectively.

Figure 10 about here

Figure 11 about here

Figure 12 about here

Appendix References

- ABRAMOWITZ, M., AND I. A. STEGUN (1964): *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover.
- CHERNOZHUKOV, V., AND H. HONG (2003): “A MCMC Approach to Classical Estimation,” *Journal of Econometrics*, 115(2), 293–346.
- DUTTA, P. K., AND R. K. SUNDARAM (1998): “The Equilibrium Existence Problem in General Markovian Games,” in *Organizations with Incomplete Information*, ed. by M. Majumdar, pp. 159–207. Cambridge University Press.
- GAMERMAN, D., AND H. F. LOPES (2006): *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference, 2nd Edition*. Chapman & Hall.
- GOLUB, G. H. (1973): “Some modified matrix eigenvalue problems,” *SIAM Review*, 15, 318–334.
- GOLUB, G. H., AND J. H. WELSCH (1969): “Calculation of Gaussian quadrature rules,” *Mathematics of Computation*, 23, 221–230.
- HOWARD, R. A. (1960): *Dynamic Programming and Markov Processes*. Wiley.
- KUHN, M. (2006): “Notes on Numerical Dynamic Programming in Economic Applications,” CDSEM, University of Mannheim, Denmark.
- PARTHASARATHY, T. (1973): “Discounted, Positive, and Noncooperative Stochastic Games,” *International Journal of Game Theory*, 2.
- RUST, J. (1996): “Numerical dynamic programming in economics,” *Handbook of computational economics*, pp. 619–729.

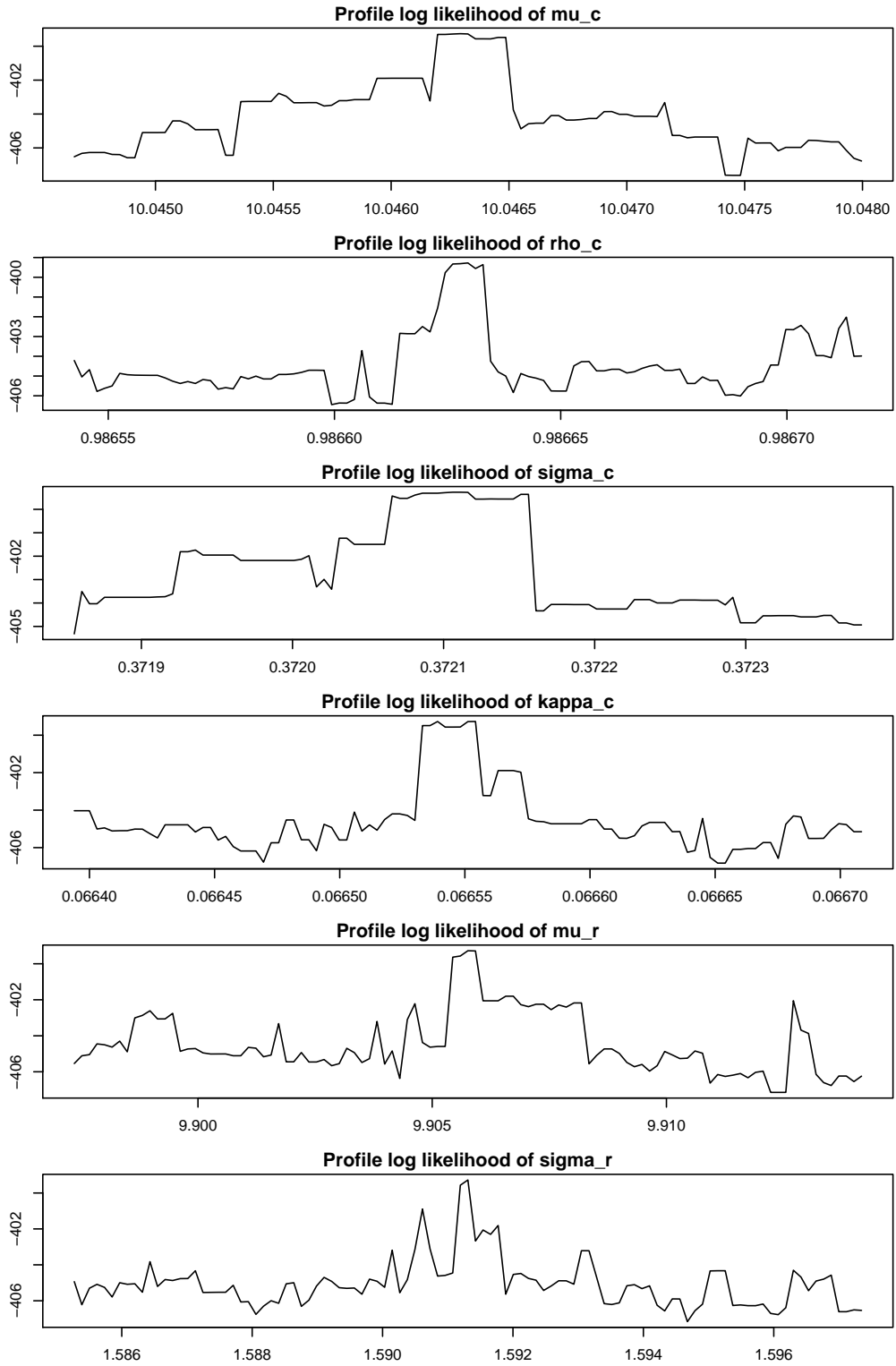


Figure 4. Profile Log Likelihood, Three Firm Model. Shown is the logarithm of the profile likelihood plotted for a tenth of the posterior standard deviation to the left and right of the maximum of the likelihood.

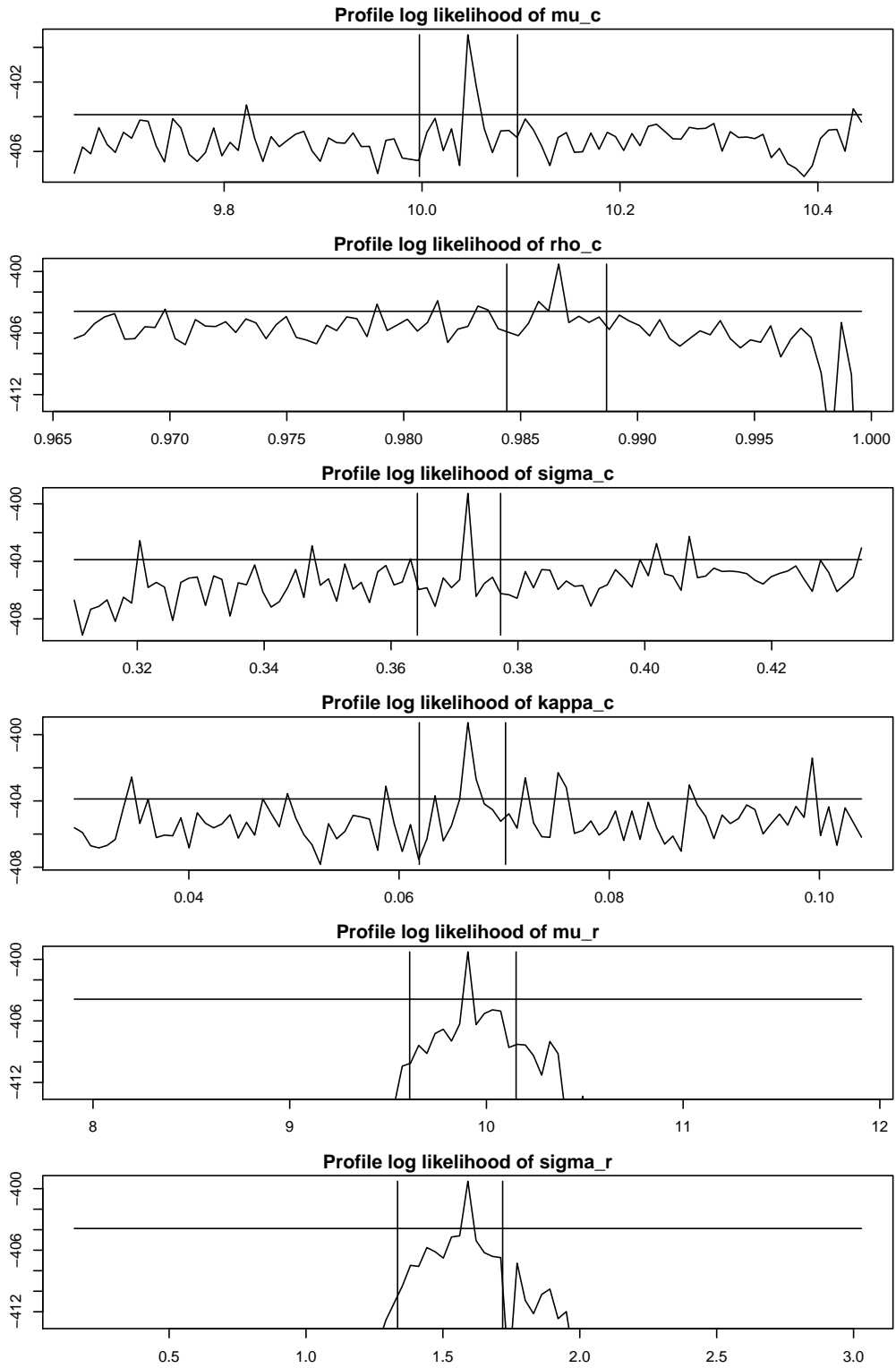


Figure 5. Profile Log Likelihood, Three Firm Model. Shown is the logarithm of the profile likelihood plotted for 24 posterior standard deviations to the left and right of the maximum of the likelihood. Points that violate support conditions and points below 10^{-6} of the maximum of the likelihood are not plotted. The horizontal line is at 10^{-3} of the maximum. The vertical lines indicate the range of the MCMC chain's excursions after the transient elements of the chain have died out.

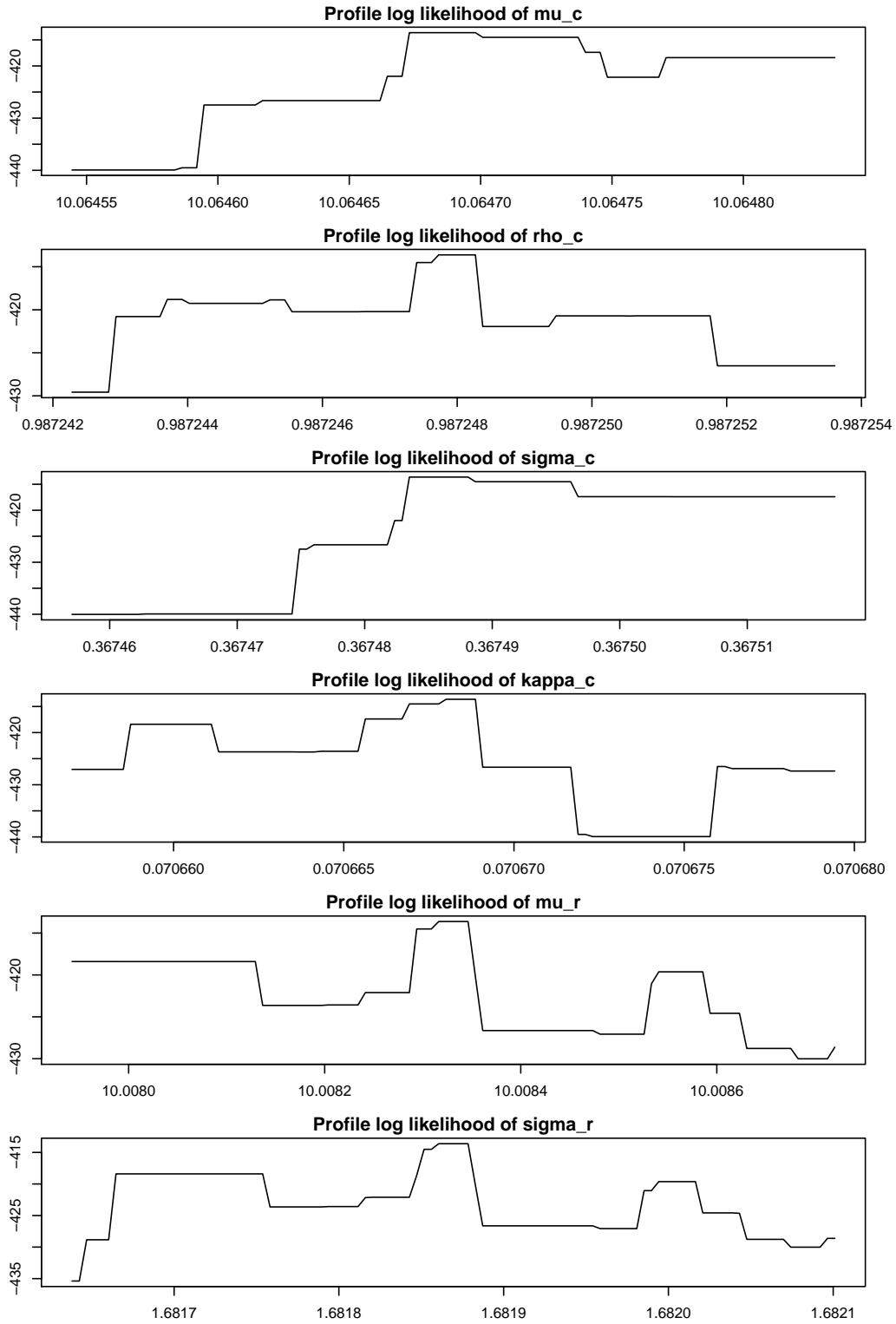


Figure 6. Profile Log Likelihood, Four Firm Model. Shown is the logarithm of the profile likelihood plotted for a tenth of a posterior standard deviations to the left and right of the maximum of the likelihood.

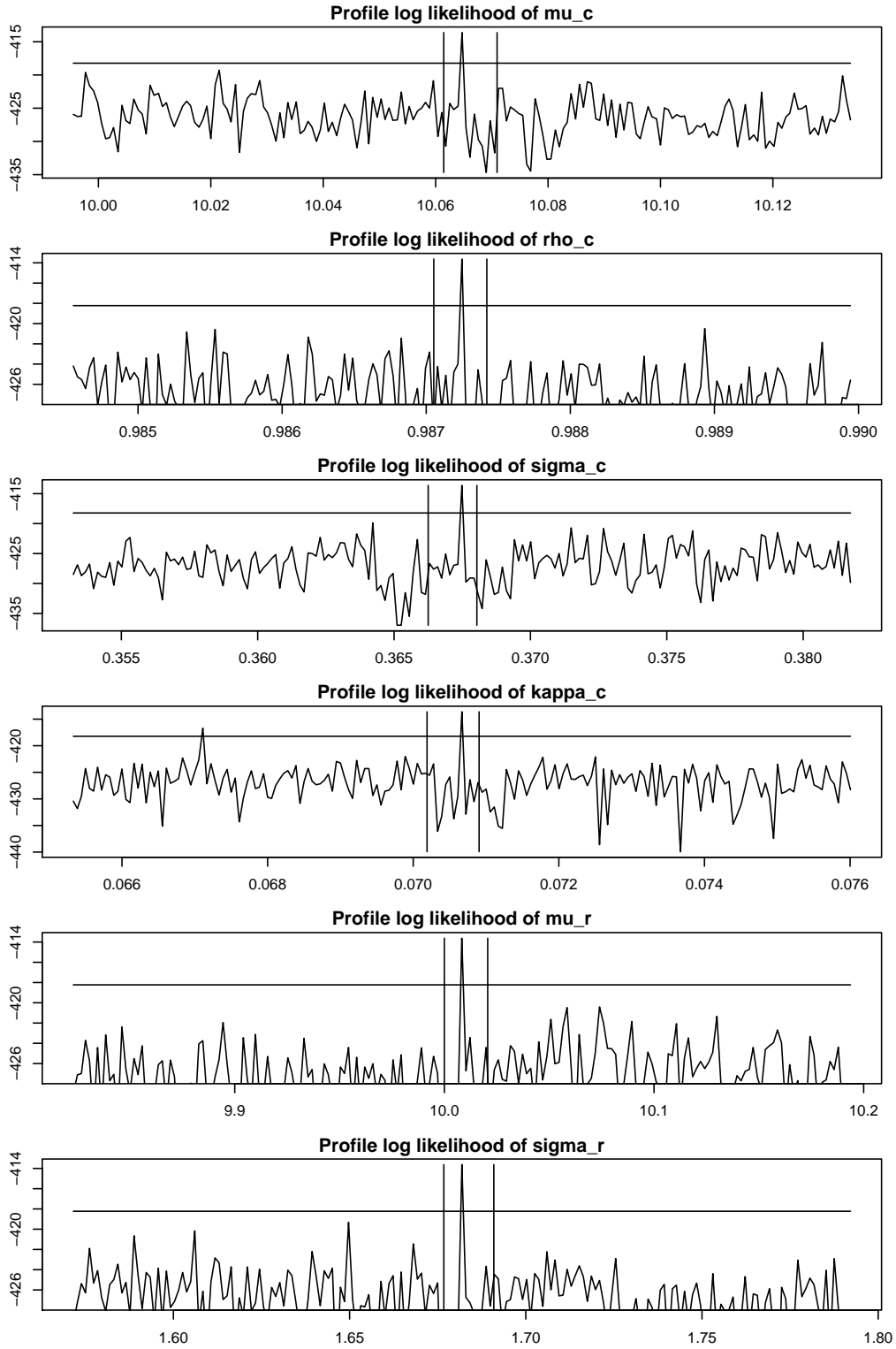


Figure 7. Profile Log Likelihood, Four Firm Model. Shown is the logarithm of the profile likelihood plotted for 48 posterior standard deviations to the left and right of the maximum of the likelihood. Points that violate support conditions and points below 10^{-6} of the maximum of the likelihood are not plotted. The horizontal line is at 10^{-3} of the maximum. The vertical lines indicate the range of the MCMC chain's excursions after the transient elements of the chain have died out.

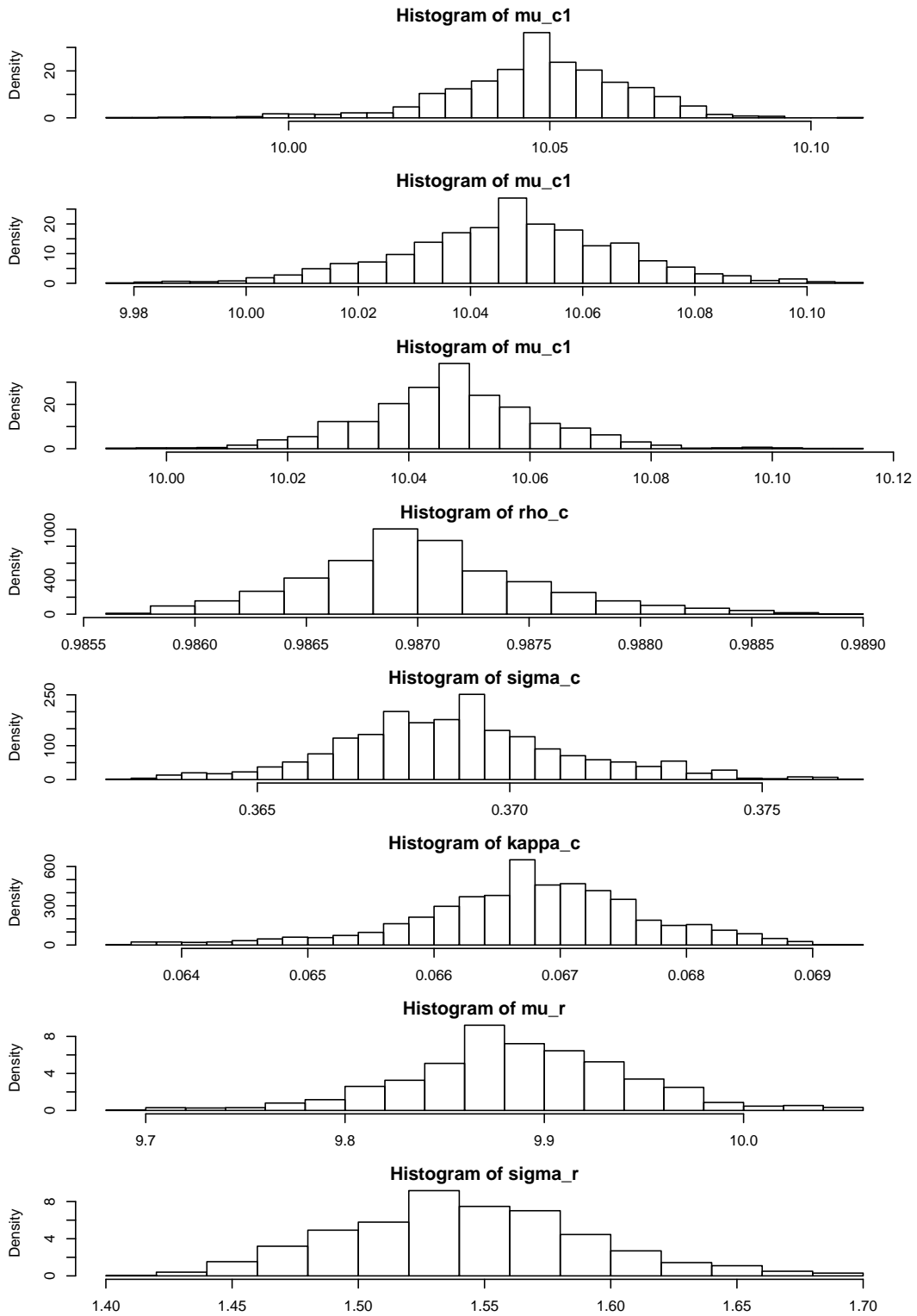


Figure 8. Marginal Posterior Distributions, Three Firm Model. Shown are histograms constructed from an MCMC chain for the three firm model with 3,000,000 repetitions at a stride of 375 for 8000 net. The salient feature of this graphic is the contrast of the histogram for the parameter κ_c compared to that shown in Figure 9.

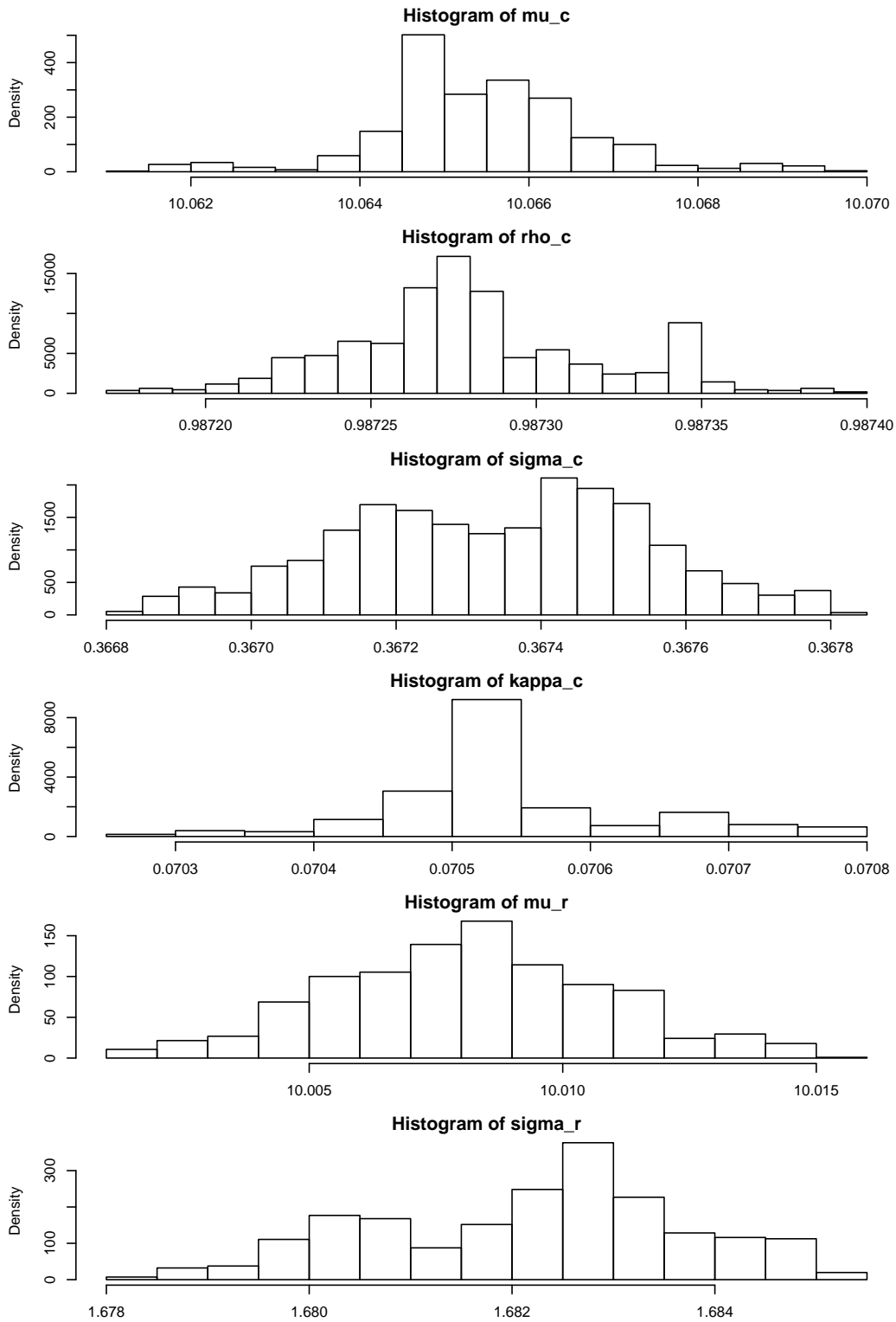


Figure 9. Marginal Posterior Distributions, Four Firm Model. Shown are histograms constructed from an MCMC chain for the four firm model with 3,000,000 repetitions at a stride of 375 for 8000 net. The salient feature of this graphic is the contrast of the histogram for the parameter κ_c compared to that shown in Figure 8.

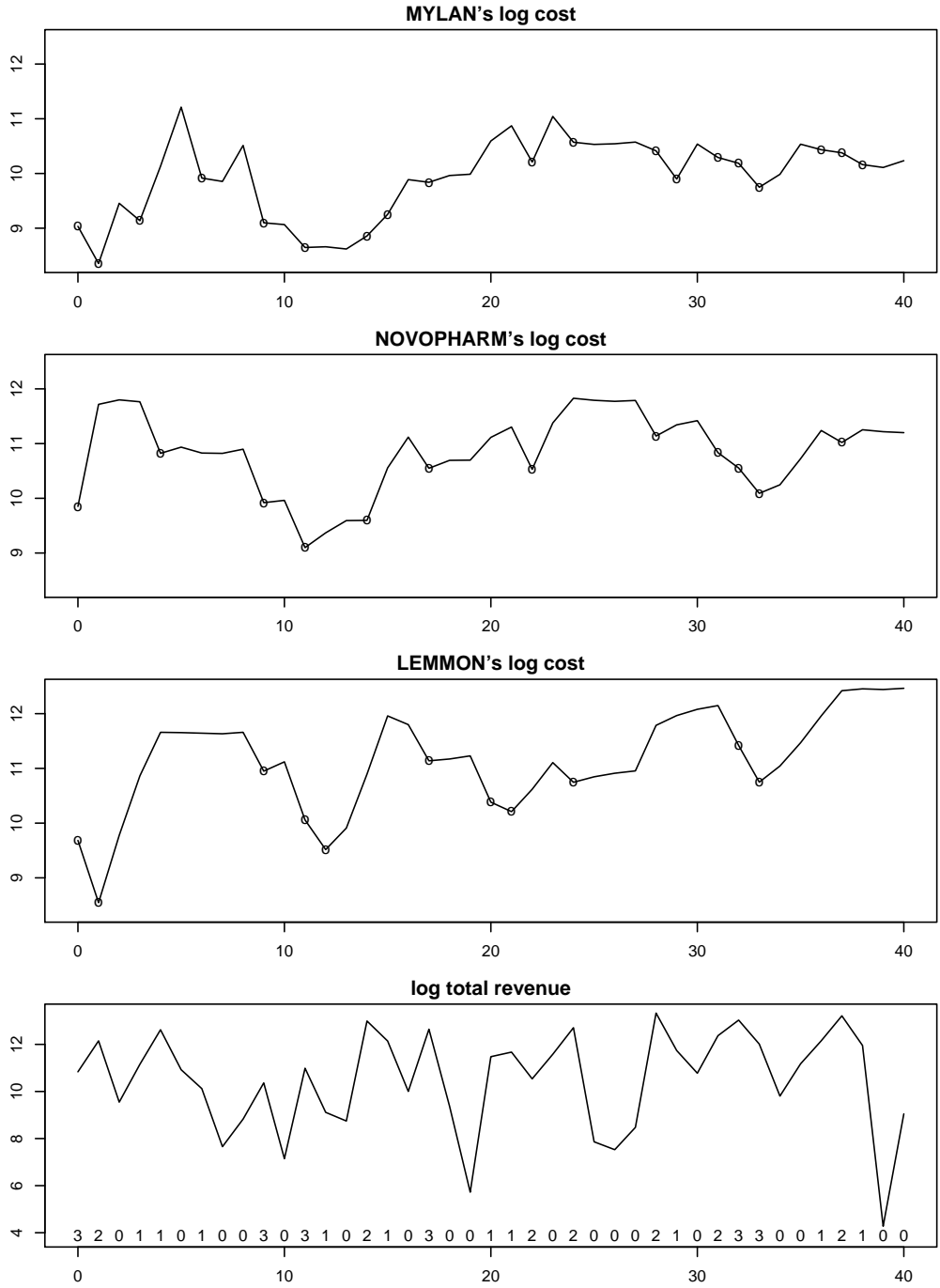


Figure 10. Cost, Revenue, and Entry Decisions. Plotted as a solid line in the first three panels is the logarithm of cost for the three dominant firms in the boundedly rational three firm model. The logarithm of cost is computed by averaging at Step 2e of the importance sampler at the maximum likelihood estimate. The circles in these plots indicate that the firm entered the market at that time point. The bottom panel shows the logarithm of total revenue. The numbers at the bottom are the count of the number of dominant firms who entered the market at that time point.

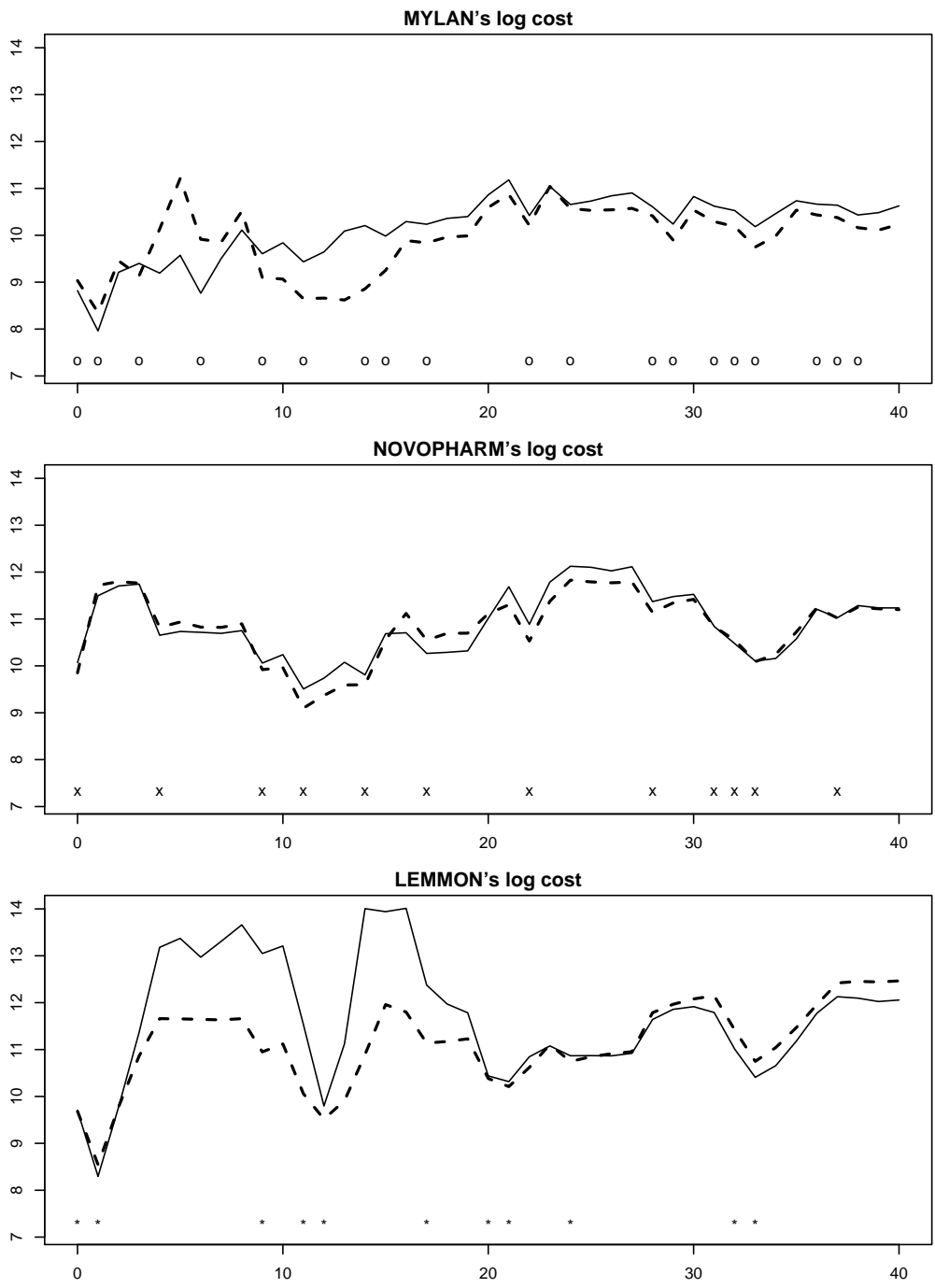


Figure 11. Cost and Entry Decisions of the Dominant Firms. Plotted is the logarithm of cost for the three dominant firms. The dashed line is under the boundedly rational three firm model, and the solid under the four firm model. The circles indicate the markets that Mylan entered, crosses the same for Novopharm, and the asterisks for Lemmon. The logarithm of cost as described in the legend of Figure 10.

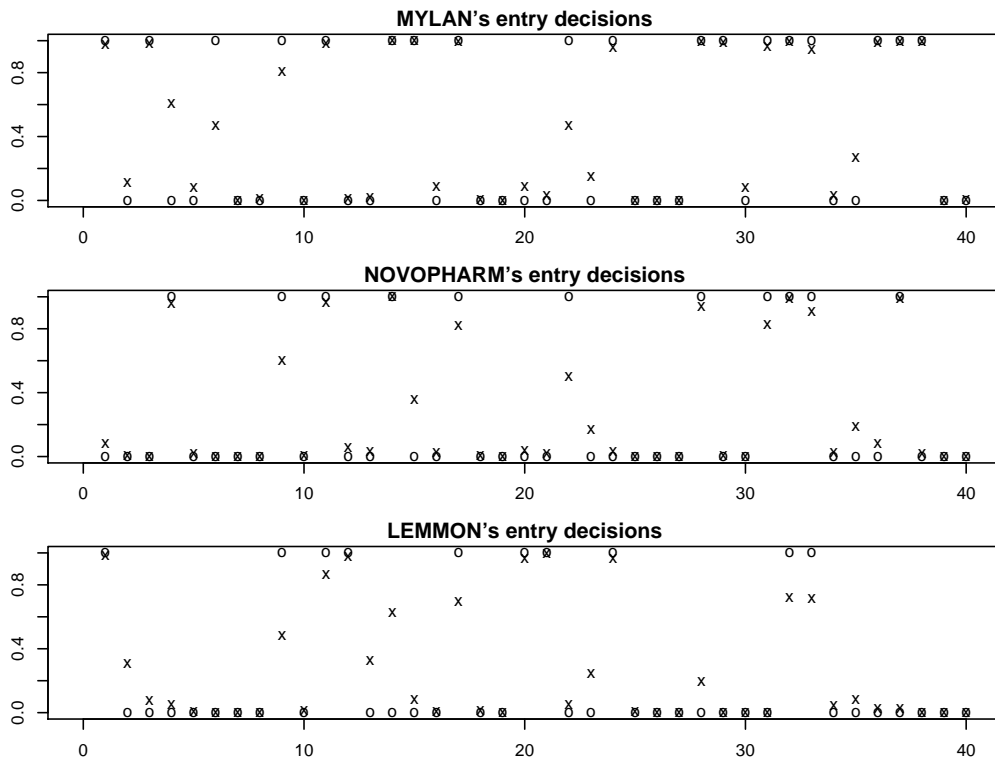


Figure 12. Actual and Predicted Entry Decisions. Plotted as circles are the entry decisions of the three dominant firms in the boundedly rational three firm model. The crosses are the average predictions of the three firm model computed by averaging game solutions at Step 2e of the importance sampler at the maximum likelihood estimate.