Written Exam at the Department of Economics
Winter 2019–20

Advanced Microeconometrics

Final Exam

— Suggested Answers —

Problem 1

Consider the following censored regression model, for a sample of individuals $i = 1, \dots, N$:

$$y_i = \max\{0, y_i^\star\}, \qquad y_i^\star = x_i'\beta + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}\left(0, \sigma^2\right), \tag{1}$$

where the explanatory variables are contained in the $K \times 1$ vector x_i, and are related to the latent variable $y_i^\star$ through the vector of regression coefficients β.

Question 1.1: Discuss *briefly* the identification of the model (without any derivations). In particular, explain if σ^2 is identified, and why.

Suggested answer

Given the functional form assumptions, both β and σ^2 are separately identified due to the observed continuous variation in y_i when $y_i^\star > 0$. Similar to Probit we have that $Pr(y_i^\star > 0|x_i) = \Phi(x_i'\beta/\sigma)$ and $Pr(y_i^\star \leq 0|x_i) = 1 - \Phi(x_i'\beta/\sigma)$ and hence the observed fraction of censored/uncensored conditional on x_i identifies β relative to σ. Note that this is all we can hope to identify in the Probit model since we only observe a binary indicator whether $y_i^\star > 0$ or not. However, in the censored regression model we do observe the latent variable $y_i^\star$ for the uncensored observations (i.e. when $y_i^\star > 0$) allowing us to identify σ separately from β. Specifically, the density of observed variable y_i given x_i is

$$f(y_i|x_i) = [1 - \Phi(x_i'\beta/\sigma)]^{1(y_i=0)} \left[(1/\sigma)\phi[(y_i - x_i'\beta)/\sigma]\right]^{1(y_i>0)}$$

which clearly depends on σ separately from β.

Given the structure of the model and identification of σ and β we have fully specified the the entire conditional distribution of both y_i and $y_i^\star$. However, identification as well as the consistency of the maximum likelihood estimator derived from the Tobit model hinges crucially on distributional assumptions made here (such as normality and conditional independence of ε_i given x_i). Violation of these assumptions generally leads to inconsistent maximum likelihood estimates. In contrast, non-normality and heterosceddstisty does not affect identification and consistent estimation of parameters in the linear regression model without censoring. So censoring is costly.

We have also assumed that the truncation point is fixed at zero, but if for example $y_i = \max\{x_i'\gamma, x_i'\beta + \varepsilon_i\}$ we would only be able to identify $\beta - \gamma$ since this is observationally equivalent model with a fixed truncation point 0 and the latent variable $y_i^\star = x_i'(\beta - \gamma) + \varepsilon_i$

Question 1.2: Show that the conditional expectation of the observed outcome is

$$\mathrm{E}[y_i \mid x_i] = x_i'\beta\,\Phi\!\left(\frac{x_i'\beta}{\sigma}\right) + \sigma\,\phi\!\left(\frac{x_i'\beta}{\sigma}\right), \tag{2}$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ denote, respectively, the cumulative distribution function(CDF) and the probability density function (PDF) of the standard normal distribution $\mathcal{N}(0,1)$.

Hint 1: To get started, remember that $\mathrm{E}[y_i \mid x_i] =$ $\mathrm{E}[y_i \mid x_i, y_i = 0]\Pr(y_i = 0 \mid x_i) + \mathrm{E}[y_i \mid x_i, y_i > 0]\Pr(y_i > 0 \mid x_i)$.

Hint 2: If $z \sim \mathcal{N}(0,1)$, then for any constant $\alpha \in \mathbb{R}$ it holds that $\mathrm{E}[z \mid z > \alpha] = \phi(\alpha)/[1 - \Phi(\alpha)]$.

Suggested answer

By the law of iterated expectations we can write

$$\mathrm{E}[y_i \mid x_i] = \mathrm{E}[y_i \mid x_i, y_i = 0]\Pr(y_i = 0 \mid x_i) + \mathrm{E}[y_i \mid x_i, y_i > 0]\Pr(y_i > 0 \mid x_i)\,.$$

Clearly $\mathrm{E}[y_i \mid x_i, y_i = 0] = 0$ and $y_i = y_i^\star$ conditional on $y_i > 0$, so we have

$$\begin{aligned}
\mathrm{E}[y_i \mid x_i] &= 0 + \mathrm{E}[y_i \mid x_i, y_i > 0] \Pr(y_i > 0 \mid x_i) \\
&= \mathrm{E}[y_i^\star \mid x_i, y_i^\star > 0] \Pr(y_i^\star > 0 \mid x_i) \\
&= \mathrm{E}[x_i'\beta + \varepsilon_i \mid x_i, x_i'\beta + \varepsilon_i > 0] \Pr(x_i'\beta + \varepsilon_i > 0 \mid x_i)
\end{aligned}$$

where $\varepsilon_i \mid x_i \sim \mathcal{N}(0, \sigma^2)$ implies that $\Pr(x_i'\beta + \varepsilon_i > 0 \mid x_i) = \Phi(x_i'\beta/\sigma)$ and that ε_i is independent of x_i. Independence implies that we can remove the conditioning on x_i in the conditional expectation of ε_i

$$\begin{aligned}
\mathrm{E}[y_i \mid x_i] &= \big(x_i'\beta + \mathrm{E}[\varepsilon_i \mid x_i'\beta + \varepsilon_i > 0]\,\big)\Phi(x_i'\beta/\sigma) \\
&= x_i'\beta\Phi(x_i'\beta/\sigma) + \sigma\mathrm{E}[\varepsilon_i/\sigma \mid \varepsilon_i/\sigma > -x_i'\beta/\sigma]\,\Phi(x_i'\beta/\sigma)
\end{aligned}$$

where $\mathrm{E}[\varepsilon_i/\sigma \mid \varepsilon_i/\sigma > -x_i'\beta/\sigma]$ is the mean of a truncated standard normal distribution with truncation point $-x_i'\beta/\sigma$. It therefore holds that

$$\begin{aligned}
\mathrm{E}[\varepsilon_i/\sigma \mid \varepsilon_i/\sigma > -x_i'\beta/\sigma] &= \phi(-x_i'\beta/\sigma)/[1 - \Phi(-x_i'\beta/\sigma)] \\
&= \phi(x_i'\beta/\sigma)/\Phi(x_i'\beta/\sigma)
\end{aligned}$$

where the last equality follows from the symmetry of the standard normal distribution.

We then have

$$\mathrm{E}[y_i \mid x_i] = x_i'\beta\,\Phi\Big(\frac{x_i'\beta}{\sigma}\Big) + \sigma\,\phi\Big(\frac{x_i'\beta}{\sigma}\Big)$$

Question 1.3: One of your colleagues suggests you construct an estimator of $\theta = (\beta', \sigma^2)'$ based on the following optimization problem:

$$\widehat{\theta} = \arg\min_{\theta} \left[\frac{1}{N}\sum_{i=1}^{N} \widehat{m}(y_i, x_i; \theta, u_{iM})\right]^2 \tag{3}$$

where $\widehat{m}(y_i, x_i; \theta, u_{iM})$ is simulated using a sample of M random draws $u_{iM} = \{u_i^{(1)}, \ldots, u_i^{(M)}\}$ from the standard normal distribution, for each $i = 1, \ldots, N$.

Describe the principle of the estimation method your colleague is referring to. As part of your answer, you are expected to provide and justify a possible expression of $\widehat{m}(y_i, x_i; \theta, u_{iM})$ *[hint: you may or may not use the result in Eq.* (2) *to do this]*, and to outline the steps of the corresponding estimation approach.

Suggested answer

The principle of the estimation method the colleague is referring to is the *Method of Simulated Moments* (MSM) estimator.

A possible choice would be to rely on the moment condition

$$\mathrm{E}[m(y_i, x_i; \theta)] = \mathrm{E}[y_i - g(x_i; \theta)] = 0$$

where $g(x_i; \theta)$ is an expression for the conditional mean $\mathrm{E}[y_i \mid x_i]$ derived from the model.

Suppose first that the expression for $\mathrm{E}[y_i \mid x_i]$ in Eq. (2) is unknown, we could simulate data from the model to obtain

$$\widehat{g}(x_i; \theta, u_{iM}) = 1/M \sum_{m=1}^{M} \max\{0, x_i'\beta + \sigma u_i^{(m)}\}$$

so that the expression for $\widehat{m}(y_i, x_i; \theta, u_{iM})$ becomes

$$\widehat{m}(y_i, x_i; \theta, u_{iM}) = y_i - 1/M \sum_{m=1}^{M} \max\{0, x_i'\beta + \sigma u_i^{(m)}\}$$

where we note that $\widehat{m}(y_i, x_i; \theta, u_{iM})$ is an unbiased simulator for $m(y_i, x_i; \theta)$ in the sense that $\mathrm{E}[\widehat{m}(y_i, x_i; \theta, u_{iM})] = \mathrm{E}[m(y_i, x_i; \theta)]$.

We would then solve the minimization problem in Eq. (3) holding fixed the simulation draws for each evaluation of the objective function as we search over the parameter space.

A potential challenge with this particular choice of simulated moment is that the simulator is not smooth since the max operator introduces a kink in the truncation point. This makes the problem locally non-differentiable with a finite number of observations and simulation draws. We can ameliorate these problems by using gradient free optimization routines such as Nelder-Mead, or by introducing artificial smoothness by replacing the max operator by it's logit smoothed version (the so called "log-sum formula").

The problem is reduced in larger samples and for large values of M.

Alternatively, we could simply estimate θ using Method of Moments (MM) or Nonlinear Least Squares (NLS) since the nonlinear regression function $\mathrm{E}[y_i \mid x_i] = g(y_i, x_i; \theta)$ in Eq. (2) is available in closed form.

$$g(y_i, x_i; \theta) = x_i'\beta\, \Phi\left(\frac{x_i'\beta}{\sigma}\right) - \sigma\, \phi\left(\frac{x_i'\beta}{\sigma}\right)$$

so that

$$\widehat{\theta_{MM}} = \arg\min_{\theta} \left[\frac{1}{N}\sum_{i=1}^{N} y_i - g(y_i, x_i; \theta)\right]^2$$

or

$$\widehat{\theta_{NLS}} = \arg\min_{\theta} \frac{1}{N}\sum_{i=1}^{N} [y_i - g(y_i, x_i; \theta)]^2$$

In the Tobit example with normally distributed independent errors there is no reason to use simulation to approximate the moment condition since we express $m(y_i, x_i; \theta)$ in closed form. However, MSM easily gives the flexibility of choosing other moments that does not rely on the distributional assumptions or allow for a more flexible specification of the distribution of the error term.

Question 1.4: How do you recommend to choose the number of random draws M in Question 1.3? In particular, explain how this number affects the bias of the estimator *(no derivations required).*

Suggested answer

Given that $\widehat{m}(y_i, x_i; \theta, u_{iM})$ is an unbiased simulator for $m(y_i, x_i; \theta)$ the MSM estimator is asymptotically equivalent to the MM estimator as M increase without bound. However, the MSM estimator has the remarkable property of being consistent even for $M = 1$. While there is an efficiency loss of finite M because of simulation noise, it disappears as $M \to \infty$. In the special case of a frequency simulator the variance is inflated by the factor $(1+1/M)$, so that $V_{y,u}(\widehat{m}(\theta)) = (1+1/M)V_y(m(\theta))$. Hence, in larger samples simulation variance is also expected to be smaller. The only cost of increasing M is computational, so the choice of M is really a tradeoff between patience and computational power and the overall (simulation inflated) variance.

Question 1.5: How would you modify the optimization problem in Eq. (3) to improve the efficiency of the estimator $\widehat{\theta}$? Describe *briefly* the corresponding approach.

Suggested answer

The most efficient estimator is obviously the Maximum Likelihood estimator (MLE) which is readily available given the model assumptions. But there are several other ways of improving efficiency if you prefer a moment based estimator and are not willing to impose the necessary distributional assumptions or because MLE is intractable. One improvement could for example be to use importance sampling or variance reductions techniques such as antithetics, Halton sequences or Sobold draws to reduce simulation noise for MSM.

Here we focus on how we can improve efficiency by including more moment conditions and weight them optimally. Hence, we may consider the estimator

$$\widehat{\theta} = \arg\min_{\theta} \left[\frac{1}{N}\sum_{i=1}^{N} \mathbf{z}_i \widehat{m}(y_i, x_i; \theta, u_{iM})\right]' W_N \left[\frac{1}{N}\sum_{i=1}^{N} \mathbf{z}_i \widehat{m}(y_i, x_i; \theta, u_{iM})\right]$$

where z_i is a r dimensional vector of instruments and W_N is a $(r \times r)$ symmetric positive definite weighting matrix. W_N is possibly stochastic with finite probability limit and does not depend on θ and the subscript N on W_N is used to indicate that its value may depend on the sample. Different choices of weighting matrix W_N lead to different estimators that, although consistent, have different variances if the number of moment restrictions r, exceeds the number of parameters, q. A simple choice is to let W_N be the identity matrix. However, the optimal GMM estimator weights the moments with the inverse of the variance matrix of the sample moment conditions. Intuitively, this makes a lot of sense since we would like to put more emphasis on moments that are more precisely estimated (i.e. has lower variance).

Problem 2

Consider the following two-parameter model

$$y \sim \mathcal{N}(\theta_1 + \theta_2, 1)\,, \tag{4}$$

with prior distributions $\theta_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $\theta_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$.

Question 2.1: Derive the conditional distributions $p(\theta_1 \mid \theta_2, y)$ and $p(\theta_2 \mid \theta_1, y)$.

Hint: Given the symmetry of the problem, you need to do the derivations only once.

Suggested answer

Prior distribution:

$$p(\theta_1) \propto \exp\left\{-\frac{1}{2\sigma_1^2}(\theta_1 - \mu_1)^2\right\}$$

Likelihood:

$$p(y \mid \theta_1, \theta_2) \propto \exp\left\{-\frac{1}{2}(y - \theta_1 - \theta_2)^2\right\}$$

Posterior, applying Bayes' theorem:

$$\begin{aligned} p(\theta_1 \mid \theta_2, y) &\propto p(y \mid \theta_1, \theta_2)p(\theta_1), \\ &\propto \exp\left\{-\frac{1}{2}(y - \theta_1 - \theta_2)^2\right\} \exp\left\{-\frac{1}{2\sigma_1^2}(\theta_1 - \mu_1)^2\right\}, \\ &\propto \exp\left\{-\frac{1}{2}\left[\theta_1^2\left(1 + \frac{1}{\sigma_1^2}\right) - 2\theta_1\left(y - \theta_2 + \frac{\mu_1}{\sigma_1^2}\right)\right]\right\}, \end{aligned}$$

which is the kernel of the following normal distribution:

$$\theta_1 \mid \theta_2, y \sim \mathcal{N}\left(\left(1 + \frac{1}{\sigma_1^2}\right)^{-1}\left(y - \theta_2 + \frac{\mu_1}{\sigma_1^2}\right), \left(1 + \frac{1}{\sigma_1^2}\right)^{-1}\right),$$

or, equivalently:

$$\theta_1 \mid \theta_2, y \sim \mathcal{N}\left(\frac{\mu_1 + \sigma_1^2(y - \theta_2)}{1 + \sigma_1^2}, \frac{\sigma_1^2}{1 + \sigma_1^2}\right).$$

Similarly, due to the symmetry of the problem we obtain

$$\theta_2 \mid \theta_1, y \sim \mathcal{N}\left(\frac{\mu_2 + \sigma_2^2(y - \theta_1)}{1 + \sigma_2^2}, \frac{\sigma_2^2}{1 + \sigma_2^2}\right).$$

Question 2.2: Outline the different steps of a Gibbs sampler that can be designed to produce random draws from the posterior distribution of θ_1 and θ_2. Be as precise as possible.

Suggested answer

Set a starting value $\theta_2^{(0)}$, either fixed to a given value or sampled from the prior (note that θ_1 does not need to be initialized, as it is updated first in the Gibbs sampler).

Repeat the following two steps, for each MCMC iteration $t = 1, \ldots, T$, and until practical convergence of the sampler:

1) Sample $\theta_1^{(t)}$ from $p(\theta_1 \mid y, \theta_2^{(t-1)})$.
2) Sample $\theta_2^{(t)}$ from $p(\theta_2 \mid y, \theta_1^{(t)})$.

Using the conditional distributions derived in Question 2.2.

Question 2.3: Assuming we observe $y = 4$ and we set $\mu_1 = \mu_2 = 50$ and $\sigma_1^2 = \sigma_2^2 = 100$, we run the Gibbs sampler derived in Question 2.2 for 1,000 iterations. The corresponding trace plots of the two parameters θ_1 and θ_2, as well as the trace of their sum $\theta_1 + \theta_2$, are shown in Fig. 2.1.

Does the algorithm converge in any sense? Comment on the trace plots and explain the results, both intuitively and formally.

Suggested answer

The Gibbs sampler does not converge for θ_1 nor for θ_2, but does for the sum $\theta_1 + \theta_2$. This is because only the mean of the normal distribution specified for y in Eq. (4) is identified, but not the individual parameters

θ_1 and θ_2 — it is possible to transform these parameters as $\widetilde{\theta}_1 = \theta_1 + c$ and $\widetilde{\theta}_2 = \theta_2 - c$, for any constant $c \in \mathbb{R}$, without changing the mean of the normal distribution, i.e., without changing the likelihood. This lack of identification translates into a lack of convergence of the Gibbs sampler for the corresponding two parameters taken separately. Their sum, however, is not affected by this problem, as it is identified.

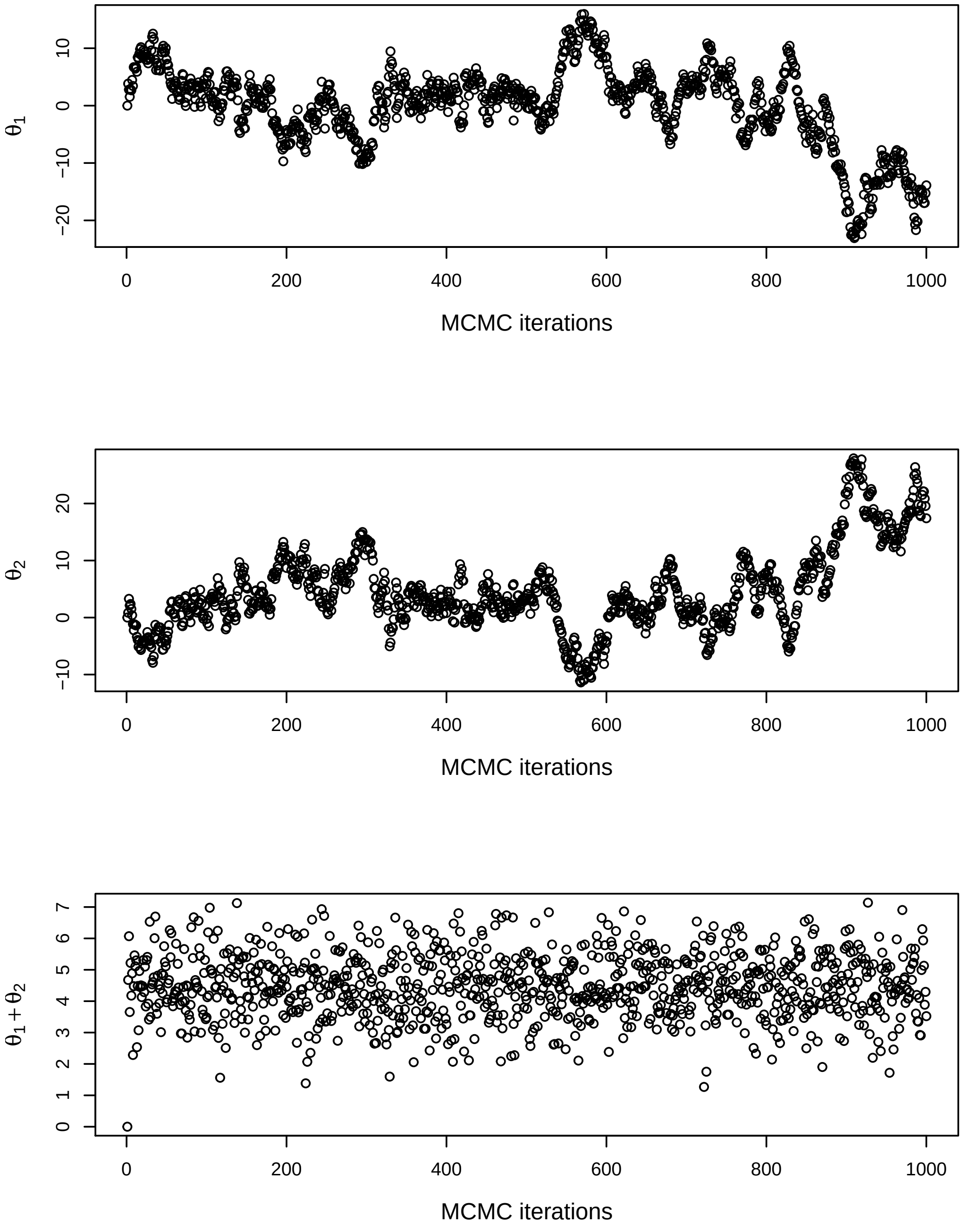

Figure 2.1: Trace plots of the Gibbs sampler for the parameters θ_1, θ_2, and their sum $\theta_1 + \theta_2$.

Problem 3

Consider the following `MATLAB` functions:

```
function [x] = simul1(n, fun, a, b)
    z = betarnd(a, b, n, 1);
    x = mean(fun(z));
end

function [x] = simul2(n, fun, a, b)
    z = rand(n, 1);
    x = mean(fun(z) .* betapdf(z, a, b));
end
```

and the following piece of code:

```
rng(123);
h = @(x) (x - 3).^2;
n = 10000;
fprintf('simul1 output = %6.4f\n', simul1(n, h, 2, 3));
fprintf('simul2 output = %6.4f\n', simul2(n, h, 2, 3));
```

which produces the following output:

```
simul1 output = 6.7954
simul2 output = 6.7502
```

Question 3.1: Express in mathematical terms what these two functions do. You should just provide a few equations to answer this question. Be explicit about the notation.

[Note: The `MATLAB` function `betarnd(a, b, m, n)` produces a $m \times n$ matrix of random draws from the Beta distribution with shape parameters a and b, while the function `betapdf(z, a, b)` returns the probability density function of the corresponding Beta distribution evaluated at each entry of z.]

Suggested answer

The first function `x=simul1(n, fun, a, b)` returns

$$x = 1/n \sum_{i=1}^{n} h(z_i)$$
$$\text{where } z_i \sim Beta(a, b)$$

That is `simul1`, first takes n independent random numbers from the Beta distribution with parameters specified by the inputs a and b and saves them in to the $n \times 1$ vector $z = (z_1, \ldots, z_n)'$. It then evaluate the sample average of a function, $h(z_i)$ over these draws. The input argument `fun` is a function handle that points to a vector function $h(x)$ that for each value z_i computes the function $h(z_i)$.

For example `simul1(10000, @(x) (x - 3).^2, 2, 3)` returns

$$x = 1/10000 \sum_{i=1}^{10000} (z_i - 3)^2$$
$$\text{where } z_i \sim Beta(2, 3)$$

The second function `simul2(n, fun, a, b)` returns

$$x = 1/n \sum_{i=1}^{n} f(z_i; a, b) h(z_i)$$
$$\text{where } z_i \sim U(0, 1)$$

where now $z_1, \ldots, z_n$ are n are independent random numbers from the uniform distribution on the unit interval and $f(z_i; a, b)$ is the pfd of the Beta distribution with parameters a and b. The inputs are the same as above.

Question 3.2: Explain precisely the two approaches implemented by the functions `simul1()` and `simul2()`, and why the corresponding results look similar.

Suggested answer

For large n, both these Matlab functions approximate the mean of a specified function $h(z)$ of beta distributed random variables, i.e. $\mathrm{E}[h(z)]$ for $z \sim Beta(a,b)$.

The first MATLAB function `simul1()` implements a simulator that approximates $\mathrm{E}[h(z)]$ by direct Monte Carlo integration

$$x_{Simul1} = 1/n \sum_{i=1}^{n} h(z_i) \xrightarrow[n\to\infty]{} \mathrm{E}[h(z)] = \int_0^1 h(z) f(z;a,b) dz$$

where $z_1, \ldots, z_n$ are n are independent random numbers from the Beta distribution with parameters a and b, density $f(z;a,b)$ and bounded support on the unit interval $[0,1]$. Here convergence of the average to it's expected mean is a simple application of the law of large numbers.

The second MATLAB function `simul1()` implements a simulator that approximates $\mathrm{E}[h(z)]$ using importance sampling. We have

$$\mathrm{E}[h(z)] = \int_0^1 h(z) f(z;a,b) dz = \int_0^1 \frac{h(z) f(z;a,b)}{p(z)} p(z) dz$$

Using Monte Carlo integration we can approximate the integral by

$$x_{Simul2} = 1/n \sum_{i=1}^{n} \frac{h(z) f(z;a,b)}{p(z)} \xrightarrow[n\to\infty]{} \mathrm{E}[h(z)]$$

where $z_1, \ldots, z_n$ are n are now independent random draws from the $p(z)$ rather than from the beta distribution, $f(z;a,b)$; and where $p(z)$ has the same support as the original domain of integration (i.e. $[0,1]$ in this case). In `simul2()` we have set p(z) to be the uniform distribution, which has same support as beta [0,1] and density 1 over the unit interval, ie. $p(z) = 1$.

One advantage of this approach is that we are not required to draw from the distribution of interest; in this case the beta distribution. Instead we should be able to draw from $p(z)$.