

Exercise – Understanding importance sampling using tail probability integration

1. Aim of the exercise

The aim of this exercise is to understand importance sampling as a method for reducing the variance of a Monte Carlo integral estimator. By estimating a tail probability under the standard normal distribution, the exercise demonstrates how different proposal distributions can enhance estimator efficiency. Through a combination of theoretical exposition and practical implementation, it highlights how the choice of proposal density influences bias, variance, and convergence behavior.

2. Theory

The theoretical foundations of Monte Carlo integration and importance sampling are addressed in other exercises; this document focuses on their practical implementation, in particular that of importance sampling.

3. Clear the memory

Remove all existing variables from the workspace to ensure a clean computational environment.

```
13 %% 3. Clear the memory
14
15 % Clear the memory
16 clear;
```

4. Define and plot tail probability region

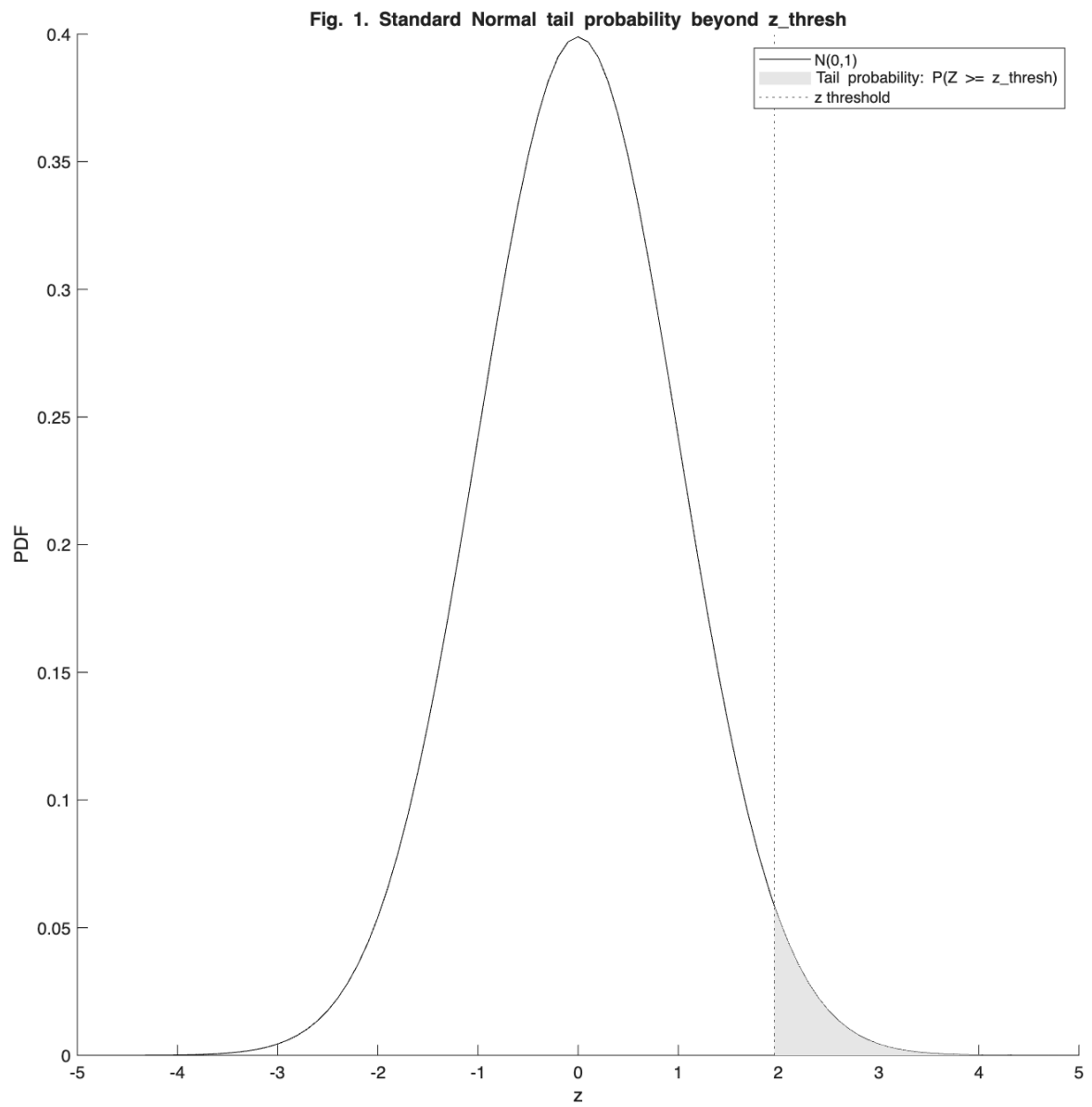
Our aim is to estimate a tail probability using alternative simulation-based estimators. Here we first visualize this tail probability. We begin by constructing a symmetric evaluation grid, ensuring inclusion of the threshold, and evaluate the standard normal PDF over this grid. We then identify the portion of the grid corresponding to $z \geq 1.96$ and highlight this region in the plot. Figure 1 shows the standard normal curve with the tail area shaded, illustrating the probability mass associated with values exceeding the threshold.

```
18 %% 4. Define and plot tail probability region
19
20 % 4.1. Define evaluation grid centered at z_thresh
21 z_base = -5:0.1:5;
22
23 % 4.2. Define threshold for tail probability
24 z_thresh = 1.96;
25
26 % 4.3. Ensure z_thresh is included and grid is sorted
27 z = unique([z_base, z_thresh]);
28
29 % 4.4. Define Standard Normal distribution
30 pd_base_norm = makedist('Normal', 'mu', 0, 'sigma', 1);
```

```

31
32 % 4.5. Evaluate PDF over the grid
33 PDF_target_norm = pdf(pd_base_norm,z);
34
35 % 4.6. Region to highlight where z is at least z_thresh
36 z_tail = z >= z_thresh;
37
38 % 4.7. Plot Standard Normal PDF and highlight tail region
39 figure
40 hold on
41 plot(z,PDF_target_norm, ...
42      'Color',[0.0000 0.0000 0.0000], ...
43      'DisplayName','N(0,1)');
44 fill( ...
45      [z(z_tail),fliplr(z(z_tail))], ...
46      [PDF_target_norm(z_tail),zeros(size(PDF_target_norm(z_tail)))], ...
47      [0.9000 0.9000 0.9000], ...
48      'EdgeColor','none', ...
49      'DisplayName','Tail probability: P(Z >= z\_thresh)');
50 xline(z_thresh,':', ...
51      'DisplayName','z threshold');
52 title('Fig. 1. Standard Normal tail probability beyond z\_thresh');
53 xlabel('z');
54 ylabel('PDF');
55 legend('show');
56 hold off

```



5. Compute true tail probability

Here we compute the area under the tail probability of interest. The variable `int_true_value` stores the exact value of this integral, evaluated using the CDF of the standard normal distribution.

```
58 %% 5. Compute true tail probability
59
60 % 5.1. Define mean of the Normal distribution
61 mu = 0;
62
63 % 5.2. Define standard deviation of the Normal distribution
64 sigma = 1;
65
66 % 5.3. Evaluate tail probability beyond z_thresh as true integral value
67 int_true_value = 1-cdf('Normal',z_thresh,mu,sigma);
```

6. Define number of samples

Specify the number of random samples to be used in simulation-based estimators.

```
69 %% 6. Define number of samples
70
71 % Number of random samples to estimate the integral
72 N_samples = 10000;
```

7. Monte Carlo integration using standard sampling

Let $Z \sim \mathcal{N}(0, 1)$ be a standard normal random variable with PDF:

$$f_Z(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}. \quad (1)$$

To compute the probability that Z exceeds 1.96, we evaluate the tail integral:

$$\begin{aligned} \mathbb{P}(Z \geq 1.96) &= \int_{1.96}^{\infty} f_Z(z) dz \\ &= \int_{1.96}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2} dz \\ &= \int_{-\infty}^{\infty} \mathbb{I}\{z \geq 1.96\} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2} dz \\ &= \mathbb{E}_{f_Z} [\mathbb{I}\{Z \geq 1.96\}] \\ &=: \theta. \end{aligned} \quad (2)$$

The Monte Carlo estimator of this integral is given by:

$$\hat{\theta}_{\text{MC}} = \frac{1}{n} \sum_{i=1}^n \mathbb{I}\{Z_i \geq 1.96\}, \quad (3)$$

where each Z_i is an i.i.d. draw from $\mathcal{N}(0, 1)$. That is, we approximate the tail probability $\mathbb{P}(Z \geq 1.96)$ by averaging indicator values over independent samples from the standard normal distribution.

Since the Z_i are independent, the variance of the estimator is

$$\begin{aligned}\text{Var}(\hat{\theta}_{\text{MC}}) &= \text{Var}\left(\frac{1}{n} \sum_{i=1}^n \mathbb{I}\{Z_i \geq 1.96\}\right) \\ &= \frac{1}{n^2} \sum_{i=1}^n \text{Var}(\mathbb{I}\{Z_i \geq 1.96\}) \\ &= \frac{1}{n^2} \sum_{i=1}^n \mathbb{P}(Z_i \geq 1.96) (1 - \mathbb{P}(Z_i \geq 1.96)) \\ &= \frac{1}{n} \mathbb{P}(Z \geq 1.96) (1 - \mathbb{P}(Z \geq 1.96)).\end{aligned}\tag{4}$$

To see this, note first that the indicator $\mathbb{I}\{Z_i \geq 1.96\}$ is a Bernoulli random variable with success probability $p = \mathbb{P}(Z \geq 1.96)$. For each i , the indicator variable $\mathbb{I}\{Z_i \geq 1.96\}$ equals 1 if the sample Z_i exceeds the threshold, and 0 otherwise. That is, the event $\mathbb{I}\{Z_i \geq 1.96\} = 1$ occurs if and only if $Z_i \geq 1.96$. This equivalence follows directly from the definition of the indicator function, which encodes the event as a binary outcome: 1 when the condition is satisfied, 0 otherwise. As a result, the indicator defines a Bernoulli random variable whose success corresponds to the occurrence of the tail event $Z_i \geq 1.96$. The success probability of this Bernoulli variable is therefore

$$p = \mathbb{P}(\mathbb{I}\{Z_i \geq 1.96\} = 1) = \mathbb{P}(Z_i \geq 1.96),$$

confirming that the probability of the indicator equaling 1 is precisely the probability of the underlying event. For a Bernoulli random variable $X \in \{0, 1\}$ with success probability p , the variance is given by

$$\text{Var}(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2.$$

Since $X^2 = X$ for binary variables, we have $\mathbb{E}[X^2] = \mathbb{E}[X]$. Again for a Bernoulli variable,

$$\mathbb{E}[X] = 0 \cdot (1 - p) + 1 \cdot p = p,$$

so the variance simplifies to

$$\text{Var}(X) = p - p^2 = p(1 - p).$$

Applying this to our indicator clarifies equation (4).

Equation (4) shows that the variance of the Monte Carlo estimator decreases linearly with the sample size n , but remains proportional to the success probability $\mathbb{P}(Z \geq 1.96)$. When $\mathbb{P}(Z \geq 1.96)$ is small, as it is for tail probabilities, the variance $\mathbb{P}(Z \geq 1.96)(1 - \mathbb{P}(Z \geq 1.96))$ is numerically close to $\mathbb{P}(Z \geq 1.96)$, and the estimator becomes noisy unless n is large. Most samples drawn from the standard normal distribution contribute zero, and only a few contribute one, leading to slow convergence.

This variance-driven instability explains why estimating the tail probability $\mathbb{P}(Z \geq 1.96)$ directly via standard Monte Carlo integration can be inefficient. To address this, we apply the technique of importance sampling. By drawing samples from alternative, or proposal, distributions that place more weight in the tail region $Z \geq 1.96$, importance sampling increases

the frequency of nonzero contributions and reduces the estimator's variance. We implement this method and compare its performance across four distinct proposal PDFs.

The code below draws `N_samples` from the standard normal distribution, estimates the tail probability using the MC integral estimator, and computes the mean squared error (MSE) to quantify the estimator's variance.

```

74 %% 7. Monte Carlo integration using standard sampling
75
76 % 7.1. Draw samples from the Standard Normal distribution
77 Z_samples = random('Normal',mu,sigma,[N_samples 1]);
78
79 % 7.2. Estimate tail probability using indicator function
80 int_est_mc = mean(Z_samples >= z_thresh);
81
82 % 7.3. Compute MSE of the Monte Carlo estimator
83 MSE_mc = 1/(N_samples*(N_samples-1)) * ...
84         sum(((Z_samples >= z_thresh)-int_est_mc).^2);

```

8. Importance sampling integration using Normal proposal

The theory of importance sampling requires that p_X matches the product $g(x) \cdot f_X(x)$ as much as possible. Using this condition, we observe that a Normal PDF centered near the threshold 1.96 with small variance concentrates mass in the region of interest. Therefore, we select $\mathcal{N}(\mu, \sigma^2)$ as a proposal PDF.

Let $Y \sim \mathcal{N}(\mu, \sigma^2)$ be a Normal proposal distribution with PDF

$$p_Y(z) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2} \left(\frac{z-\mu}{\sigma}\right)^2\right).$$

Rewriting the integral,

$$\begin{aligned}
\mathbb{P}(Z \geq 1.96) &= \int_{1.96}^{\infty} f_Z(z) dz \\
&= \int_{1.96}^{\infty} \frac{f_Z(z)}{p_Y(z)} \cdot p_Y(z) dz \\
&= \int_{1.96}^{\infty} \frac{\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}}{\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{z-\mu}{\sigma}\right)^2}} \cdot p_Y(z) dz \\
&= \int_{-\infty}^{\infty} \frac{\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}}{\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{z-\mu}{\sigma}\right)^2}} \cdot \mathbb{I}\{z \geq 1.96\} \cdot p_Y(z) dz \\
&= \mathbb{E}_{p_Y} \left[\frac{\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}Z^2}}{\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{Z-\mu}{\sigma}\right)^2}} \cdot \mathbb{I}\{Z \geq 1.96\} \right] \\
&= \mathbb{E}_{p_Y} \left[\frac{f_Z(Z)}{p_Y(Z)} \cdot \mathbb{I}\{Z \geq 1.96\} \right].
\end{aligned} \tag{5}$$

Hence, an importance sampling estimator is

$$\hat{\theta}_{IS}^{\mathcal{N}(\mu, \sigma^2)} = \frac{1}{n} \sum_{i=1}^n \frac{f_Z(Z_i)}{p_Y(Z_i)} \cdot \mathbb{I}\{Z_i \geq 1.96\} \quad (6)$$

where $Z_i \sim \mathcal{N}(\mu, \sigma^2)$.

The code below performs importance sampling using a Normal proposal distribution centered near the tail threshold. It draws samples from the proposal, computes importance weights as the ratio of target to proposal densities, estimates the tail probability, and calculates the MSE of the estimator to assess its variance.

```

86 %% 8. Importance sampling integration using Normal proposal
87
88 % 8.1. Define location parameter of Normal proposal
89 mu = z_thresh;
90
91 % 8.2. Define scale parameter of Normal proposal
92 sigma = 0.5;
93
94 % 8.3. Draw samples from Normal proposal
95 norm_samples = random('Normal', mu, sigma, [N_samples 1]);
96
97 % 8.4. Compute importance weights: target density/proposal density
98 weights_norm = ...
99     normpdf(norm_samples, 0, 1) ./ ...
100     normpdf(norm_samples, mu, sigma);
101
102 % 8.5. Estimate tail probability via importance sampling
103 int_est_is_norm = mean(weights_norm.*(norm_samples >= z_thresh));
104
105 % 8.6. Compute MSE of the importance sampling estimator
106 MSE_norm = var(weights_norm.*(norm_samples >= z_thresh))/N_samples;

```

9. Importance sampling integration using Truncated Normal proposal

The Normal proposal PDF can be improved by truncating the PDF so that its support does not include values lower than 1.96. We consider a truncated Normal distribution $\mathcal{TN}(\mu, \sigma^2, 1.96)$, that is, a Normal distribution truncated below $z = 1.96$ with PDF

$$p_Y(z) = \frac{\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{z-\mu}{\sigma}\right)^2}}{1 - \Phi\left(\frac{1.96-\mu}{\sigma}\right)} \cdot \mathbb{I}\{z \geq 1.96\}, \quad (7)$$

where $\Phi(\cdot)$ denotes the CDF of the standard normal distribution and the denominator is the normalization constant ensuring the PDF integrates to 1 over $[1.96, \infty)$.

Performing the importance sampling technique with proposal PDF p_Y yields:

$$\begin{aligned}
\mathbb{P}(Z \geq 1.96) &= \int_{1.96}^{\infty} f_Z(z) dz \\
&= \int_{1.96}^{\infty} \frac{f_Z(z)}{p_Y(z)} \cdot p_Y(z) dz \\
&= \mathbb{E}_{p_Y} \left[\frac{f_Z(Z)}{p_Y(Z)} \right] \\
&= \mathbb{E}_{p_Y} \left[\frac{\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}Z^2}}{\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{Z-\mu}{\sigma}\right)^2} / (1 - \Phi(\frac{1.96-\mu}{\sigma}))} \right]
\end{aligned} \tag{8}$$

Hence, another importance sampling estimator is

$$\begin{aligned}
\hat{\theta}_{IS}^{\mathcal{TN}(\mu, \sigma^2, 1.96)} &= \frac{1}{n} \sum_{i=1}^n \frac{\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}Z_i^2}}{\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{Z_i-\mu}{\sigma}\right)^2} / (1 - \Phi(\frac{1.96-\mu}{\sigma}))}, \\
Z_i &\sim \mathcal{TN}(\mu, \sigma^2, 1.96)
\end{aligned} \tag{9}$$

where $Z_i \sim \mathcal{TN}(\mu, \sigma^2, 1.96)$.

The code below defines a Normal proposal distribution truncated below the threshold z_{thresh} , draws N samples from it, computes importance weights as the ratio of the target to proposal densities, estimates the tail probability via importance sampling, and calculates the MSE to assess the estimator's variance. MATLAB provides built-in support for truncated normal distributions via the `truncate` function, which eliminates the need for inverse transform sampling, an approach that would otherwise require computing and inverting the CDF of the base normal distribution to generate samples restricted to the truncated domain. By using `truncate`, one can directly define and sample from a truncated distribution, making it straightforward to draw samples from p_Y without manually implementing inversion logic or handling numerical edge cases near the truncation boundary.

```

108 %% 9. Importance sampling integration using Truncated Normal proposal
109
110 % 9.1. Define location parameter of Normal proposal
111 mu = z_thresh;
112
113 % 9.2. Define scale parameter of Normal proposal
114 sigma = 0.5;
115
116 % 9.3. Define base Normal distribution for truncation
117 pd_base_trunc_norm = makedist('Normal','mu',mu,'sigma',sigma);
118
119 % 9.4. Define Trunc. Normal proposal: N(z_thresh,0.5) on [z_thresh,inf)
120 pd_prop_trunc_norm = truncate(pd_base_trunc_norm,z_thresh,inf);
121
122 % 9.5. Draw samples from the Truncated Normal distribution
123 trunc_norm_samples = random(pd_prop_trunc_norm,[N_samples 1]);
124
125 % 9.6. Compute importance weights

```



```

126 weights_trunc_norm = ...
127     normpdf(trunc_norm_samples,0,1) ./ ...
128     (normpdf(trunc_norm_samples,mu,sigma) ./ ...
129     (1 - normcdf(z_thresh,mu,sigma)));
130
131 % 9.7. Estimate tail probability via importance sampling
132 int_est_is_trunc_norm = mean(weights_trunc_norm);
133
134 % 9.8. Compute MSE of the importance sampling estimator
135 MSE_trunc_norm = var(weights_trunc_norm)/N_samples;

```

10. Importance sampling integration using Gamma proposal

Here we try another proposal PDF that matches the product $\mathbb{I}\{z \geq 1.96\} \cdot \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}z^2}$ as much as possible. We consider a Gamma distribution.

Let $Y \sim \text{Gamma}(\alpha, \theta)$ with $\alpha = 5$ and $\theta = 0.5$, with PDF taking the form

$$p_Y(z) = \mathbb{I}\{z > 0\} \cdot \frac{z^{1-1}}{\Gamma(1)2^1}e^{-\frac{z}{2}} = \mathbb{I}\{z > 0\} \cdot \frac{1}{2}e^{-\frac{1}{2}z}.$$

This is somewhat similar to the product we aim to match. Performing the importance sampling technique with this proposal PDF p_Y yields:

$$\begin{aligned}
\mathbb{P}(Z \geq 1.96) &= \int_{1.96}^{\infty} \frac{\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}z^2}}{\frac{1}{2}e^{-\frac{1}{2}z}} \cdot \frac{1}{2}e^{-\frac{1}{2}z} dz \\
&= \int_0^{\infty} \frac{\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}z^2}}{\frac{1}{2}e^{-\frac{1}{2}z}} \cdot \mathbb{I}\{z \geq 1.96\} \cdot \frac{1}{2}e^{-\frac{1}{2}z} dz \\
&= \mathbb{E}_{p_Y} \left[\frac{\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}z^2}}{\frac{1}{2}e^{-\frac{1}{2}z}} \cdot \mathbb{I}\{z \geq 1.96\} \right]
\end{aligned} \tag{10}$$

Hence, another importance sampling estimator is

$$\hat{\theta}_{\text{IS}}^{\text{Gamma}(\alpha, \theta)} = \frac{1}{n} \sum_{i=1}^n \frac{\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}Z_i^2}}{\frac{1}{2}e^{-\frac{1}{2}Z_i}} \cdot \mathbb{I}\{Z_i \geq 1.96\}, \tag{11}$$

where $Z_i \sim \text{Gamma}(1, 2)$.

The code below defines a Gamma proposal distribution, draws N samples, filters those within the integration domain $z \geq z_{\text{thresh}}$, computes importance weights as the ratio of the target to proposal densities, estimates the tail probability via importance sampling, and calculates the MSE to evaluate the estimator's variance.

```

137 %% 10. Importance sampling integration using Gamma proposal
138
139 % 10.1. Define shape parameter of Gamma proposal
140 alpha = 5;
141
142 % 10.2. Define scale parameter of Gamma proposal

```

```

143 theta = 0.5;
144
145 % 10.3. Draw samples from Gamma proposal
146 gamma_samples = random('Gamma',alpha,theta,[N_samples 1]);
147
148 % 10.4. Select samples in the integration domain: z >= z_thresh
149 gamma_selected = gamma_samples(gamma_samples >= z_thresh);
150
151 % 10.5. Compute importance weights
152 weights_gamma = ...
153     normpdf(gamma_selected,0,1)./ ... % Standard Normal
154     gampdf(gamma_selected,alpha,theta); % Gamma
155
156 % 10.6. Estimate tail probability via importance sampling
157 int_est_is_gamma = mean(weights_gamma);
158
159 % 10.7. Compute MSE of the importance sampling estimator
160 MSE_gamma = var(weights_gamma)/length(weights_gamma);

```

11. Compare target and proposal PDFs

We have gathered a Monte Carlo integration estimator and three importance sampling estimators for the integral in equation (2). To understand how each proposal distribution aligns with the target distribution, we produce a visual comparison of their PDFs. Figure 2 shows the standard normal target PDF alongside the proposal PDFs: Normal, Truncated Normal, and Gamma.

All proposals satisfy the basic validity requirements of importance sampling: they assign positive probability wherever the target density is positive (support coverage) and avoid division by zero. Beyond these essentials, important differences remain. The shifted Normal proposal does not meet the condition that the proposal should have heavier tails than the target, so it may perform poorly in extreme regions. The Truncated Normal proposal aligns well with the integrand's restricted support but likewise fails the heavier-tails condition. The Gamma proposal fails the shape-matching condition, since its distributional form does not resemble the Normal tail, and it also lacks heavier tails. These shortcomings can explain the variation in efficiency across proposals.

```

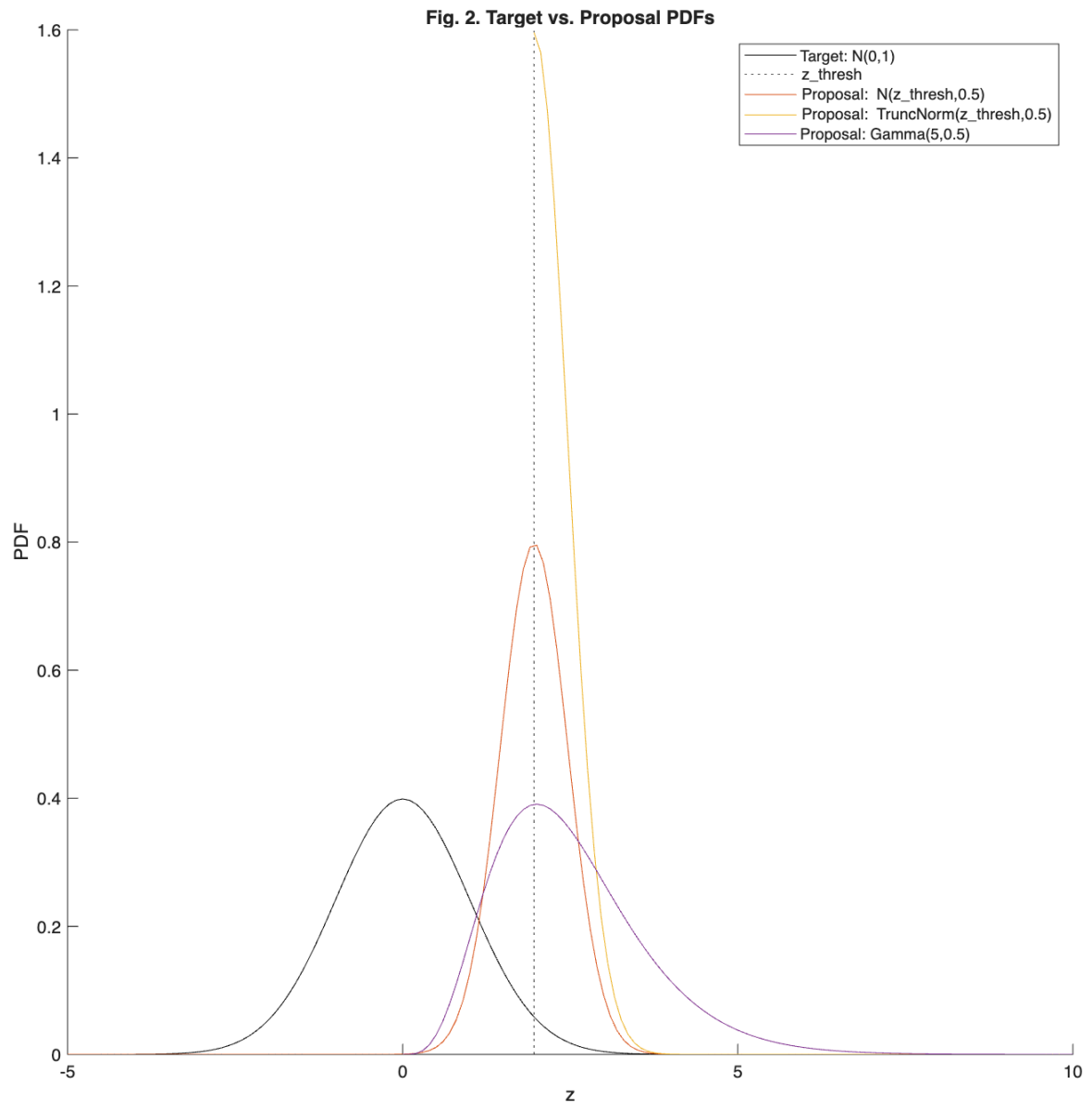
162 %% 11. Compare target and proposal PDFs
163
164 % 11.1. Define full evaluation grid for symmetric distributions
165 x = -5:0.1:10;
166
167 % 11.2. Define positive evaluation grid for Gamma support
168 x_pos = 0:0.1:10;
169
170 % 11.3. Define Truncated evaluation grid for Truncated Normal
171 x_trunc = z_thresh:0.1:10;
172
173 % 11.4. Create Standard Normal distribution object
174 pd_target_norm = makedist('Normal','mu',0,'sigma',1);

```

```

175
176 % 11.5. Evaluate Standard Normal PDF over full grid
177 PDF_target_norm = pdf(pd_target_norm,x);
178
179 % 11.6. Create Normal proposal distribution centered at z_thresh
180 pd_prop_norm = makedist('Normal','mu',z_thresh,'sigma',0.5);
181
182 % 11.7. Evaluate Normal proposal PDF over full grid
183 PDF_prop_norm = pdf(pd_prop_norm,x);
184
185 % 11.8. Create base Normal distribution for truncation
186 pd_base_trunc_norm = makedist('Normal','mu',z_thresh,'sigma',0.5);
187
188 % 11.9. Truncate base Normal distribution to [z_thresh,inf)
189 pd_prop_trunc_norm = truncate(pd_base_trunc_norm,z_thresh,inf);
190
191 % 11.10. Evaluate Truncated Normal PDF over truncated grid
192 PDF_prop_trunc_norm = pdf(pd_prop_trunc_norm,x_trunc);
193
194 % 11.11. Create Gamma proposal distribution
195 pd_prop_gamma = makedist('Gamma',5,0.5);
196
197 % 11.12. Evaluate Gamma proposal PDF over positive grid
198 PDF_prop_gamma = pdf(pd_prop_gamma,x_pos);
199
200 % 11.13. Create comparison plot
201 figure
202 hold on
203 plot(x,PDF_target_norm, ...
204      'Color',[0.0000 0.0000 0.0000], ...
205      'DisplayName','Target: N(0,1)');
206 xline(z_thresh,':', ...
207      'Color',[0.0000 0.0000 0.0000], ...
208      'DisplayName','z\_thresh');
209 plot(x,PDF_prop_norm, ...
210      'Color',[0.8500 0.3250 0.0980], ...
211      'DisplayName','Proposal: N(z\_thresh,0.5)');
212 plot(x_trunc,PDF_prop_trunc_norm, ...
213      'Color',[0.9290 0.6940 0.1250], ...
214      'DisplayName','Proposal: TruncNorm(z\_thresh,0.5)');
215 plot(x_pos,PDF_prop_gamma, ...
216      'Color',[0.4940 0.1840 0.5560], ...
217      'DisplayName','Proposal: Gamma(5,0.5)');
218 title('Fig. 2. Target vs. Proposal PDFs');
219 xlabel('z');
220 ylabel('PDF');
221 legend('show');
222 hold off

```



12. Compare likelihood ratios

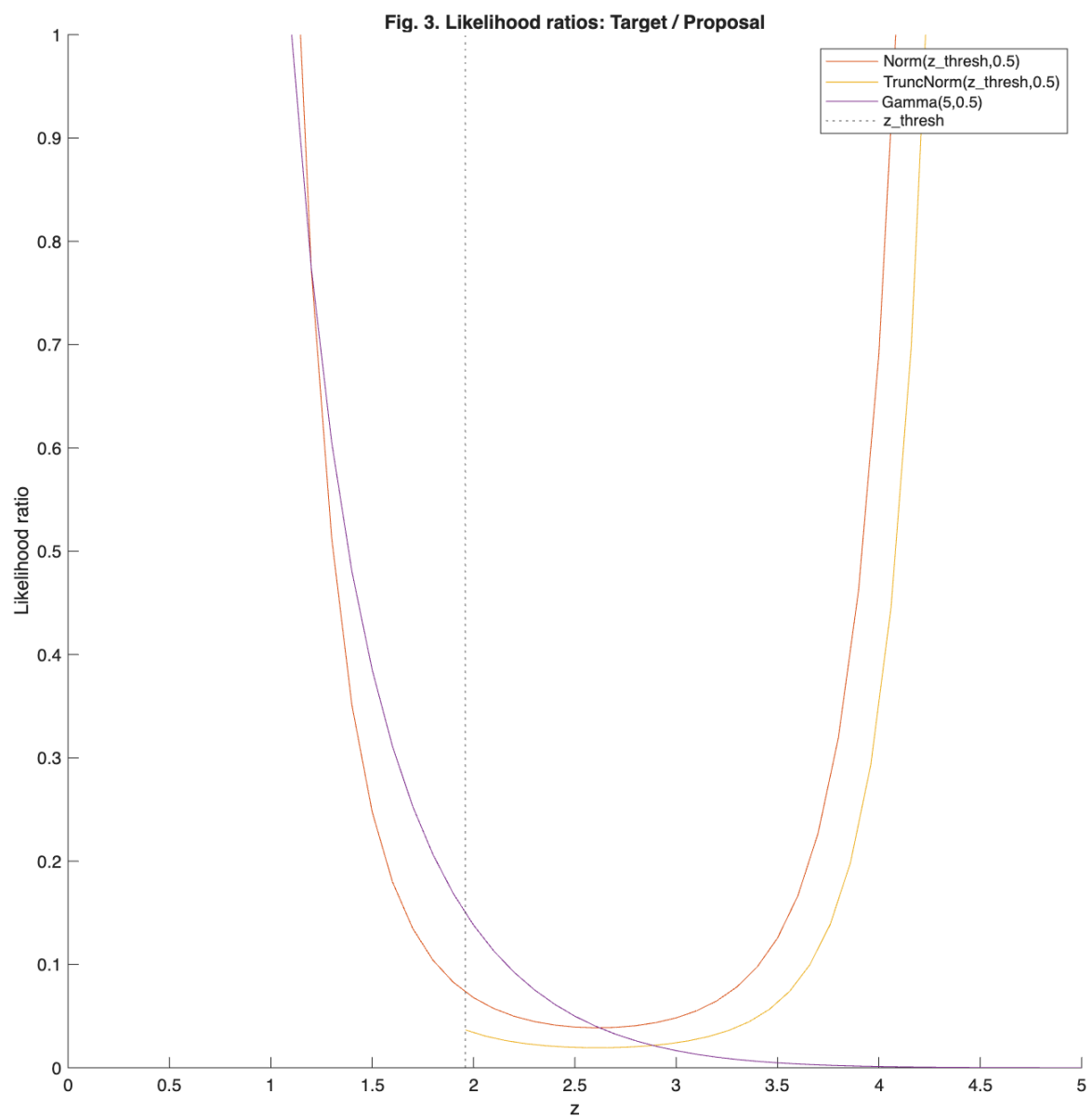
To better understand how different proposal distributions contribute to the importance sampling estimate, we examine the behavior of the likelihood ratio $\frac{p(x)}{q(x)}$ across the domain of integration. Figure 3 plots this ratio for three proposals: a Normal distribution centered at z_{thresh} , a Truncated Normal restricted to $[z_{\text{thresh}}, \infty)$, and a Gamma distribution with positive support. The likelihood ratio quantifies the weight each sample contributes to the estimate. Ideally, this ratio should be stable and bounded to avoid high-variance contributions. The Truncated Normal proposal yields the most favorable profile, smooth, bounded, and concentrated in the region of interest, indicating efficient use of samples. The Normal proposal performs reasonably well but includes regions with larger ratios due to sampling outside the tail. The Gamma proposal exhibits the most erratic behavior, with large likelihood ratios in regions where the target density is small, leading to higher estimator variance. This analysis highlights the importance of proposal-target alignment in reducing variance and improving estimator efficiency.

```
224 %% 12. Compare likelihood ratios
225
226 % 12.1. Define evaluation grid
227 x = -5:0.1:10;
228
229 % 12.2. Define evaluation grid for truncated proposals
230 x_trunc = z_thresh:0.1:10;
231
232 % 12.3. Define evaluation grid for positive-support proposals
233 x_pos = 0:0.1:10;
234
235 % 12.4. Create Standard Normal distribution object
236 pd_target_norm = makedist('Normal','mu',0,'sigma',1);
237
238 % 12.5. Evaluate Standard Normal PDF over full grid
239 PDF_target_norm = pdf(pd_target_norm,x);
240
241 % 12.6. Evaluate Standard Normal PDF over truncated grid
242 PDF_target_trunc_norm = pdf(pd_target_norm,x_trunc);
243
244 % 12.7. Evaluate Standard Normal PDF over positive grid
245 PDF_target_pos_norm = pdf(pd_target_norm,x_pos);
246
247 % 12.8. Create Normal proposal distribution
248 pd_prop_norm = makedist('Normal','mu',z_thresh,'sigma',0.5);
249
250 % 12.9. Evaluate Normal proposal PDF over full grid
251 PDF_prop_norm = pdf(pd_prop_norm,x);
252
253 % 12.10. Create base Normal distribution for truncation
254 pd_base_trunc_norm = makedist('Normal','mu',z_thresh,'sigma',0.5);
255
256 % 12.11. Truncate base Normal distribution to [z_thresh, inf)
257 pd_prop_trunc_norm = truncate(pd_base_trunc_norm,z_thresh,inf);
258
```

```

259 % 12.12. Evaluate Truncated Normal proposal PDF over truncated grid
260 PDF_prop_trunc_norm = pdf(pd_prop_trunc_norm,x_trunc);
261
262 % 12.13. Create Gamma proposal distribution
263 pd_prop_gamma = makedist('Gamma',5,0.5);
264
265 % 12.14. Evaluate Gamma proposal PDF over positive grid
266 PDF_prop_gamma = pdf(pd_prop_gamma,x_pos);
267
268 % 12.15. Create likelihood ratio plot
269 figure
270 hold on
271 plot(x,PDF_target_norm./PDF_prop_norm, ...
272      'Color',[0.8500 0.3250 0.0980], ...
273      'DisplayName','Norm(2,0.5)');
274 plot(x_trunc,PDF_target_trunc_norm./PDF_prop_trunc_norm, ...
275      'Color',[0.9290 0.6940 0.1250], ...
276      'DisplayName','TruncNorm(z\_thresh,0.5)');
277 plot(x_pos,PDF_target_pos_norm./PDF_prop_gamma, ...
278      'Color',[0.4940 0.1840 0.5560], ...
279      'DisplayName','Gamma(5,0.5)');
280 xline(z_thresh,':', ...
281      'Color',[0.0000 0.0000 0.0000], ...
282      'DisplayName','z\_thresh');
283 xlim([0 5]); % xlim([-5 10]) is full domain, including symmetric region
284 ylim([0 1]); % Rescale y-axis to reveal structure
285 title('Fig. 3. Likelihood ratios: Target / Proposal');
286 xlabel('z');
287 ylabel('Likelihood ratio');
288 legend('show');
289 hold off

```



13. Convergence behavior of importance sampling estimators

To evaluate the convergence behavior of different estimators without embedding the entire estimation procedure in a `for` loop, we compare their performance at two representative sample sizes: 1,000 and 100,000. The quantitative results of this exercise naturally depend on the random number generator used to draw samples. While specific estimates and MSE values may vary across runs, the qualitative conclusions, such as the relative performance and convergence behavior of the estimators, should remain largely consistent.

At sample size 1,000, the Normal proposal yields an accurate estimate with a relatively low MSE, outperforming the standard Monte Carlo estimator, which overestimates the tail probability and suffers from the highest MSE. The Truncated Normal performs slightly better in terms of proximity to the true value and achieves a much lower MSE than both Monte Carlo and Normal. The Gamma proposal significantly overestimates and exhibits higher variance than the other estimators. At sample size 100,000, all estimators improve. The Truncated Normal matches the true value exactly and achieves the lowest MSE, followed by the Normal proposal. Both outperform the standard Monte Carlo estimator. The Gamma proposal, although improved in variance, continues to overestimate the tail probability.

This two-point comparison highlights that both Normal and Truncated Normal proposals consistently yield more accurate and lower-variance estimates than standard Monte Carlo sampling, with Truncated Normal showing the closest estimates and strongest performance overall.

Table 1: Comparison of estimators across different sample sizes

Estimator	Sample size			
	1,000		100,000	
	Estimate	MSE	Estimate	MSE
Monte Carlo	0.0290	2.82×10^{-5}	0.0260	2.53×10^{-7}
Normal	0.0256	7.10×10^{-7}	0.0249	8.90×10^{-9}
Truncated Normal	0.0252	4.27×10^{-8}	0.0250	7.34×10^{-10}
Gamma	0.0380	2.65×10^{-6}	0.0389	2.69×10^{-8}

14. Final notes

This file is prepared and copyrighted by Jelmer Wieringa and Tunga Kantarcı. This file and the accompanying MATLAB file are available on GitHub and can be accessed using this [link](#).