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.

```
%% 3. Clear the memory

% Clear the memory
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.

```
%% 4. Define and plot tail probability region

% 4.1. Define evaluation grid centered at z_thresh
z_base = -5:0.1:5;

% 4.2. Define threshold for tail probability
z_thresh = 1.96;

% 4.3. Ensure z_thresh is included and grid is sorted
z = unique([z_base,z_thresh]);

% 4.4. Define Standard Normal distribution
pd_base_norm = makedist('Normal','mu',0,'sigma',1);
```

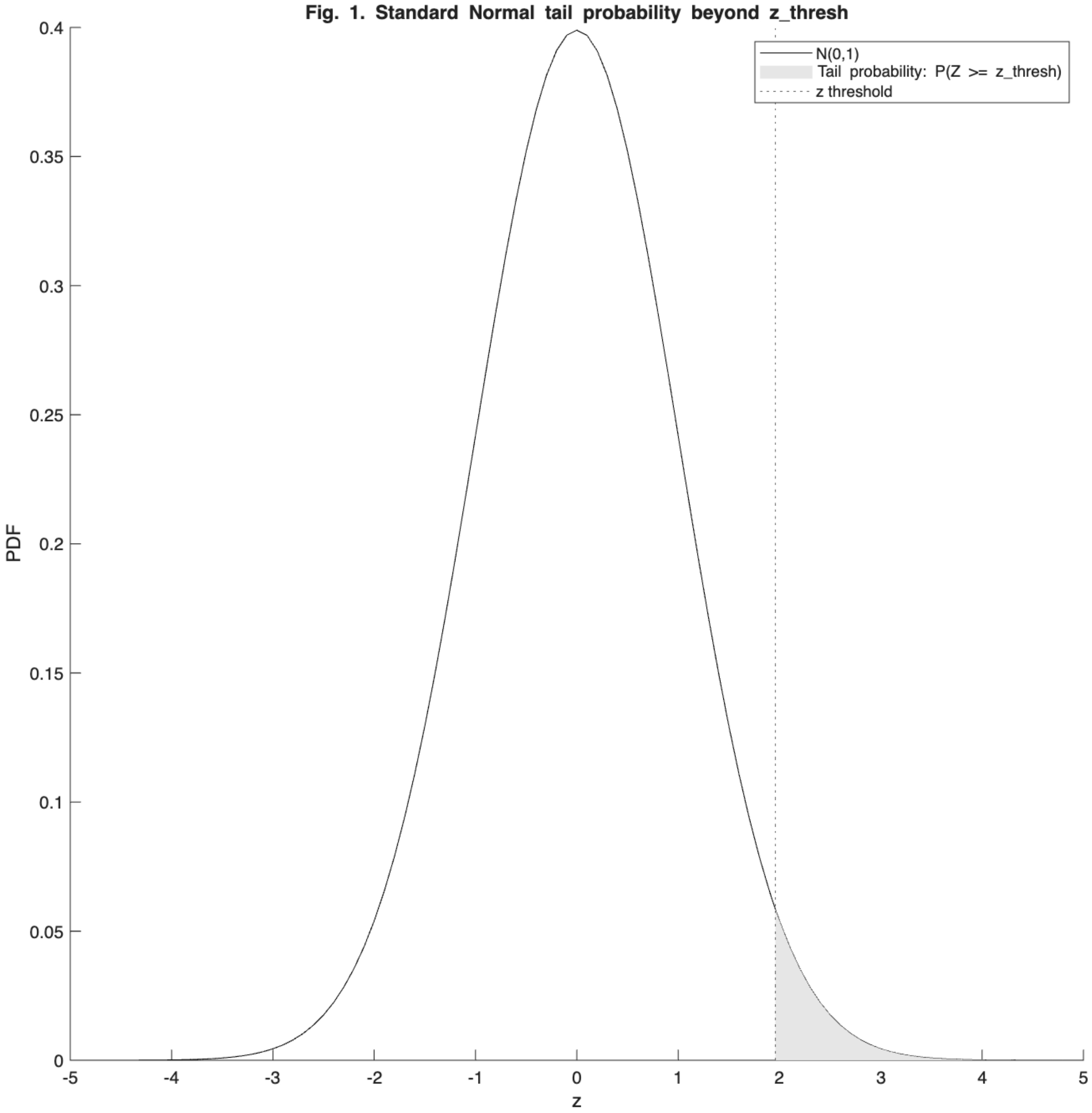
```matlab

% 4.5. Evaluate PDF over the grid
PDF_target_norm = pdf(pd_base_norm,z);

% 4.6. Region to highlight where z is at least z_thresh
z_tail = z >= z_thresh;

% 4.7. Plot Standard Normal PDF and highlight tail region
figure
hold on
plot(z,PDF_target_norm, ...
    'Color',[0.0000 0.0000 0.0000], ...
    'DisplayName','N(0,1)');
fill( ...
    [z(z_tail),fliplr(z(z_tail))], ...
    [PDF_target_norm(z_tail),zeros(size(PDF_target_norm(z_tail)))], ...
    [0.9000 0.9000 0.9000], ...
    'EdgeColor','none', ...
    'DisplayName','Tail probability: P(Z >= z\_thresh)');
xline(z_thresh,':', ...
    'DisplayName','z threshold');
title('Fig. 1. Standard Normal tail probability beyond z\_thresh');
xlabel('z');
ylabel('PDF');
legend('show');
hold off
```

Fig. 1. Standard Normal tail probability beyond z_thresh



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.

```
%% 5. Compute true tail probability

% 5.1. Define mean of the Normal distribution
mu = 0;

% 5.2. Define standard deviation of the Normal distribution
sigma = 1;

% 5.3. Evaluate tail probability beyond z_thresh as true integral value
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.

```
%% 6. Define number of samples

% Number of random samples to estimate the integral
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:

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

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

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

The Monte Carlo estimator of this integral is given by:

$$\hat{\theta}_{\text{MC}} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\left\{Z_i \geq 1.96\right\}, \tag{3}$$

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

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

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

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

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

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

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

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

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.

```
%% 7. Monte Carlo integration using standard sampling

% 7.1. Draw samples from the Standard Normal distribution
Z_samples = random('Normal',mu,sigma,[N_samples 1]);

% 7.2. Estimate tail probability using indicator function
int_est_mc = mean(Z_samples >= z_thresh);

% 7.3. Compute MSE of the Monte Carlo estimator
MSE_mc = 1/(N_samples*(N_samples-1)) * ...
    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

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

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.

```
%% 8. Importance sampling integration using Normal proposal

% 8.1. Define location parameter of Normal proposal
mu = z_thresh;

% 8.2. Define scale parameter of Normal proposal
sigma = 0.5;

% 8.3. Draw samples from Normal proposal
norm_samples = random('Normal',mu,sigma,[N_samples 1]);

% 8.4. Compute importance weights: target density/proposal density
weights_norm = ...
    normpdf(norm_samples,0,1)./ ...
    normpdf(norm_samples,mu,sigma);

% 8.5. Estimate tail probability via importance sampling
int_est_is_norm = mean(weights_norm.*(norm_samples >= z_thresh));

% 8.6. Compute MSE of the importance sampling estimator
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

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

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:

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

Hence, another importance sampling estimator is

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

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.

```
%% 9. Importance sampling integration using Truncated Normal proposal

% 9.1. Define location parameter of Normal proposal
mu = z_thresh;

% 9.2. Define scale parameter of Normal proposal
sigma = 0.5;

% 9.3. Define base Normal distribution for truncation
pd_base_trunc_norm = makedist('Normal','mu',mu,'sigma',sigma);

% 9.4. Define Trunc. Normal proposal: N(z_thresh,0.5) on [z_thresh,inf)
pd_prop_trunc_norm = truncate(pd_base_trunc_norm,z_thresh,inf);

% 9.5. Draw samples from the Truncated Normal distribution
trunc_norm_samples = random(pd_prop_trunc_norm,[N_samples 1]);

% 9.6. Compute importance weights
```

```
weights_trunc_norm = ...
    normpdf(trunc_norm_samples,0,1) ./ ...
    (normpdf(trunc_norm_samples,mu,sigma) ./ ...
    (1 - normcdf(z_thresh,mu,sigma)));

% 9.7. Estimate tail probability via importance sampling
int_est_is_trunc_norm = mean(weights_trunc_norm);

% 9.8. Compute MSE of the importance sampling estimator
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 $\mathbb{I}\left\{z \geq 1.96\right\} \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}$ 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

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

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

$$\begin{aligned} \mathbb{P}(Z \geq 1.96) &= \int_{1.96}^{\infty} \frac{\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}}{\frac{1}{2} e^{-\frac{1}{2}z}} \cdot \frac{1}{2} e^{-\frac{1}{2}z} \, dz \\ &= \int_{0}^{\infty} \frac{\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}}{\frac{1}{2} e^{-\frac{1}{2}z}} \cdot \mathbb{I}\left\{z \geq 1.96\right\} \cdot \frac{1}{2} e^{-\frac{1}{2}z} \, dz \qquad (10) \\ &= \mathbb{E}_{p_Y}\left[\frac{\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}}{\frac{1}{2} e^{-\frac{1}{2}z}} \cdot \mathbb{I}\left\{z \geq 1.96\right\} \right] \end{aligned}$$

Hence, another importance sampling estimator is

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

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.

```
%% 10. Importance sampling integration using Gamma proposal

% 10.1. Define shape parameter of Gamma proposal
alpha = 5;

% 10.2. Define scale parameter of Gamma proposal
```

```
theta = 0.5;

% 10.3. Draw samples from Gamma proposal
gamma_samples = random('Gamma',alpha,theta,[N_samples 1]);

% 10.4. Select samples in the integration domain: z >= z_thresh
gamma_selected = gamma_samples(gamma_samples >= z_thresh);

% 10.5. Compute importance weights
weights_gamma = ...
    normpdf(gamma_selected,0,1)./ ... % Standard Normal
    gampdf(gamma_selected,alpha,theta); % Gamma

% 10.6. Estimate tail probability via importance sampling
int_est_is_gamma = mean(weights_gamma);

% 10.7. Compute MSE of the importance sampling estimator
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.

```
%% 11. Compare target and proposal PDFs

% 11.1. Define full evaluation grid for symmetric distributions
x = -5:0.1:10;

% 11.2. Define positive evaluation grid for Gamma support
x_pos = 0:0.1:10;

% 11.3. Define Truncated evaluation grid for Truncated Normal
x_trunc = z_thresh:0.1:10;

% 11.4. Create Standard Normal distribution object
pd_target_norm = makedist('Normal','mu',0,'sigma',1);
```

```matlab

% 11.5. Evaluate Standard Normal PDF over full grid
PDF_target_norm = pdf(pd_target_norm,x);

% 11.6. Create Normal proposal distribution centered at z_thresh
pd_prop_norm = makedist('Normal','mu',z_thresh,'sigma',0.5);

% 11.7. Evaluate Normal proposal PDF over full grid
PDF_prop_norm = pdf(pd_prop_norm,x);

% 11.8. Create base Normal distribution for truncation
pd_base_trunc_norm = makedist('Normal','mu',z_thresh,'sigma',0.5);

% 11.9. Truncate base Normal distribution to [z_thresh,inf)
pd_prop_trunc_norm = truncate(pd_base_trunc_norm,z_thresh,inf);

% 11.10. Evaluate Truncated Normal PDF over truncated grid
PDF_prop_trunc_norm = pdf(pd_prop_trunc_norm,x_trunc);

% 11.11. Create Gamma proposal distribution
pd_prop_gamma = makedist('Gamma',5,0.5);

% 11.12. Evaluate Gamma proposal PDF over positive grid
PDF_prop_gamma = pdf(pd_prop_gamma,x_pos);

% 11.13. Create comparison plot
figure
hold on
plot(x,PDF_target_norm, ...
    'Color',[0.0000 0.0000 0.0000], ...
    'DisplayName','Target: N(0,1)');
xline(z_thresh,':', ...
    'Color',[0.0000 0.0000 0.0000], ...
    'DisplayName','z\_thresh');
plot(x,PDF_prop_norm, ...
    'Color',[0.8500 0.3250 0.0980], ...
    'DisplayName','Proposal: N(z\_thresh,0.5)');
plot(x_trunc,PDF_prop_trunc_norm, ...
    'Color',[0.9290 0.6940 0.1250], ...
    'DisplayName','Proposal: TruncNorm(z\_thresh,0.5)');
plot(x_pos,PDF_prop_gamma, ...
    'Color',[0.4940 0.1840 0.5560], ...
    'DisplayName','Proposal: Gamma(5,0.5)');
title('Fig. 2. Target vs. Proposal PDFs');
xlabel('z');
ylabel('PDF');
legend('show');
hold off
```

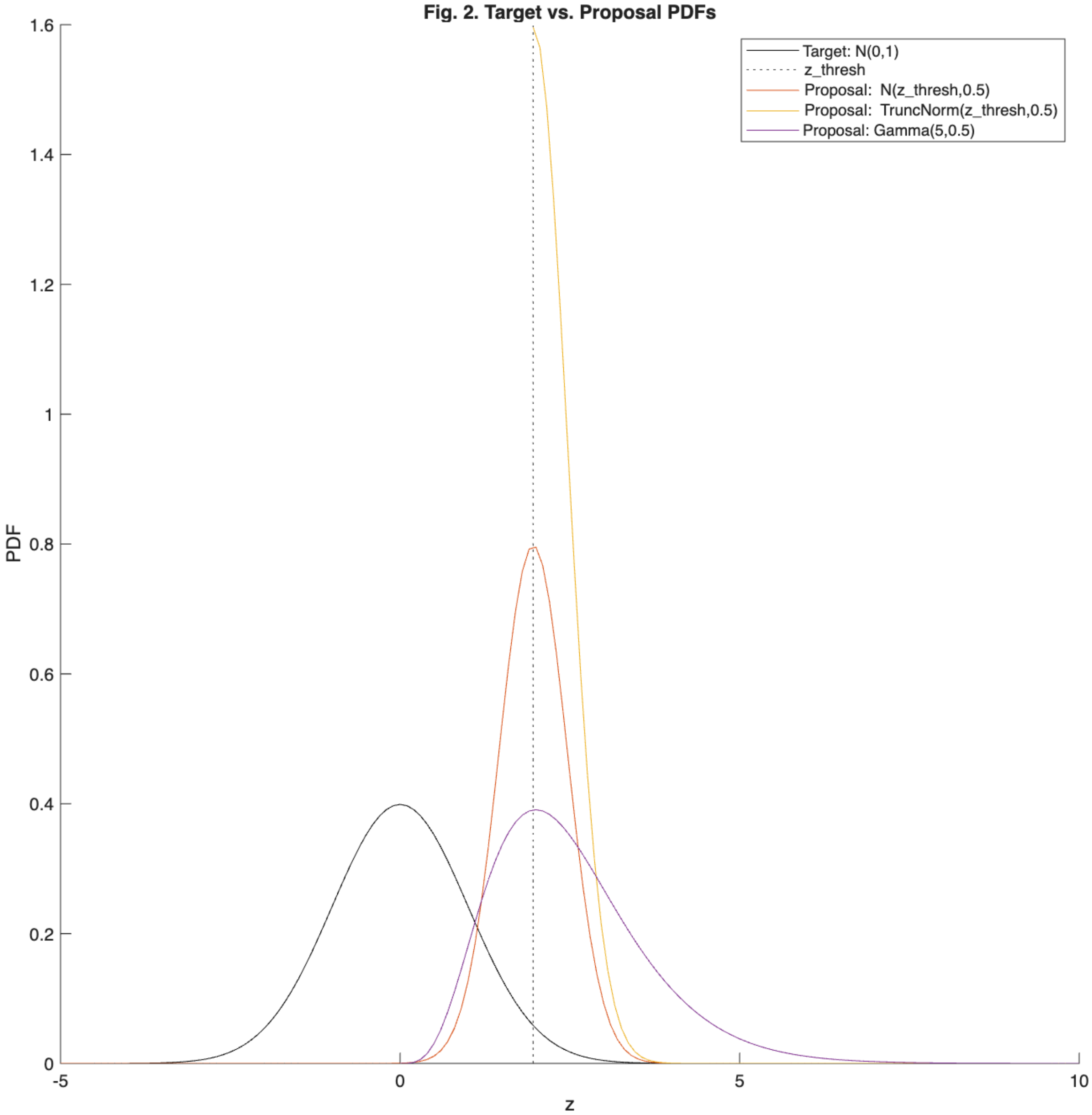


Fig. 2. Target vs. Proposal PDFs

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.

```
%% 12. Compare likelihood ratios

% 12.1. Define evaluation grid
x = -5:0.1:10;

% 12.2. Define evaluation grid for truncated proposals
x_trunc = z_thresh:0.1:10;

% 12.3. Define evaluation grid for positive-support proposals
x_pos = 0:0.1:10;

% 12.4. Create Standard Normal distribution object
pd_target_norm = makedist('Normal','mu',0,'sigma',1);

% 12.5. Evaluate Standard Normal PDF over full grid
PDF_target_norm = pdf(pd_target_norm,x);

% 12.6. Evaluate Standard Normal PDF over truncated grid
PDF_target_trunc_norm = pdf(pd_target_norm,x_trunc);

% 12.7. Evaluate Standard Normal PDF over positive grid
PDF_target_pos_norm = pdf(pd_target_norm,x_pos);

% 12.8. Create Normal proposal distribution
pd_prop_norm = makedist('Normal','mu',z_thresh,'sigma',0.5);

% 12.9. Evaluate Normal proposal PDF over full grid
PDF_prop_norm = pdf(pd_prop_norm,x);

% 12.10. Create base Normal distribution for truncation
pd_base_trunc_norm = makedist('Normal','mu',z_thresh,'sigma',0.5);

% 12.11. Truncate base Normal distribution to [z_thresh, inf)
pd_prop_trunc_norm = truncate(pd_base_trunc_norm,z_thresh,inf);

```

```matlab
% 12.12. Evaluate Truncated Normal proposal PDF over truncated grid
PDF_prop_trunc_norm = pdf(pd_prop_trunc_norm,x_trunc);

% 12.13. Create Gamma proposal distribution
pd_prop_gamma = makedist('Gamma',5,0.5);

% 12.14. Evaluate Gamma proposal PDF over positive grid
PDF_prop_gamma = pdf(pd_prop_gamma,x_pos);

% 12.15. Create likelihood ratio plot
figure
hold on
plot(x,PDF_target_norm./PDF_prop_norm, ...
    'Color',[0.8500 0.3250 0.0980], ...
    'DisplayName','Norm(2,0.5)');
plot(x_trunc,PDF_target_trunc_norm./PDF_prop_trunc_norm, ...
    'Color',[0.9290 0.6940 0.1250], ...
    'DisplayName','TruncNorm(z\_thresh,0.5)');
plot(x_pos,PDF_target_pos_norm./PDF_prop_gamma, ...
    'Color',[0.4940 0.1840 0.5560], ...
    'DisplayName','Gamma(5,0.5)');
xline(z_thresh,':', ...
    'Color',[0.0000 0.0000 0.0000], ...
    'DisplayName','z\_thresh');
xlim([0 5]); % xlim([-5 10]) is full domain, including symmetric region
ylim([0 1]); % Rescale y-axis to reveal structure
title('Fig. 3. Likelihood ratios: Target / Proposal');
xlabel('z');
ylabel('Likelihood ratio');
legend('show');
hold off
```

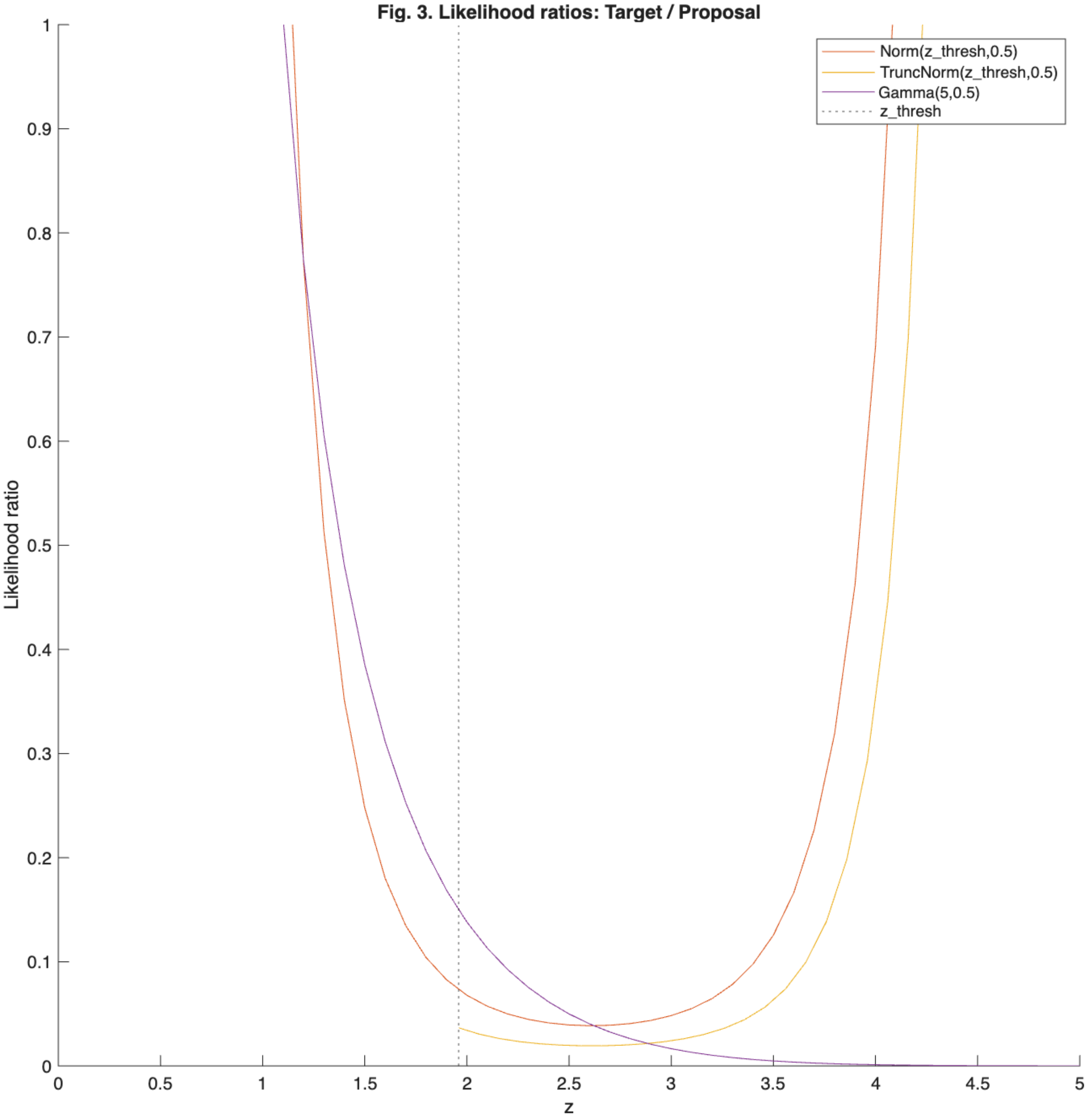


Fig. 3. Likelihood ratios: Target / Proposal

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: N = 1,000 and N = 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 N = 1,000, the Normal proposal yields an accurate estimate (0.0256) with a relatively low MSE (7.10×10^{-7}), outperforming the standard Monte Carlo estimator, which overestimates the tail probability (0.0290) and suffers from the highest MSE (2.82×10^{-5}). The Truncated Normal performs slightly better in terms of proximity to the true value (0.0252) and achieves a much lower MSE (4.27×10^{-8}) than both Monte Carlo and Normal. The Gamma proposal significantly overestimates (0.0380) and exhibits higher variance than the other estimators (2.65×10^{-6}).

At N = 100,000, all estimators improve. The Truncated Normal matches the true value exactly (0.0250) and achieves the lowest MSE (7.34×10^{-10}), followed by the Normal proposal (0.0249, MSE 8.90×10^{-9}). Both outperform the standard Monte Carlo estimator (0.0260, MSE 2.53×10^{-7}). The Gamma proposal, although improved in variance (0.0389, MSE 2.69×10^{-8}), 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.

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.