Assignment 6

1. **Rejection sampling of Gamma.**

   The *gamma* distribution is the PDF

$$f(t) = \begin{cases} \frac{1}{\Gamma}\, t^{p-1} e^{-t} & \text{if } t \geq 0 \\ 0 & \text{if } t < 0 \end{cases} \quad . \tag{1}$$

   It is common to write this as[1] $f(t) \propto t^{p-1}e^{-t}$. The constant of proportionality here is written $\frac{1}{\Gamma}$ (because the usual $\frac{1}{Z}$ is used elsewhere in this exercise). The normalizing constant Γ insures that the integral is equal to 1, which gives

$$\Gamma = \Gamma(p) = \int_0^\infty t^{p-1}e^{-t}\, dt \; .$$

   This is the *Euler gamma function*, which is used in parts of pure math. You can see (using repeated integration by parts) that if p is an integer, then $\Gamma(p) = (p-1)!$, so $\Gamma(1) = 1$, $\Gamma(2) = 1$, $\Gamma(3) = 2$, etc. You may use the Python `scipy` function `scipy.special.gamma` to evaluate $\Gamma(p)$. Rejection sampling from the proposal distribution $g(t) = \lambda e^{-\lambda t}$ leads to

$$Z\frac{1}{\Gamma}t^{p-1}e^{-t} = A(t)\lambda e^{-\lambda t} \; .$$

   Here, Z is the overall acceptance probability and $A(T)$ is the acceptance probability for a proposal[2] $T \sim g(t) = \lambda e^{-\lambda t}$.

   (a) (All the formulas, and $t > 0$ always). First, find a formula for $A(t)$ in terms of t, Z, Γ, and p. Second, find the t_* that maximizes $A(t)$. Third, find the value of Z that makes $A(t_*) = 1$, which is the largest acceptance probability possible for this λ and p. Fourth, find the λ_* as a function of p that maximizes Z (the overall acceptance probability). Fifth, as a reality check, (verify and) explain why $\lambda_* \to 1$ and $Z(\lambda_*) \to 1$ as $p \to 1$. Sixth (extra credit, attempt only when everything else is done and you're still interested) explain why $\lambda_* \to 0$ and $Z \to 0$ as $p \to \infty$. *Remark.* The algorithm is *correct* for any $\lambda \in (0, 1)$, in the sense that the random variables T produced have the target gamma distribution. It is most efficient when $\lambda = \lambda_*$.

[1] The symbol $\propto$ means "is proportional to".

[2] The symbol $\sim$ means "from the probability density". In this case, the PDF of the random variable T is $g(t)$.

(b) Write a Python function (pick another name if you want) `g_samp( lam, p, rng)` that returns one sample of the gamma distribution using rejection. Print an error message if there are more than `max_trials` rejected proposals. Take `max_trials` to be pretty large (maybe a thousand or even ten thousand). Use n (a large number) samples and the histogram method to estimate the PDF they come from. Make a plot with the estimated and target density. Plot the target density (the formula (1)) as a line and the estimated densities at the bin centers as dots. Choose n and Δt (the bin size) so that the estimated and target densities are similar but the dots are not exactly on the curve. The histogram plot should start at $t = 0$ and go out to a $t_{\max}$ where the probability is small. Make a plot with a moderate p and a larger p. Pay attention to the efficiency, which should be low for large p. You will experience inefficiency (small Z) if the code takes a long time to run. (*Extra credit*) Have `g_samp` return a tuple `( T, n_trials)` where `n_trials` is the number of proposals needed to get the accepted sample T. Use this to test whether the actual efficiency is close to Z.

(c) If you plot the gamma density for large p, you will see that it looks approximately Gaussian. It might be more convincing to plot $\log(f(t))$ and see that it is nearly quadratic except in the tails where the density is very small. This suggests using a Gaussian proposal distribution. Explain why this can't work, at least not in this simple form. *Hint.* Compare the Gaussian tails to the exponential or gamma tails.

Histogram hints

- Write a function `hist( X, a, b, n)` that makes n equal sized bins between a and b and returns an integer numpy array containing the bin counts.
- You can find the k with $X_j \in B_k$ by finding the integer part (the *floor* of $(X_j - a)/\Delta x$. Before you do $B[k] = B[k]+1$, test whether k is in the range $0, 1, \cdots, n-1$. If not, then X_j is outside the range of the histogram. The sum of the bin counts is less than n if there are samples X_k outside the histogram range.
- Convert bin counts to density estimates as described in class.
- Debug your histogram function by giving it samples from a known distribution, for example the exponential distribution. That way, if your density estimates for part (b) don't match the target gamma density, you will know the part (b) code is to blame, not the histogram code.

2. **Box/ball rejection is bad.**

A multi-variate random variable $X \in \mathbb{R}^d$ is *uniformly distributed* in a set A if its PDF is

$$f(x) = \begin{cases} \frac{1}{Z} & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

As usual, Z is a normalizing constant chosen so that

$$\int_{\mathbb{R}^d} f(x)\,dx = \frac{1}{Z}\text{vol}(A) = 1\ .$$

In this formula, vol(A) is the volume of A, which a generic word for the size of a set. The "volume" of A is its length if $d = 1$, its area if $d = 2$, its 3D volume if $d = 3$, and so on. The volume is assumed to be finite and not zero. *Warning.* Parts (a), (b), and (c) may seem easy because they are easy. The challenge is to figure out how to say the reasons correctly using the concepts involved.

(a) Suppose U a univariate random variable uniformly distributed in the interval $A = [0, 1]$. Show that $V = 2U - 1$ is uniformly distributed in the interval $A = [-1, 1]$.

(b) Show that if V_1, V_2, $\cdots$, are univariate and uniformly distributed in the interval $[-1, 1]$, then the d component variable $V = (V_1, \cdots, V_d)$ is uniformly distributed in the box[3] Q, which is the set

$$x = (x_1, \cdots, x_d) \in \mathbb{R}^d \ \text{ with } \ -1 \le x_k \le 1 \ \text{ for } k = 1, \cdots, d\ .$$

(c) Suppose that $B \subset A$. Consider the rejection sampler for a uniform random variable in B that proposes X uniform in A and accepts if $X \in B$. Show that this sampler is correct in that the first accepted X is uniformly distributed in B.

(d) The box/ball sampler is a rejection sampler that finds X uniformly distributed in the unit ball by rejection from a sample uniformly distributed in Q. Let $Z(d)$ be the acceptance probability as a function of d. Write a code to implement the box/ball sampler in d dimensions, with d as an argument to the function that does the sampling and returns $\widehat{Z}$, which is the estimate of Z. For example, it could look like

```
def bb_samp(n, d, rng):
    """return an estimate of Z using n samples in d dimensions
       ... more docstring info
    """
       ... the code
    return Z_hat
```

Make a semi-log plot of (the estimated) Z as a function of d for d in a range where the estimates are at least somewhat reliable. What do you conclude about the behavior of Z for moderate to large d and how (speculation) is this related to the curse of dimensionality? You may have to use larger n with larger d to get a reliable $\widehat{Z}$. Repeat the experiment some number of times (maybe 3 or 5 or even 10) and

[3] Boxes may be called Q instead of B so that B can be a ball. "Q" is for "quadrat", which is German for square. If $d \neq 2$, we could talk about a d dimensional "square".

put all the results in the same plot. For this, plot points rather than curves. The result will be several points over each d value, which will give you more idea how accurate the estimates are.