1. **Monte Carlo Methods: Estimating the Volume of the unit n-Sphere**
In class we used a Monte Carlo approach to estimate π and an integral, where both were related to area. Let B_2 be the *unit disk* in the plane ($\mathbb{R}^2$) since it consists of all vectors of length less than or equal to one, i.e., it's the disk centered at the origin and having radius one. We can generalize this to other dimensions.

For a general integer $n \geq 1$, consider vectors of the form $[x_1, \ldots, x_n] \in \mathbb{R}^n$. Define the "box" of side length 2 centered at the origin (denoted by S_n) by:

$$S_n = \{[x_1, \ldots, x_n] : \ -1 \leq x_i \leq 1 \text{ for } i = 1, \ldots, n\}$$

and the *unit ball* in $\mathbb{R}^n$ (denoted by B_n and also known as the unit-n ball) is given by

$$B_n = \{[x_1, \ldots, x_n] : \ x_1^2 + \cdots + x_n^2 \leq 1\}$$

Note that for $n = 1$, this is just the length of the line segment from -1 to 1 in $\mathbb{R}^1$ and for $n = 2$, this is just the unit circle. It's useful to speak of the *volume* of B_n even if in certain cases of n it is thought of area or length.

Suppose we want to estimate the volume of B_n for general n. As in the case of $n = 2$, we can generate a large sample of vectors in $\mathbb{R}^n$ that are uniformly distributed in the box S_n. Recall that "uniformly distributed" just means that the probability that a sample vector lies within a particular subset S of S_n is proportional to the volume of S within S_n, specifically

$$\frac{\text{volume of } S}{\text{volume of } S_n}$$

In particular, the probability that a sample observation lies within B_n is

$$\frac{\text{volume of } B_n}{\text{volume of } S_n}$$

and thus for a large number of samples,

$$\frac{\text{volume of } B_n}{\text{volume of } S_n} \approx \text{fraction of sample in } B_n$$

where the volume of S_n is 2^n. (Recall for $n = 1$, length of line was 2 and for $n = 2$, square inscribing the unit circle was 2^2.) Since 2^n increases exponentially as n grows, this seems to suggest that the volume of the n-ball also increases exponentially with n. But does it really?

   (a) Modify `PiEstimator.m` on canvas in MonteCarlo folder to estimate the volume of B_n for a specified value of n. Using sample sizes of 10^6, verify that it produces the correct results that you expect.

   (b) Add a loop to your code so that it estimates the volume of B_n for $n = 1, 2, \ldots, 20$. Plot the estimates of the value of the volume of the unit n-ball. Make several plots for different sample sizes.

   (c) Look up the analytical formula for the volume and compare your answer for a few different sample sizes.

(d) Convert the code to make use of MATLAB's GPU capabilities and run it on Ace to see what kind of speedups you can get for $n = 10^6$ and $n = 10^7$. What happens if you take $n = 10^8$? Can you think of a way to address the problem?

MATLAB knows to complete a calculation with the GPU when you initialize the array (vector or matrix) as a gpuarray, for example:

```
rand(nObs,nDim,'gpuArray')
```

creates a random vector (on the GPU) from a uniform distribution with nObs rows and nDim columns. You should be able to run interactively (you do not necessarily need a script to submit) but make sure to request GPUs when logging in through sinteractive.

2. **Kalman Filters**
A Kalman filter is often referred to as an optimal estimator (optimal in terms of minimizing the estimated error covariance). It is able to infer parameters even in the presence of noisy data or uncertain measurements. Another feature of the algorithm is that it is recursive, being able to process new measurements as they arrive. The general application is to estimate the state of a process that is governed by a linear and noisy equation. The Kalman filter has been used for applications such as tracking objects, economics, navigation, and computer vision applications. Find an article that uses Kalman Filters for an application area that is of interest to you. Summarize the application area and how the Kalman Filter is used. In particular, describe the model of the state process, model of the measurement process, model of the noise, and testing/refinement of the filter. This should be approximately two paragraphs and you should include a complete citation of the article in the report.

3. **Sampling**
One algorithm to arrive at random points that are not too close (within some tolerance) is the Poisson disc method. One might want to use this sampling algorithm when random points that are evenly distributed in all of the regions is required since this method tightly packs random points without "clumping". Skim the following article and write a paragraph or two that describe/highlight computational complexity, some of the difficulties/considerations to implement a Poisson disc sampling method as proposed in the article, and applications. (Click on link here to get to article): Article