Task 20-01

- Create a Jupyter Lab Notebook called **mc_exp_dist.ipynb** that uses Monte Carlo estimation to calculate the probability an event will occur within one hour of an exponential distribution having a rate parameter of 90 minutes
- Take a minimum of 25,000 random samples and then indicate via different colors which of those random samples are above or below the PDF curve
- On the same graph, use matplotlib to graph the PDF of this distribution
- Calculate and display the percent relative error of your estimate for this probability versus the exact probability
- Upload your solution to the BNL QIS101 SharePoint site

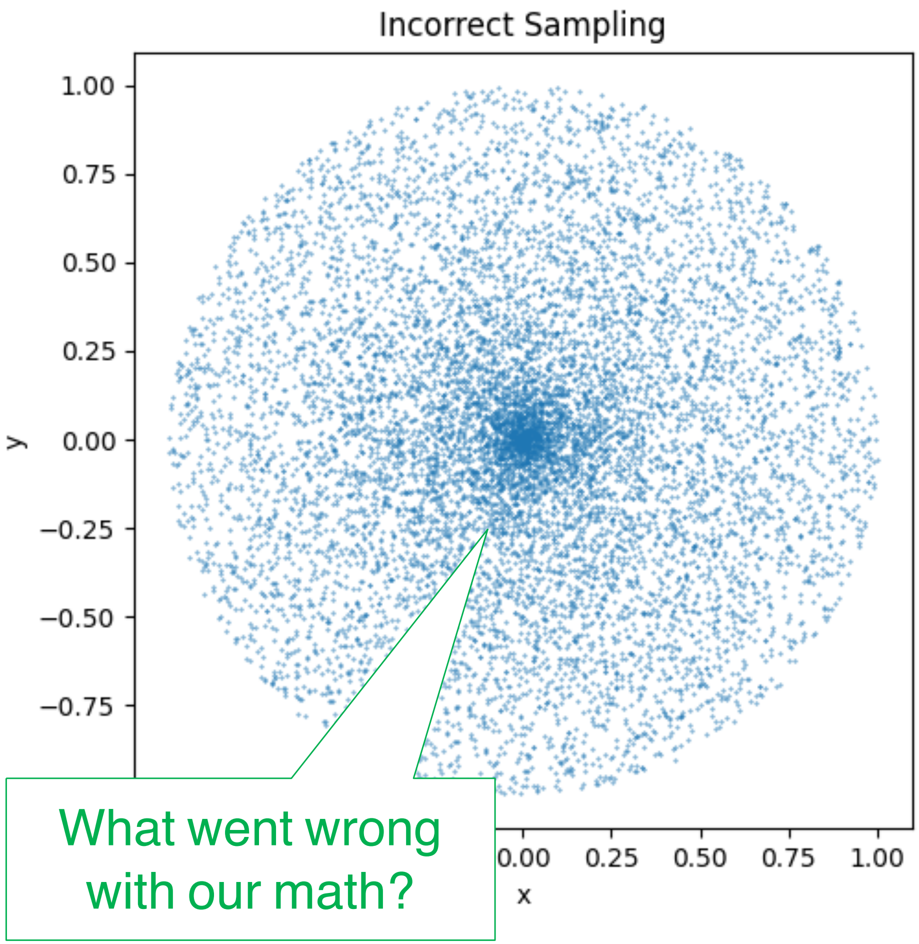
Task 20-02

- Suppose we want to generate random points **known** to be inside a unit circle centered at the origin
- Instead of picking random **Cartesian** coordinates for the sample points, we could use **polar** coordinates
  - We could pick a random **radius** and pick a random **radian angle** and then convert those polar coordinates to Cartesian coordinates
  - Using this approach, we would not have to waste time picking random dots that fall outside of the circle
- We must still ensure the random sample points are distributed uniformly throughout the circle to ensure a **fair** coverage of the entire sample area

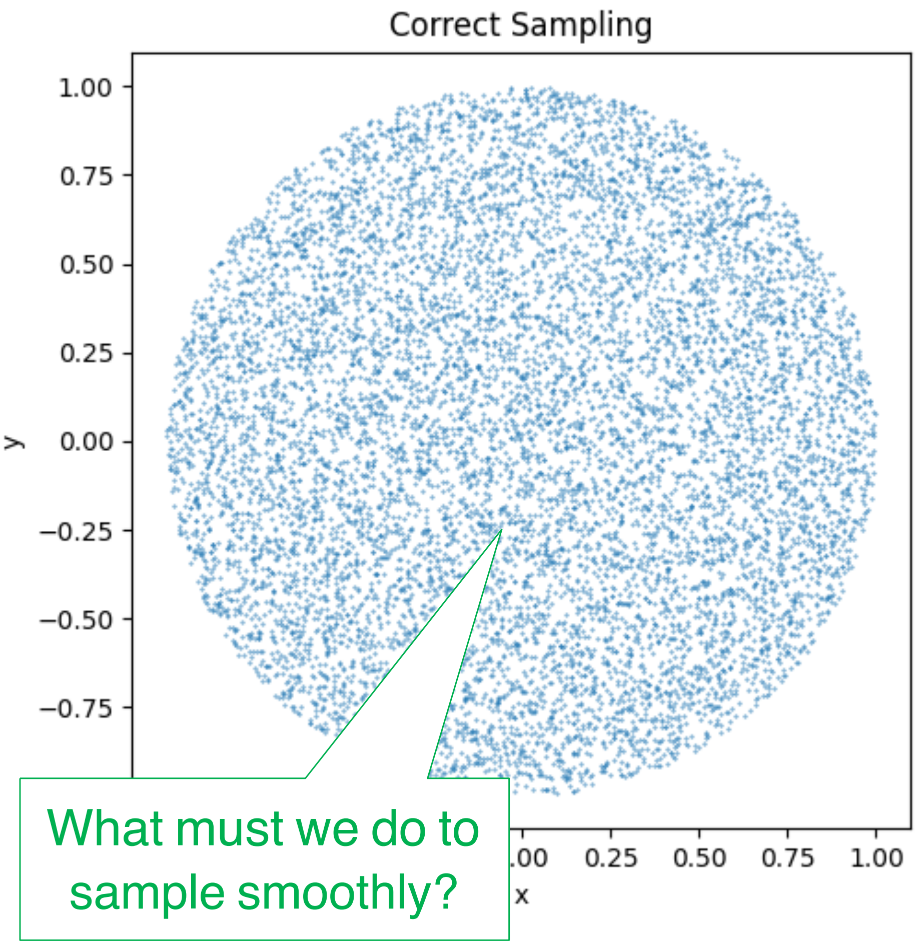
Task 20-02 (Cont.)

- Fix the code in **surface_sampling_circle.ipynb** to correct the problem where the distribution of random points fails to maintain uniform density throughout the inside of the circle
- Upload your solution to the BNL QIS101 SharePoint site

Task 20-02 (Example)



What went wrong with our math?



What must we do to sample smoothly?

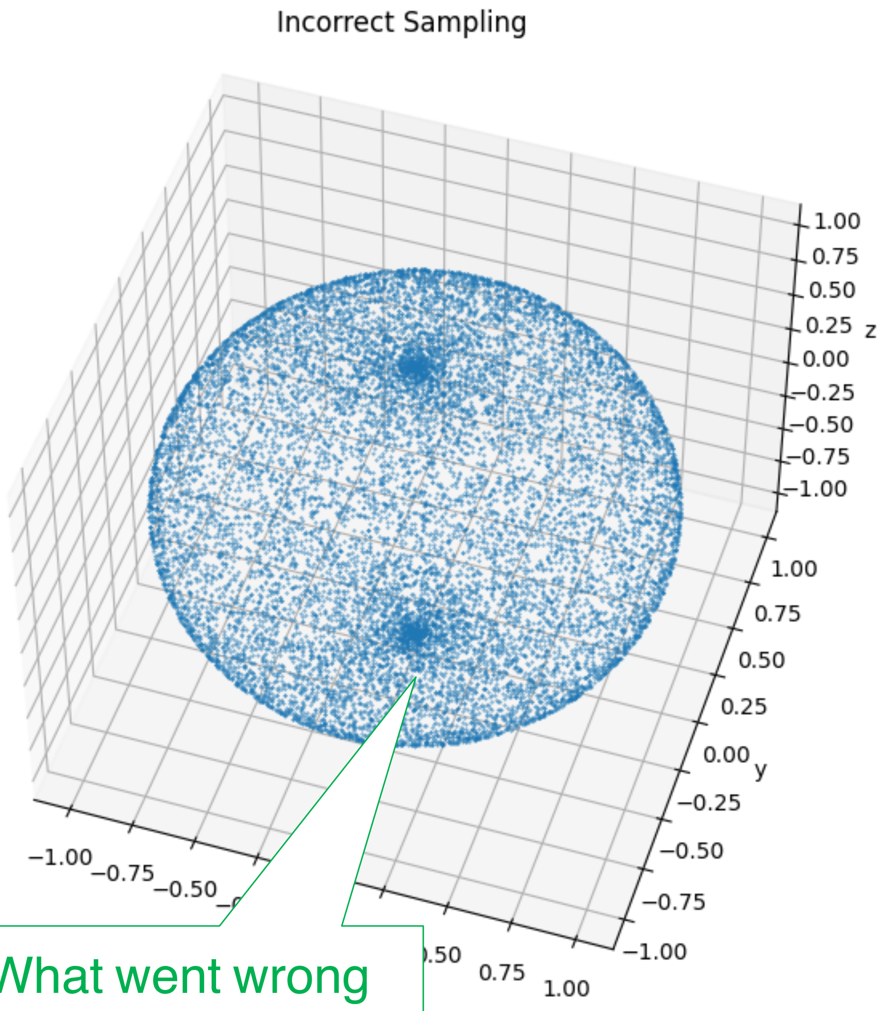
Task 20-03

- We run into a similar issue when trying to use random spherical coordinates to uniformly sample the **surface** of a 3D unit sphere
- The naive approach of picking a random **poloidal** angle and a random **toroidal** angle does not produce a uniform (fair) sampling across the full **face** of the sphere
- Consider what is it about the geometry of the sample space, **as seen from the basis (perspective) of spherical coordinates**, that causes the sampling "density" to deviate from a uniform distribution?

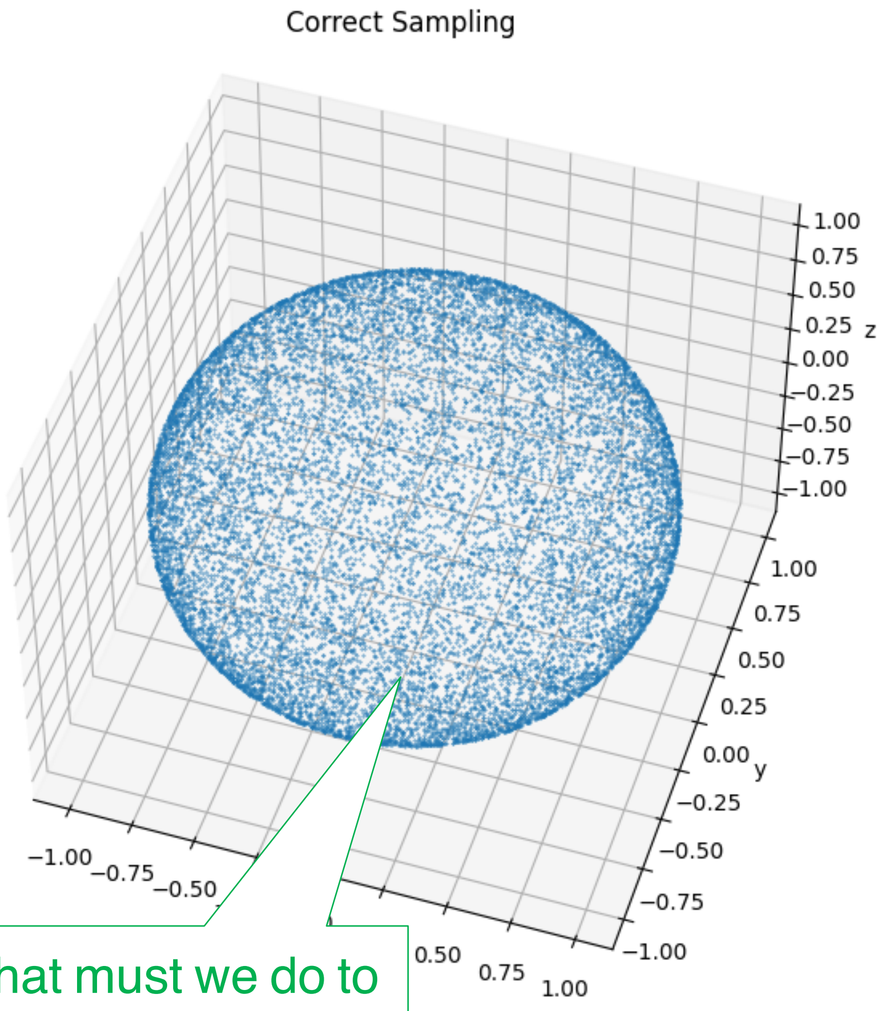
Task 20-03 (Cont.)

- Fix the code in **surface_sampling_sphere.ipynb** to correct the problem where the distribution of random points fails to maintain uniform density across the surface of the sphere
- Upload your solution to the BNL QIS101 SharePoint site

Task 20-03 (Example)



What went wrong with our math?



What must we do to sample smoothly?