Simulation

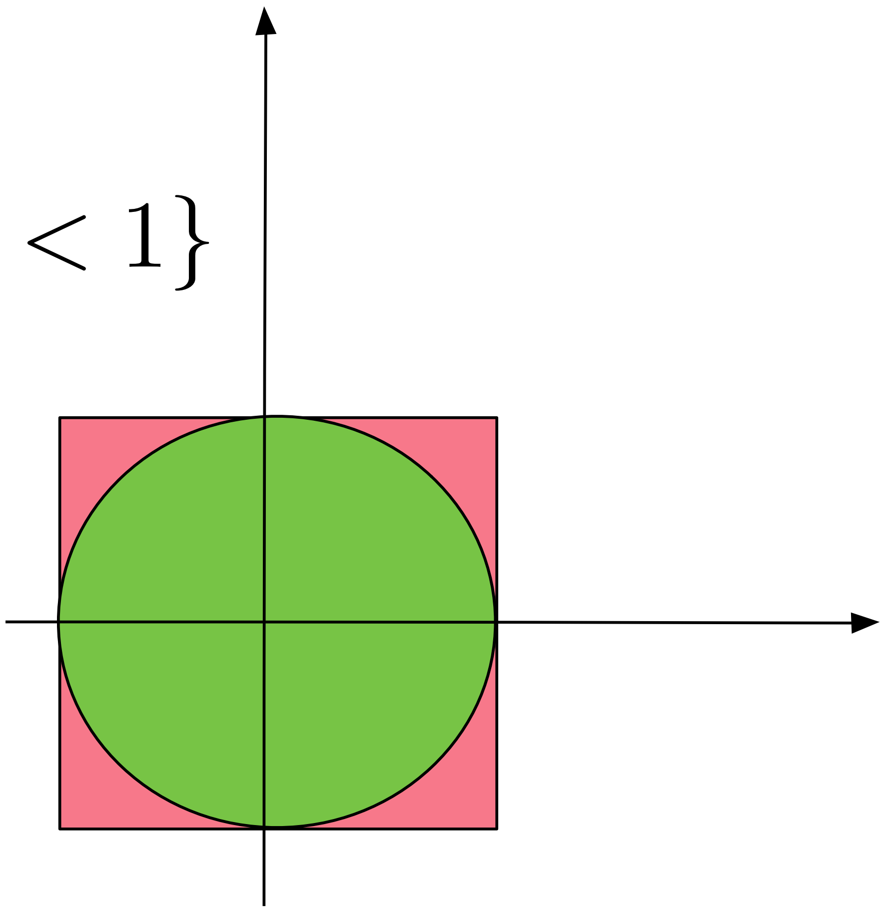
Thomas Schwarz, SJ

A Monte Carlo Method for Area calculation

- Calculate the area of a circle of radius 1
  - Can be done analytically: $$A = r^2 \cdot \pi$$
  - Can be done with Monte Carlo Method
    - Use pseudo-random numbers in order to determine values probabilistically
    - Named after Stanislav Ulam
      - Used for work on the thermo-nuclear device

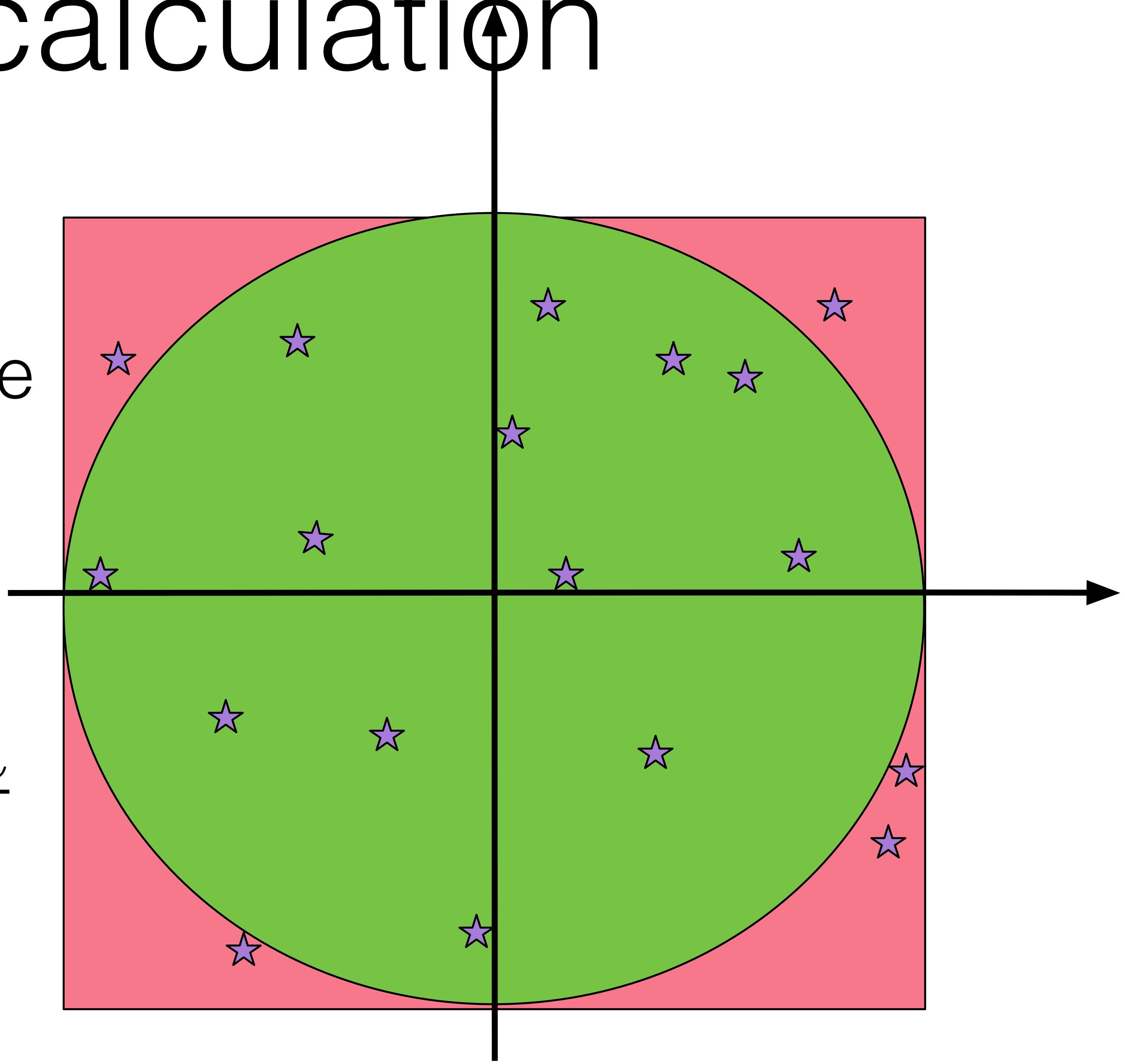
A Monte Carlo Method for Area calculation

- Inscribe Circle with a square
- Circle: $\{(x, y) | x^2 + y^2 < 1\}$
- Square: $\{(x, y) | -1 < x < 1, -1 < y < 1\}$



A Monte Carlo Method for Area calculation

- Method:
  - Choose n random points in the square
  - m points inside circe



$$\frac{\text{Area of Circle}}{\text{Area of Square}} \approx \frac{m}{n}$$

A Monte Carlo Method for Area calculation

- Use random module
  - `random.uniform(-1,1)` generates random number between -1 and 1
  - Generating *20* random numbers:

```
import random

for i in range(20):
    x = random.uniform(-1,1)
    y = random.uniform(-1,1)
    print("({:6.3f},{:6.3f})".format(x,y))
```

A Monte Carlo Method for Area calculation

- We then only count those that are inside the circle

```
import random

def approx(N):
    count = 0
    for i in range(N):
        x = random.uniform(-1,1)
        y = random.uniform(-1,1)
        if x*x+y*y<1:
            count += 1
    return (4*count/N)
```

A Monte Carlo Method for Area Calculations

- Since $\frac{\text{count}}{N} \approx \frac{\text{Area Circle}}{\text{Area Box}}$ and the area of the box is 4
- we return $\frac{4\text{count}}{N}$

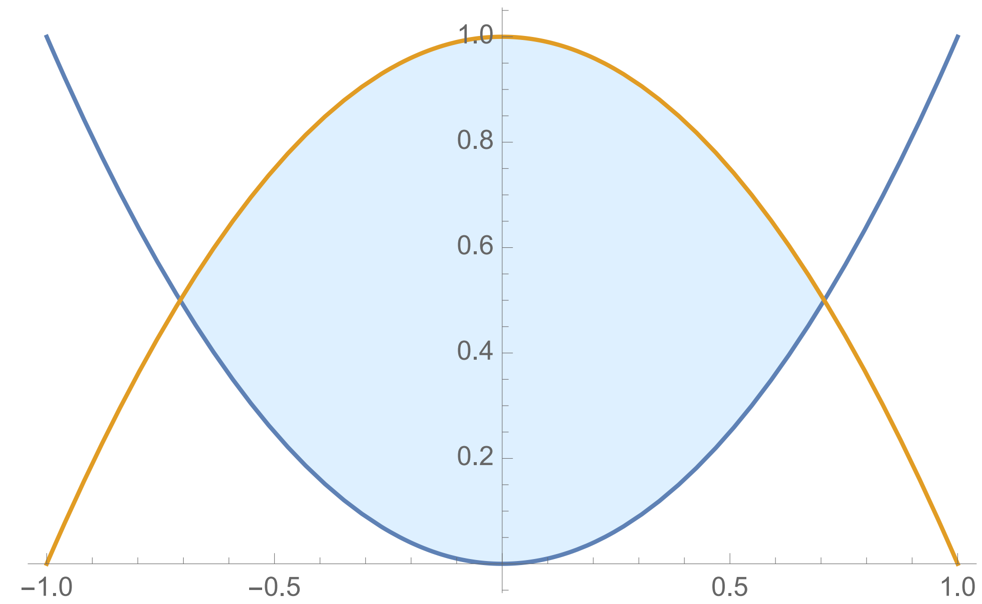
```
import random

def approx(N):
    count = 0
    for i in range(N):
        x = random.uniform(-1,1)
        y = random.uniform(-1,1)
        if x*x+y*y<1:
            count += 1
    return (4*count/N)
```

A Monte Carlo Method for Area calculation

- Need few random point to get a general idea
- Need lots to get any good accuracy
- Method of choice used to determine 6-dimensional integrals for simulation of quantum decay where accuracy is not as important as speed

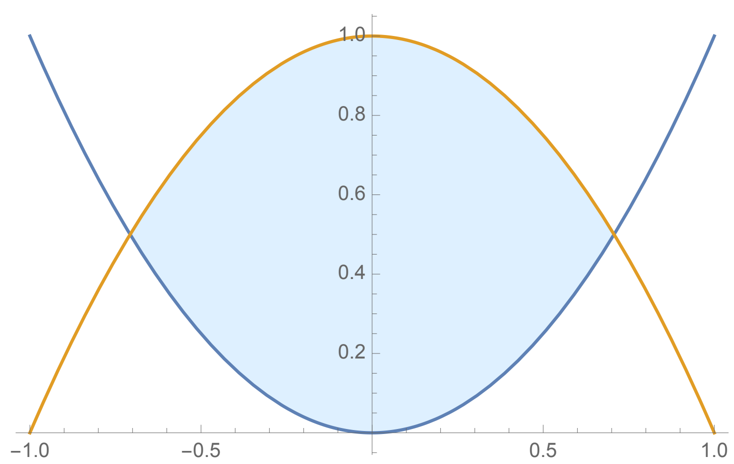
A Monte Carlo Method for Area calculation

- Your task:
  - Determine the area between the curves
$$y = x^2$$
$$y = 1 - x^2$$



  - Hint: We draw points in the rectangle [-1,1] x [0,1]
  - (x,y) lies in the area if
$$x^2 < y < 1 - x^2$$

A Monte Carlo Method for Area calculation



Select random points in the box [-1,1] x [0,1]

Count the number of times that the point falls in the area

Multiply the ratio count / #pts by the area of the box, which is 2

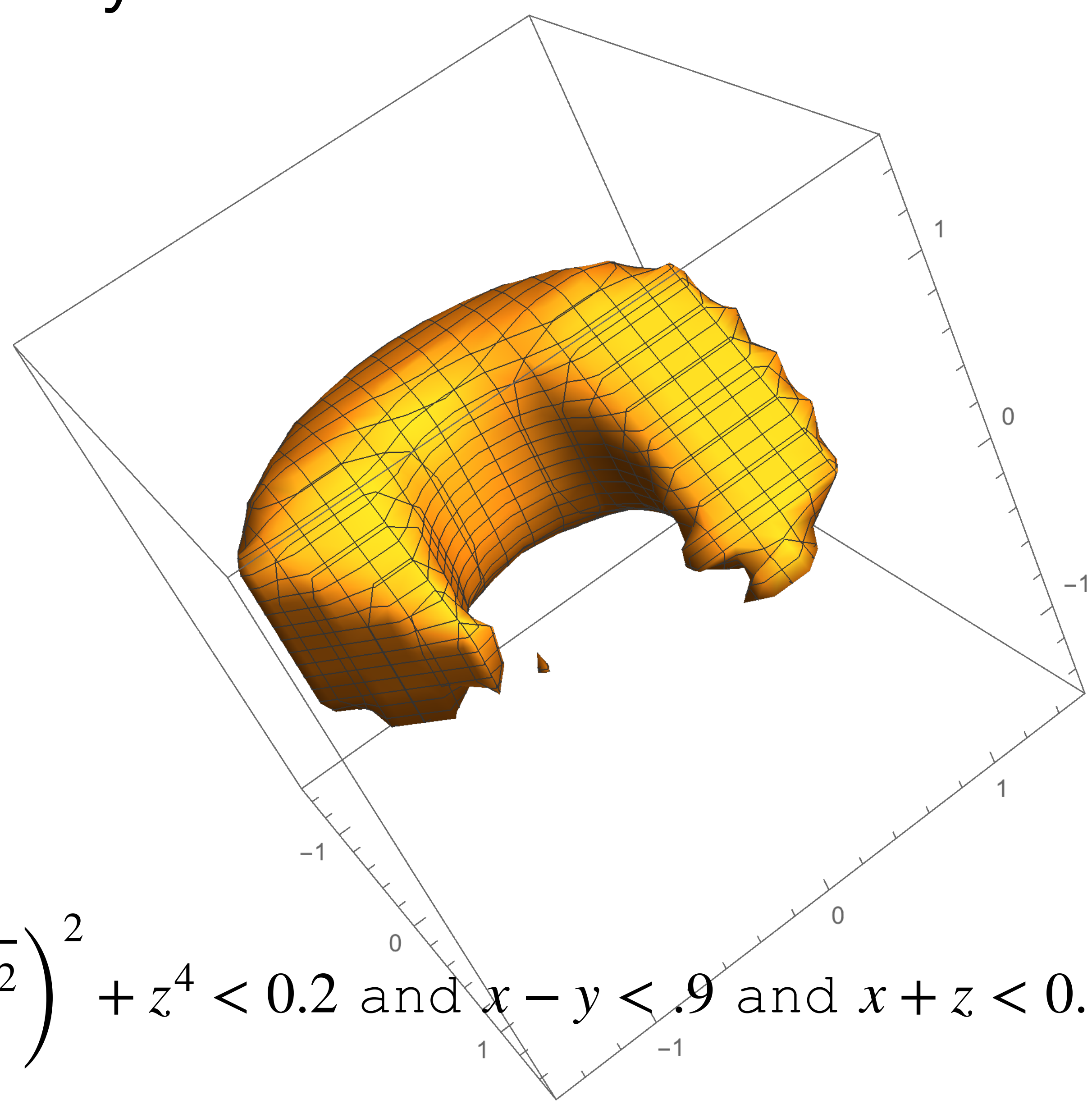
```
import random

N = int(input("Give the number of random points: "))
count = 0
for _ in range(N):
    x = random.uniform(-1,1)
    y = random.uniform(0,1)
    if x*x < y < 1-x*x:
        count += 1
print("The area is approximately", count*2/N)
```

Monte-Carlo Volume Calculation

- Sometimes, Monte-Carlo is the method of choice
  - When there is no need for super-precision
  - When the volume is not easily evaluated using analytic methods.

Volume Calculation

- A partially eaten donut



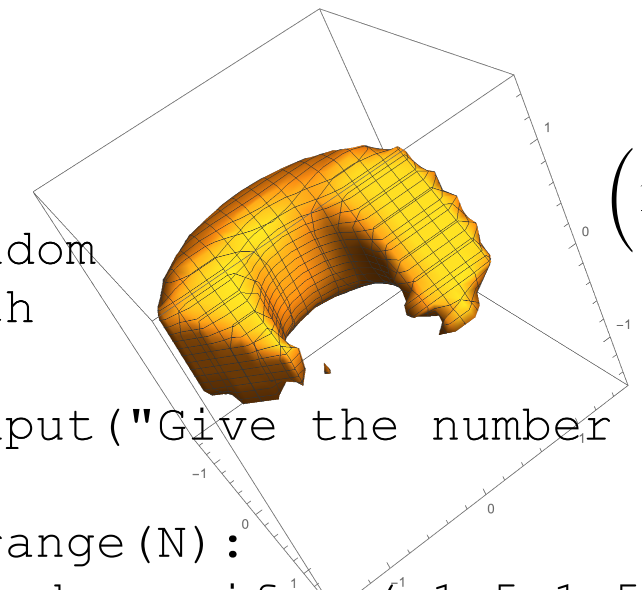
$$\left(1-\sqrt{x^2+y^2}\right)^2+z^4<0.2 \text{ and } x-y<.9 \text{ and } x+z<0.1 \text{ and } x+y<1.8$$

Volume Calculation

- Monte Carlo:
  - Select random points in the box $-1.5<x<1.5$, $-1.5<y<1.5$, $-1.5<z<1.5$.
  - Check whether they are inside the donut
  - Count over total number is approximately area of donut over area of box (which is 9).

Volume Calculation

- A partially eaten donut



$$\left(1-\sqrt{x^2+y^2}\right)^2 + z^4 < 0.2 \text{ and } x - y < .9 \text{ and } x + z < 0.1 \text{ and } x + y < 1.8$$

```
import random
import math

N = int(input("Give the number of random points: "))
count = 0
for _ in range(N):
    x = random.uniform(-1.5,1.5)
    y = random.uniform(-1.5,1.5)
    z = random.uniform(-1.5,1.5)
    if (1-math.sqrt(x**2+y**2))**2+z**4<0.2 and x-y<0.9 and x+z<0.1 and x+y<1.8:
        count += 1
print("The area is approximately", count*9/N)
```

Additional Exercises

- Find the area of

$$\{(x, y) | (x-2)^2 + 3 * (y-1)^2 < 1\}$$

  - Hint: First determine maximum and minimum values for x and y