MATH 3341: Introduction to Scientific Computing Lab

Melissa Butler

University of Wyoming

November 22, 2021

Lab 13: Random Numbers, Histogram & Monte Carlo Integration

Random Numbers and Histogram

`histogram`: Plots a histogram.

- `histogram(x)`: plots a histogram of `x`.
- `histogram(x, m)`: plots a histogram of `x` using `m` bins.

`rand`: Uniformly distributed pseudorandom numbers.

- `rand`: generate a random number that is uniformly distributed on $(0, 1)$.

`rand`: Uniformly distributed pseudorandom numbers.

- `rand`: generate a random number that is uniformly distributed on $(0, 1)$.
- `rand(m, n)` or `rand([m, n])` generates a `m`-by-`n` uniformly distributed random matrix.

`rand`: Uniformly distributed pseudorandom numbers.

- `rand`: generate a random number that is uniformly distributed on $(0, 1)$.
- `rand(m, n)` or `rand([m, n])` generates a `m`-by-`n` uniformly distributed random matrix.
- `rand(n)`: generates an `n`-by-`n` uniform distributied random matrix.

`rand`: Uniformly distributed pseudorandom numbers.

- `rand`: generate a random number that is uniformly distributed on $(0, 1)$.
- `rand(m, n)` or `rand([m, n])` generates a `m`-by-`n` uniformly distributed random matrix.
- `rand(n)`: generates an `n`-by-`n` uniform distributied random matrix.
- `rand(size(A))` generates a uniformly distributed random matrix of the same size as `A`.

rand: Uniformly distributed pseudorandom numbers.

- `rand`: generate a random number that is uniformly distributed on $(0, 1)$.
- `rand(m, n)` or `rand([m, n])` generates a `m`-by-`n` uniformly distributed random matrix.
- `rand(n)`: generates an `n`-by-`n` uniform distributied random matrix.
- `rand(size(A))` generates a uniformly distributed random matrix of the same size as `A`.
- Example:

```
left = -2;
right = 2;
% Uniformly distributed on [left, right]
numbers = rand(10, 1) * (right - left) + left;
histogram(numbers);
```

`randn`: Normally distributed pseudorandom numbers.

- `randn`: generate a randnom number that is normally distributed with mean 0 and standard deviation 1.

`randn`: Normally distributed pseudorandom numbers.

- `randn`: generate a randnom number that is normally distributed with mean 0 and standard deviation 1.
- `randn(m, n)` or `randn([m, n])` generates a `m`-by-`n` normally distributed randnom matrix.

`randn`: Normally distributed pseudorandom numbers.

- `randn`: generate a randnom number that is normally distributed with mean 0 and standard deviation 1.
- `randn(m, n)` or `randn([m, n])` generates a `m`-by-`n` normally distributed randnom matrix.
- `randn(n)`: generates an `n`-by-`n` uniform distributied randnom matrix.

`randn`: Normally distributed pseudorandom numbers.

- `randn`: generate a randnom number that is normally distributed with mean 0 and standard deviation 1.
- `randn(m, n)` or `randn([m, n])` generates a `m`-by-`n` normally distributed randnom matrix.
- `randn(n)`: generates an `n`-by-`n` uniform distributied randnom matrix.
- `randn(size(A))` generates a normally distributed randnom matrix of the same size as `A`.

randn: Normally distributed pseudorandom numbers.

- `randn`: generate a randnom number that is normally distributed with mean 0 and standard deviation 1.
- `randn(m, n)` or `randn([m, n])` generates a `m`-by-`n` normally distributed randnom matrix.
- `randn(n)`: generates an `n`-by-`n` uniform distributied randnom matrix.
- `randn(size(A))` generates a normally distributed randnom matrix of the same size as `A`.
- Example:

```
mu = -2;      % mean
sigma = 2;   % standard devidation
% Normally distributed with mean -2 and standard deviat
numbers = randn(10, 1) * sigma  + mu;
histogram(numbers);
```

Monte Carlo Integration

1-D Monte Carlo Integration

$$\begin{aligned}
\int_a^b f(x)\,dx &= \int_a^b \frac{f(x)}{p(x)} p(x)\,dx \\
&= E[f(X)/p(X)] \\
&\approx \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)} \\
&= \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{1/(b-a)} \\
&= \frac{b-a}{N}\sum_{i=1}^{N} f(x_i),
\end{aligned}$$

where $x_1, x_2, \ldots, x_N$ are uniformly distributed on $[a, b]$, hence $p(x_i) = \frac{1}{b-a}, i = 1, 2, \ldots, N$.

2-D Monte Carlo Integration

$$
\begin{aligned}
\int_a^b \int_c^d f(x,y)\,dy\,dx &= \int_a^b \int_c^d \frac{f(x,y)}{p(x,y)} p(x,y)\,dy\,dx \\
&= \int_a^b \int_c^d \frac{f(x,y)}{p_X(x)p_Y(y)} p_X(x)p_Y(y)\,dy\,dx \\
&= \int_a^b \frac{1}{p_X(x)} \left[\int_c^d \frac{f(x,y)}{p_Y(y)} p_Y(y)\,dy \right] p_X(x)\,dx \\
&= E\left[\frac{f(X,Y)}{p_Y(Y)p_X(X)} \right] \\
&\approx \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i,y_i)}{p_X(x_i)p_Y(y_i)}.
\end{aligned}
$$

2-D Monte Carlo Integration

$$\begin{aligned}\int_a^b \int_c^d f(x,y)\,dy\,dx &\approx \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i,y_i)}{p_X(x_i)p_Y(y_i)} \\ &= \frac{1}{N}\sum_{i=1}^{N} \frac{1}{[1/(b-a)][1/(d-c)]} f(x_i,y_i) \\ &= \frac{(b-a)(d-c)}{N}\sum_{i=1}^{N} f(x_i,y_i),\end{aligned}$$

where X and Y are independent and identically uniformly distributed, hence $p(x,y) = p_X(x)p_Y(y)$, and $p_X(x) = \frac{1}{b-a}$, $p_Y(y) = \frac{1}{d-c}$.

2-D Monte Carlo Integration Example

$$\int_{D: x^2+y^2 \leq 1} f(x,y)\, dA = \int_{-1}^{1} \int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} f(x,y)\, dy\, dx$$

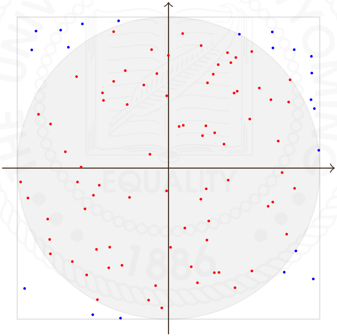

Figure 1:When $N = 100$.

2-D Monte Carlo Integration Example

$$\int_{D:x^2+y^2\le 1} f(x,y)\,dA = \int_{-1}^{1}\int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} f(x,y)\,dy\,dx$$

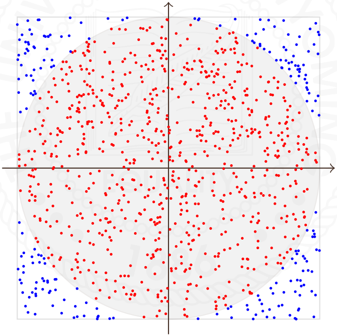

Figure 2:When $N = 1000$.

2-D Monte Carlo Integration Example

$$\int_{D: x^2+y^2 \leq 1} f(x,y)\, dA = \int_{-1}^{1} \int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} f(x,y)\, dy\, dx$$

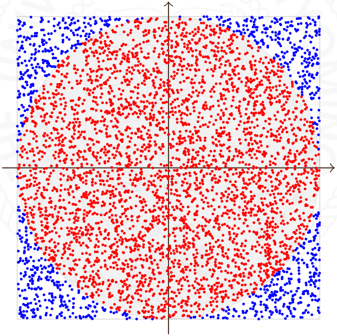

Figure 3:When $N = 5000$.

2-D Monte Carlo Integration Example

$$\int_{D:x^2+y^2\leq 1} f(x,y)\,dA = \int_{-1}^{1}\int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} f(x,y)\,dy\,dx$$

1. Since $-1 \leq x \leq 1$ and $-1 \leq -\sqrt{1-x^2} \leq y \leq \sqrt{1-x^2} \leq 1$, the bounding box is $[-1,1]\times[-1,1]$ on which we can generate the random points $(x_i, y_i), i = 1, 2, \ldots, N$.

2-D Monte Carlo Integration Example

$$\int_{D:x^2+y^2\leq 1} f(x,y)\,dA = \int_{-1}^{1}\int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} f(x,y)\,dy\,dx$$

1. Since $-1 \leq x \leq 1$ and $-1 \leq -\sqrt{1-x^2} \leq y \leq \sqrt{1-x^2} \leq 1$, the bounding box is $[-1,1]\times[-1,1]$ on which we can generate the random points $(x_i, y_i), i = 1, 2, \ldots, N$.
2. We are integrating over the disk, so we need to discard the points outside the disk (blue points):

$$g(x_i,y_i) = \begin{cases} f(x_i,y_i) & \text{if } (x_i,y_i) \text{ is inside the disk,} \\ 0 & \text{otherwise.} \end{cases}$$

2-D Monte Carlo Integration Example

$$\int_{D:x^2+y^2\le 1} f(x,y)\,dA = \int_{-1}^{1}\int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} f(x,y)\,dy\,dx$$

1. Since $-1 \le x \le 1$ and $-1 \le -\sqrt{1-x^2} \le y \le \sqrt{1-x^2} \le 1$, the bounding box is $[-1,1]\times[-1,1]$ on which we can generate the random points $(x_i,y_i), i=1,2,\dots,N$.
2. We are integrating over the disk, so we need to discard the points outside the disk (blue points):
$$g(x_i,y_i) = \begin{cases} f(x_i,y_i) & \text{if } (x_i,y_i) \text{ is inside the disk,} \\ 0 & \text{otherwise.} \end{cases}$$
3. Therefore, we can obtain
$$\int_{-1}^{1}\int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} f(x,y)\,dy\,dx \approx \frac{[1-(-1)]\cdot[1-(-1)]}{N}\sum_{i=1}^{N} g(x_i,y_i).$$

2-D Monte Carlo Integration Example

How do we check whether (x_i, y_i) is inside the disk?

$$-\sqrt{1-x_i^2} \leq y_i \leq \sqrt{1-x_i^2} \implies y_i^2 \leq 1 - x_i^2 \implies x_i^2 + y_i^2 \leq 1.$$

In MATLAB: we can define $g(x, y)$ as follows:

```
g = @(x,y) f(x,y).*(-sqrt(1-x.^2)<=y & y<=sqrt(1-x.^2));
```

or

```
g = @(x,y) f(x,y) .* (x.^2 + y.^2 <= 1);
```