ECSE 446/546

Image Synthesis

Monte Carlo Integration I

image credit: feelgrafix

Prof. Derek Nowrouzezahrai

derek@cim.mcgill.ca

(slides in part by W. Jarosz)

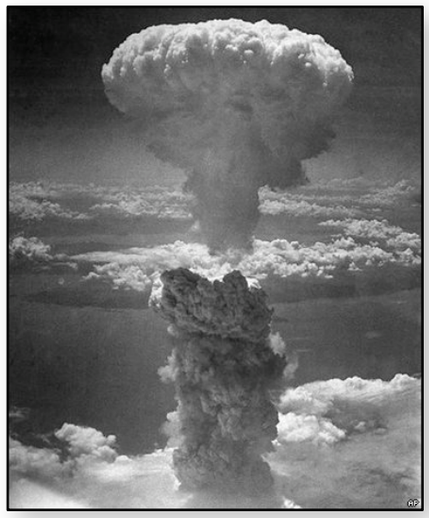
McGill

Today

- Probability review
- Monte Carlo integration
- Random sampling

Monte Carlo History

- Use random numbers to solve numerical problems
- Early use during development of atomic bomb
- Von Neumann, Ulam, Metropolis
- Named after the casino in Monte Carlo




Monte Carlo Methods

- Pros
  - Flexible
  - Easy to implement
  - Easily handles complex integrands
  - Efficient for high dimensional integrands
- Cons
  - Variance (noise)
  - Slow convergence* $O(1/\sqrt{N})$

Random Variables

- Random variable X
- Cumulative distribution function (CDF)

$$P(x) = \text{Prob}\{X \leq x\} \quad P(x) \in [0, 1]$$

- Probability density function (PDF)

$$p(x) = \frac{dP(x)}{dx} \qquad p(x) \geq 0 \qquad \int p(x) = 1$$

- Therefore

$$\text{Prob}\{a \leq X \leq b\} = \int_a^b p(x)\, dx = P(b) - P(a)$$

Random Variables

- Uniform random variables:

$$p(x) = \frac{dP}{dx}(x) = \text{const}$$

- Canonical uniform random variable ξ

$$p(x) = \begin{cases} 1 & x \in [0,1], \\ 0 & \text{otherwise.} \end{cases}$$

Expected Value & Variance

- Random variable $Y = f(X)$
- Expected Value: $E[Y] = \int_D f(x)p(x)\,dx$
- Variance: $V[Y] = E[(Y - E[Y])^2]$
- Properties:

$$E[aY] = aE[Y]$$
$$E[Y_1 + Y_2] = E[Y_1] + E[Y_2]$$
$$V[aY] = a^2V[Y]$$

- Therefore:

$$V[Y] = E[Y^2] - E[Y]^2$$

Expected Value

- Expected value:

$$E[Y] = \int_D f(x)p(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

where the x_i are distributed according to $p(x_i)$

- Strong law of large numbers:

$$\text{Prob}\left\{E[Y] = \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} f(x_i)\right\} = 1$$

Monte Carlo Integration

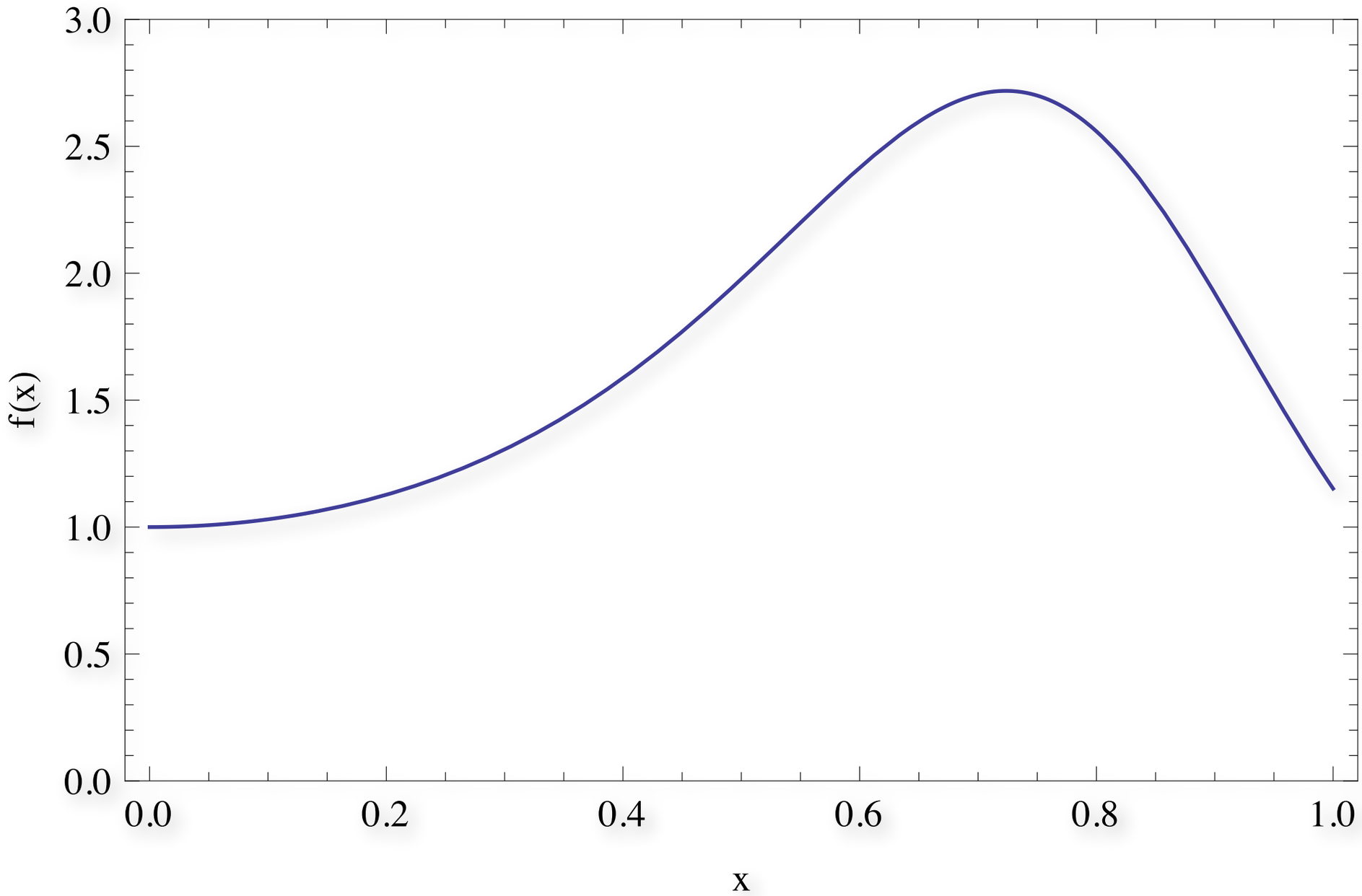
- We want to compute the value F of an integral

$$E[Y] = \int_D f(x)p(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

$$= F = \int_D \left[\frac{f(x)}{p(x)}\right] p(x)\,dx \approx \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)} = F_N$$

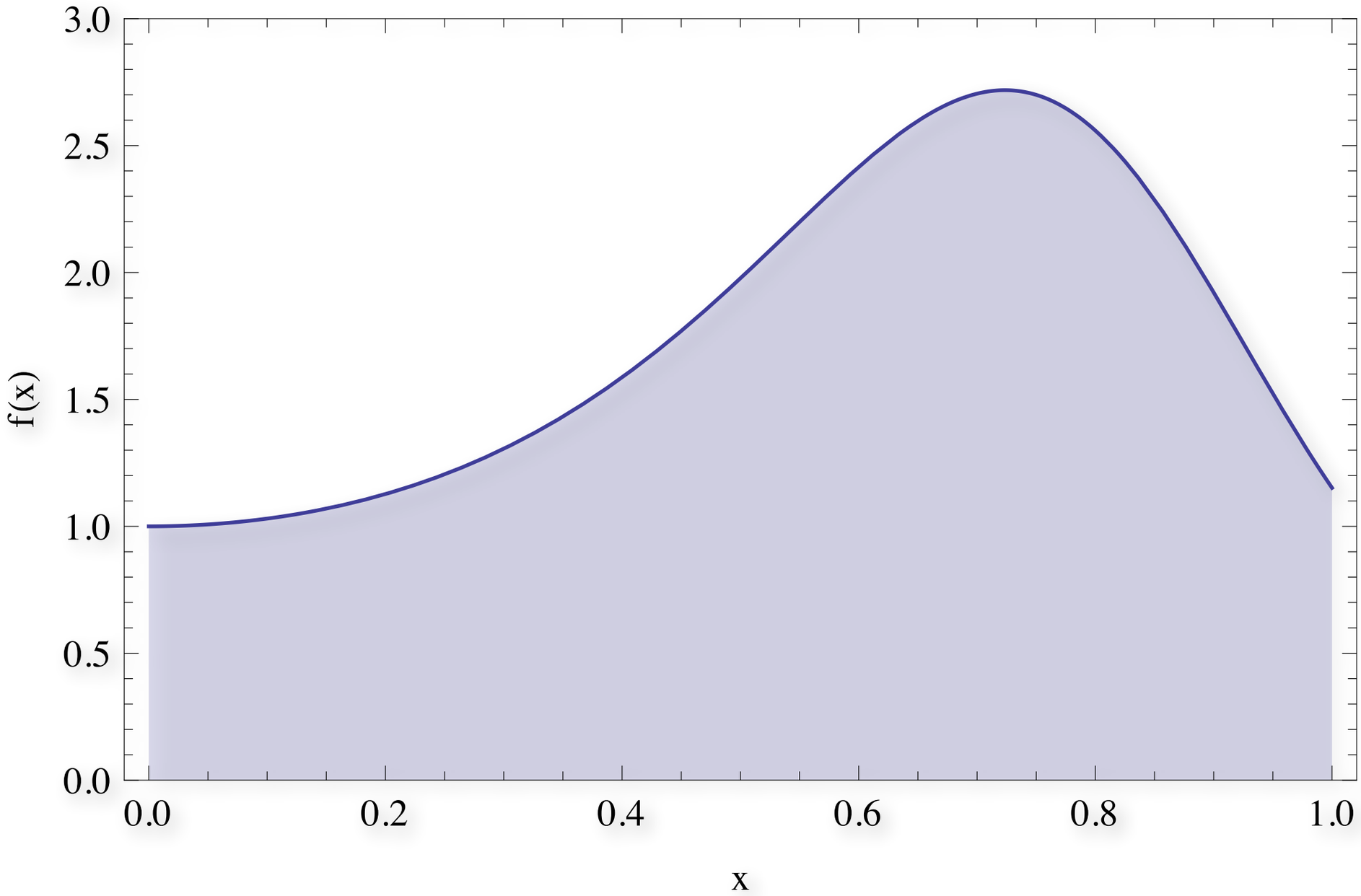
Monte Carlo Integration

$$f(x) = e^{\sin(3x^2)}$$



Monte Carlo Integration

$$F = \int_0^1 e^{\sin(3x^2)}\, dx$$



Monte Carlo Integration

$$F = \int_0^1 e^{\sin(3x^2)}\,dx \approx F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)} \Rightarrow \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

```
double integrate(int N)
{
   double x, sum=0.0;
   for (int i = 0; i < N; ++i) {
      x = randf();
      sum += exp(sin(3*x*x));
   }
   return sum / double(N);
}
```

$$p(x_i) = 1$$

Monte Carlo Integration

$$F = \int_{a}^{b} e^{\sin(3x^2)}\,dx \approx F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}$$

```
double integrate(int N, double a, double b)
{
   double x, sum=0.0;
   for (int i = 0; i < N; ++i) {
      x = randf();
      sum += exp(sin(3*x*x));
   }
   return sum / double(N);
}
```

Monte Carlo Integration

$$F = \int_a^b e^{\sin(3x^2)}\,dx \approx F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}$$

```
double integrate(int N, double a, double b)
{
   double x, sum=0.0;
   for (int i = 0; i < N; ++i) {
      x = a + randf()*(b-a);
      sum += exp(sin(3*x*x));
   }
   return sum / double(N);
}
```

$$p(x_i) = \frac{1}{b-a}$$

Monte Carlo Integration

$$F = \int_a^b e^{\sin(3x^2)}\,dx \approx F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)} \Rightarrow \frac{b-a}{N}\sum_{i=1}^{N} f(x_i)$$

```
double integrate(int N, double a, double b)
{
    double x, sum=0.0;
    for (int i = 0; i < N; ++i) {
        x = a + randf()*(b-a);
        sum += exp(sin(3*x*x));
    }
    return sum * (b-a) / double(N);
}
```

$$p(x_i) = \frac{1}{b-a}$$

Monte Carlo Integration

$$f(x) = e^{\sin(3x^2)}$$

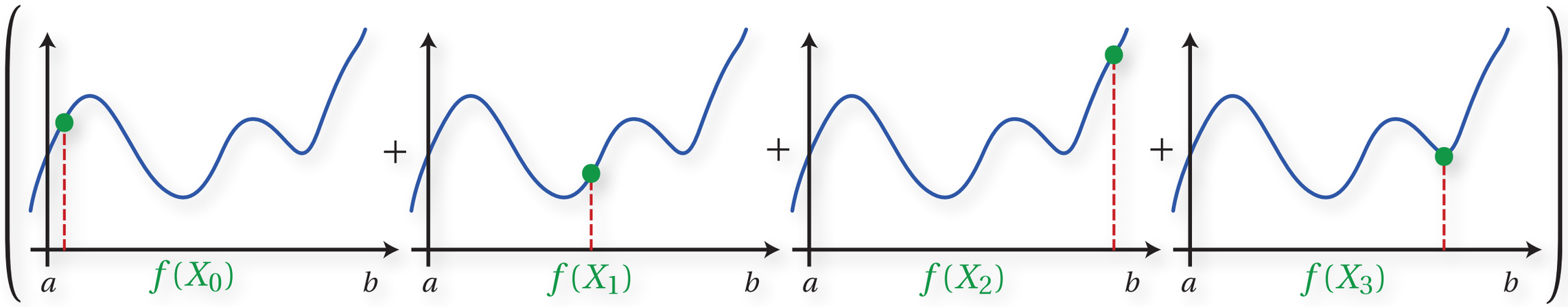
N	F_N
1	2.75039
10	1.9893
100	1.79139
1000	1.75146
10000	1.77313
100000	1.77862

True value: 1.760977217585905…

Monte Carlo Visual Interpretation

Average function value or "height"

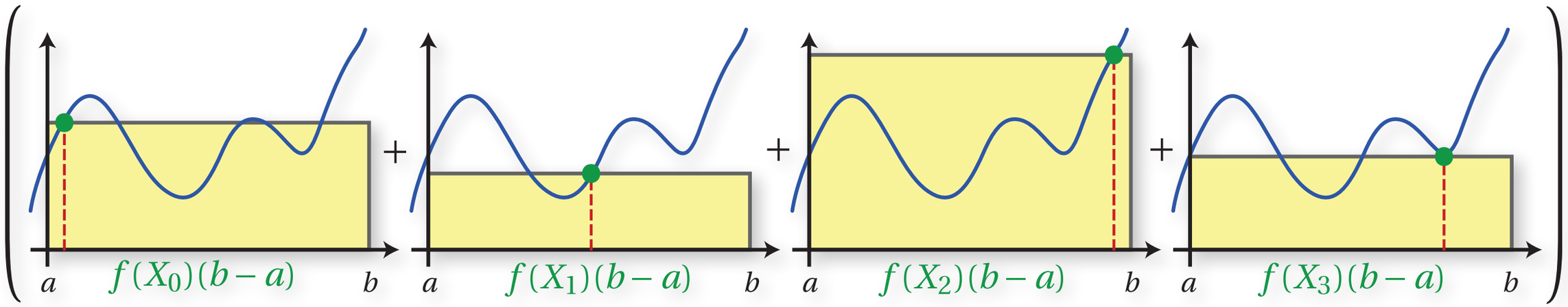
$$(b-a)\left(f(X_0) + f(X_1) + f(X_2) + f(X_3) \right)$$



4

Area of a random box

$$\frac{1}{4}\left(f(X_0)(b-a) + f(X_1)(b-a) + f(X_2)(b-a) + f(X_3)(b-a) \right)$$



Monte Carlo Integration Summary

- Goal: evaluate integral $\int_a^b f(x)dx$

- Random variable $X_i \sim p(x)$

- Monte Carlo Estimator $F_N = \frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}$

- Expectation $E[F_N] = \int_a^b f(x)dx$

Monte Carlo Estimator

- *Unbiased* estimator
  - expected value equals integral
- Extension to higher dimensions straightforward
  - number of samples independent of dimension (cf. standard quadrature)
- Convergence* $O(1/\sqrt{N})$
  - reducing the error by a factor of 2 requires 4 times as many samples!

Today's Agenda

$$F_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)}$$

- Main practical issues:
  - How to choose $p(x)$
  - How to generate x_i according to $p(x)$
- Case Study: **Ambient Occlusion**

$$L_r(\mathbf{x}, \vec{\omega}_r) = \int_{H^2} f_r(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_r) L_i(\mathbf{x}, \vec{\omega}_i) \cos\theta_i \, d\vec{\omega}_i$$

Visual Break

Sampling Random Variables

- Sampling the function domain:
  - Uniform unit interval (0,1)
  - Uniform interval (a,b)
  - Circle?
  - Sphere?
  - Hemisphere?
  - More complex domains?

Example: uniformly sampling a disk

- Uniform probability density on a unit disk

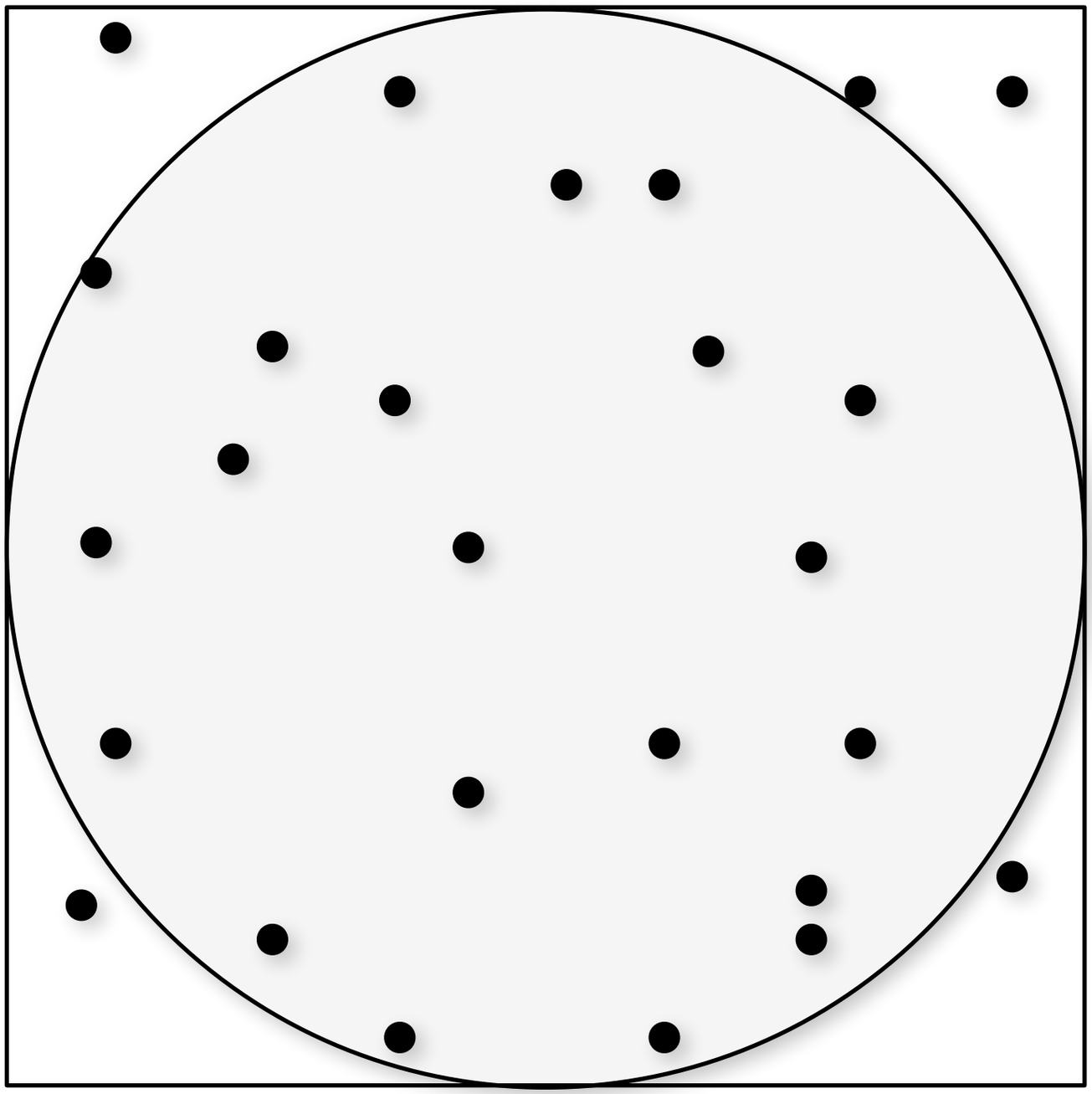
$$p(x,y) = \begin{cases} \frac{1}{\pi} & x^2 + y^2 < 1 \\ 0 & \text{otherwise} \end{cases}$$

- Goal: draw samples X_i,Y_i that are distributed as:

$$(X_i, Y_i) \sim p(x,y)$$

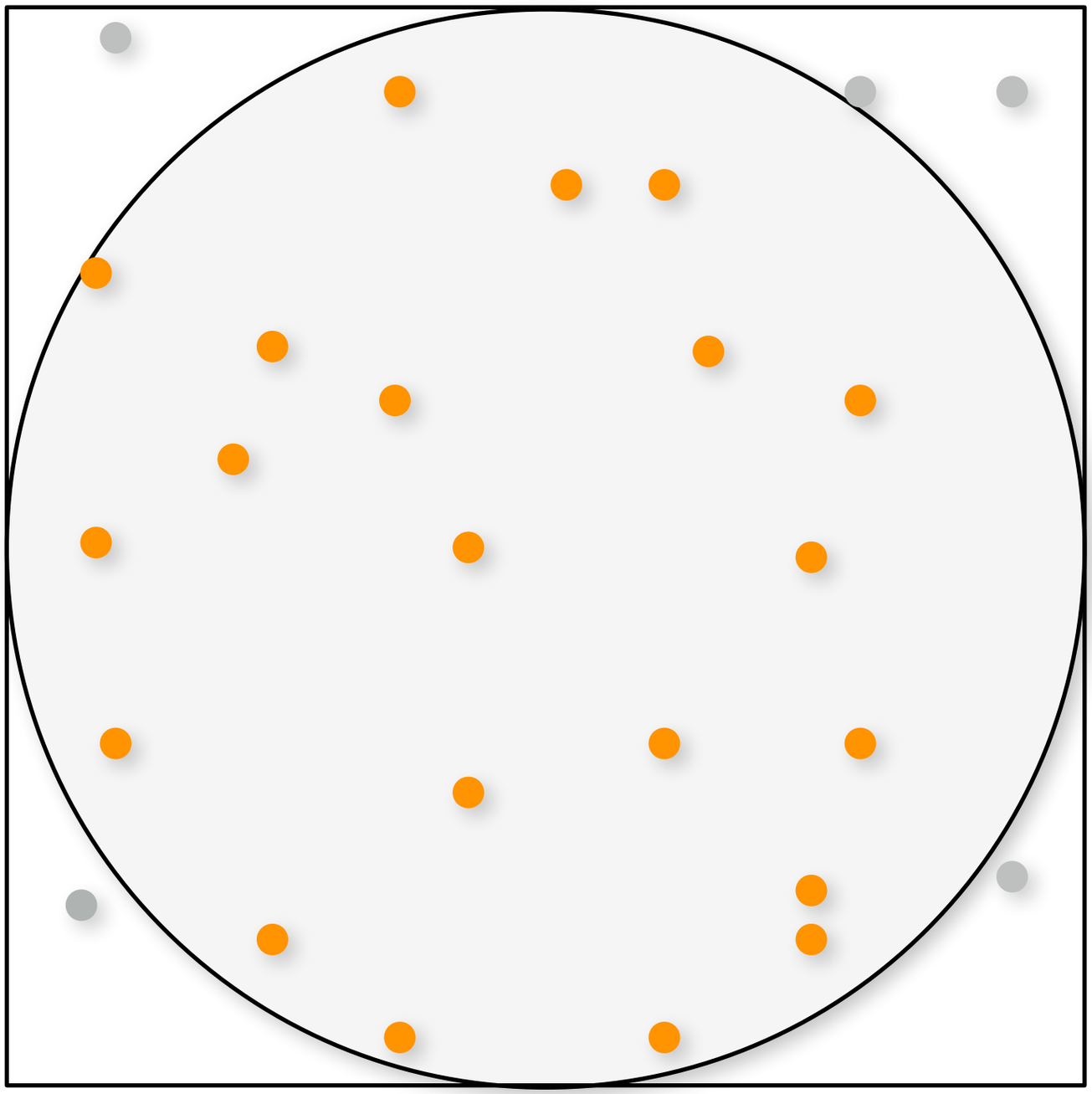
- Problem: pseudo-random number generators only allow us to draw samples from a canonical uniform distribution

Rejection Sampling a Disk



```
Vector2 v;

v.x = 1-2*randf();
v.y = 1-2*randf();
```

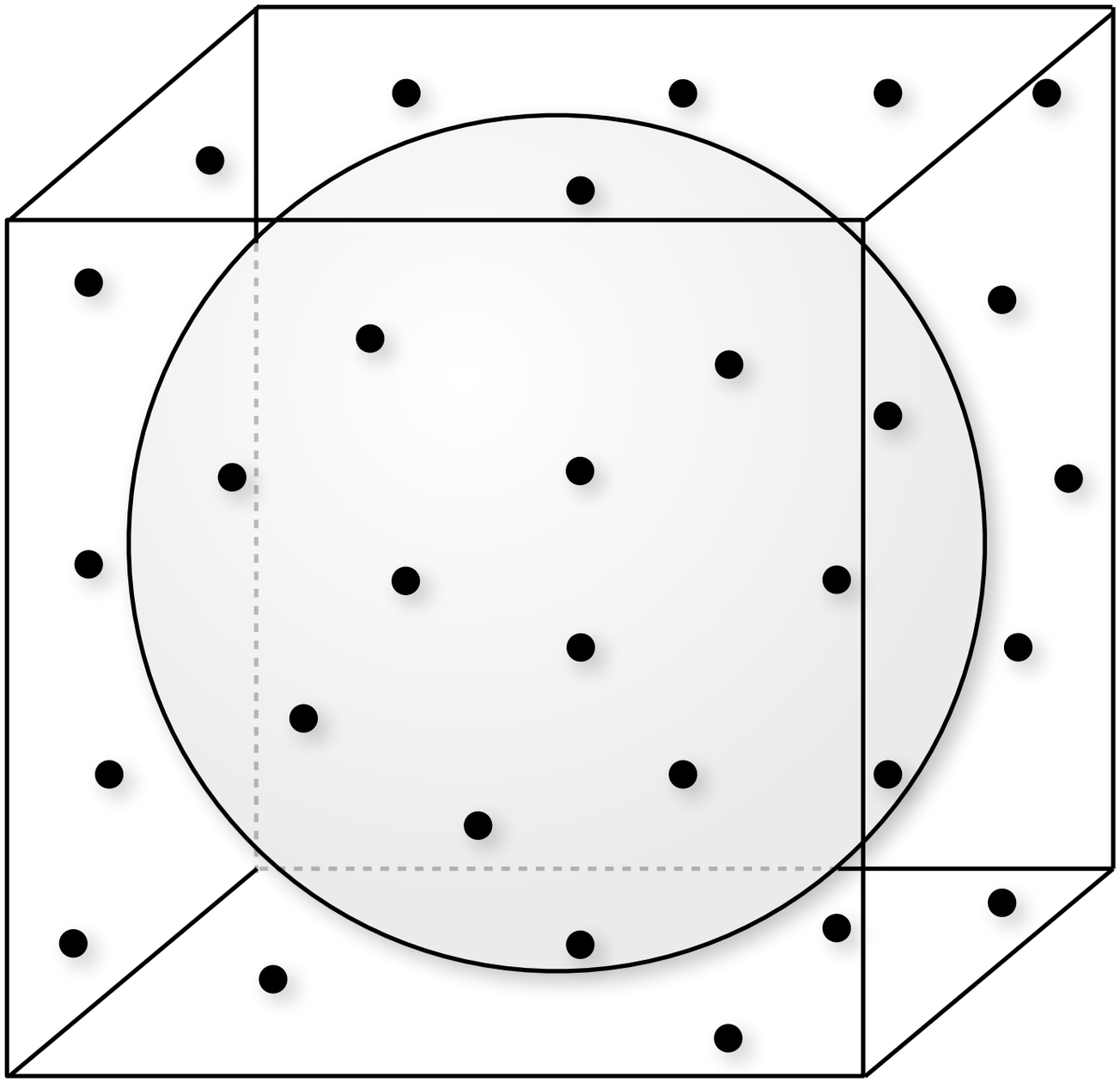
Rejection Sampling a Disk



```
Vector2 v;
do
{
    v.x = 1-2*randf();
    v.y = 1-2*randf();
} while(v.length²() > 1)
```

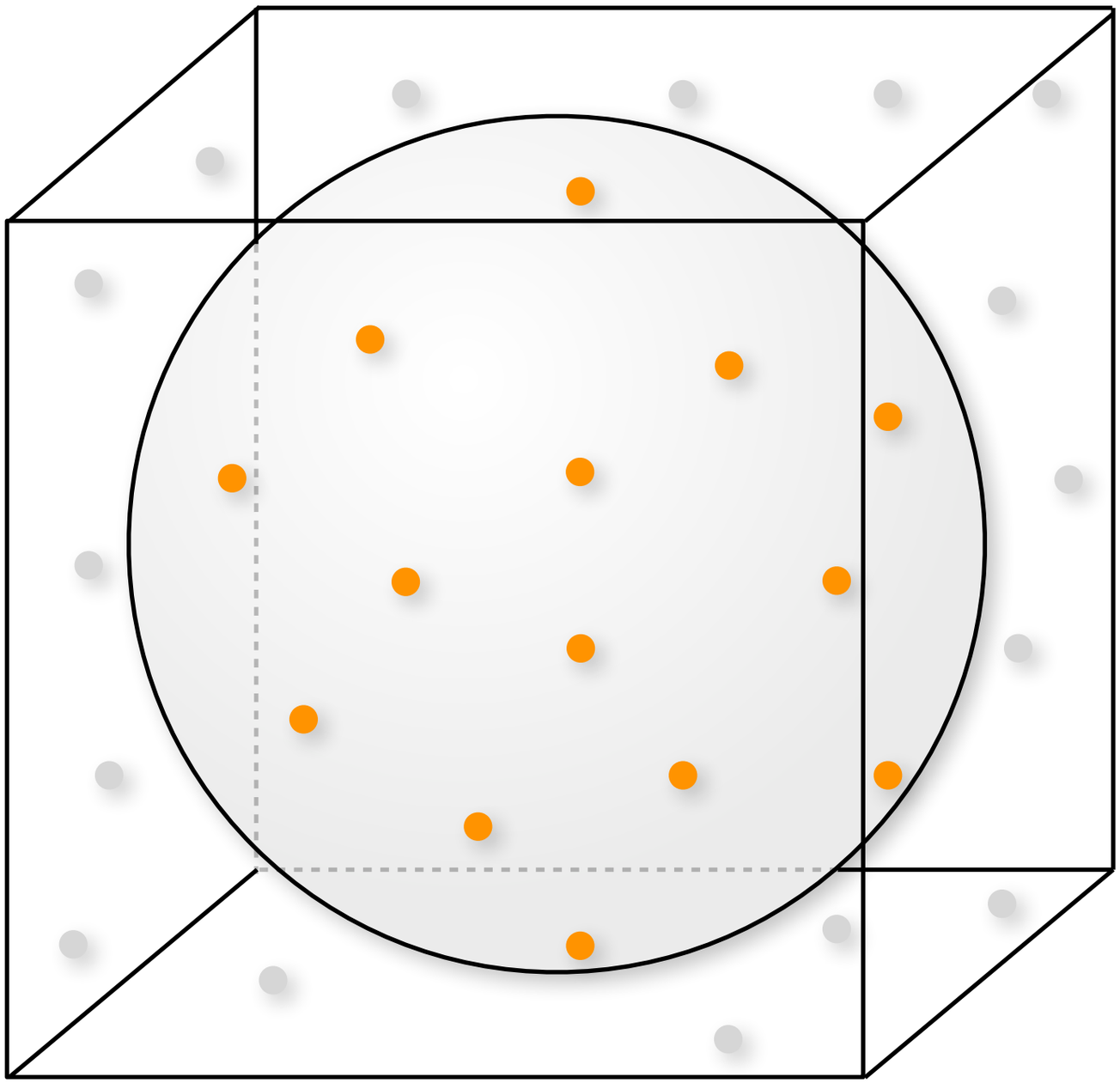
- Similar technique for sampling a sphere

Rejection Sampling a Sphere



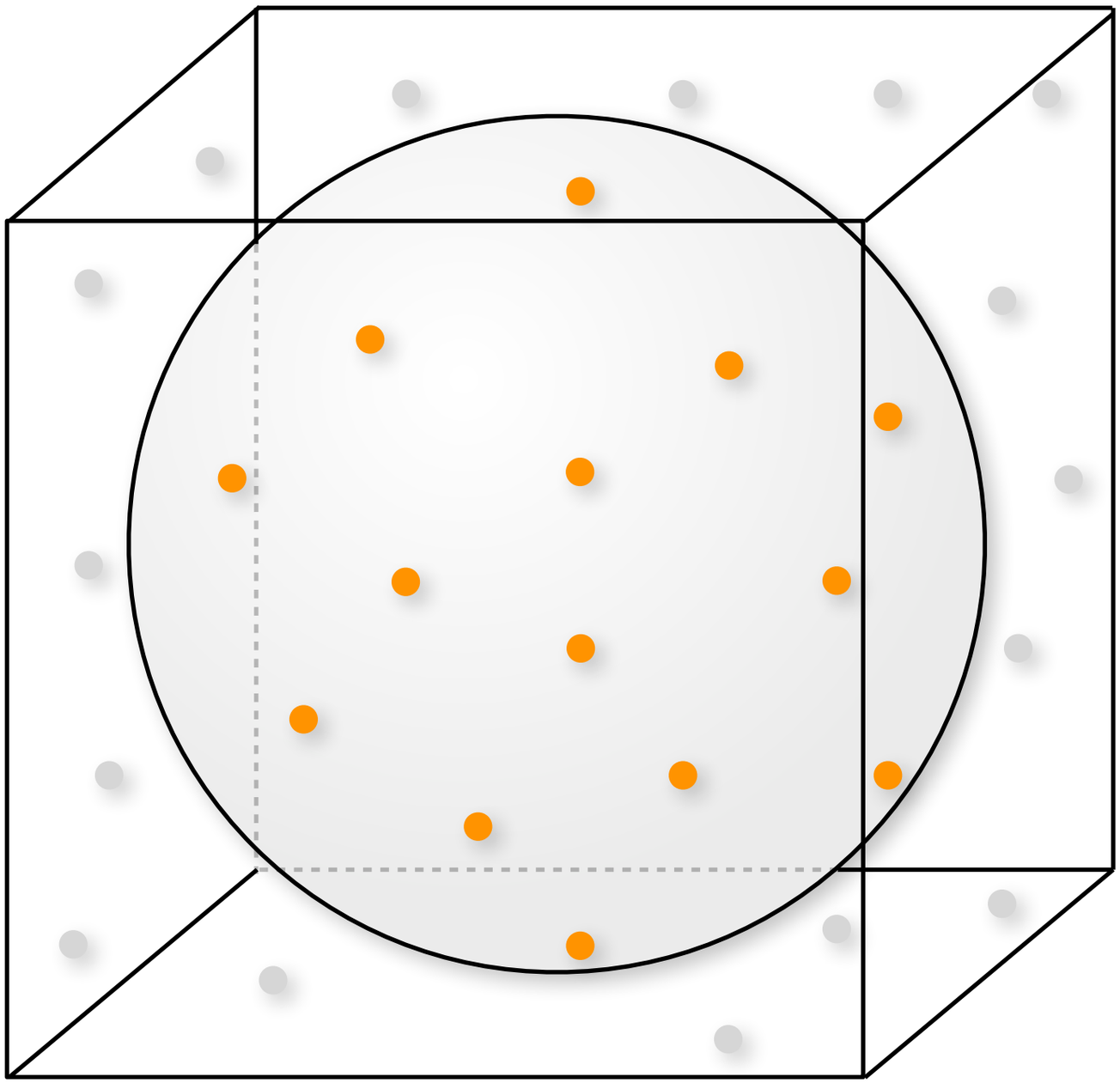
```
Vector3D v;

    v.x = 1-2*randf();
    v.y = 1-2*randf();
    v.z = 1-2*randf();
```

Rejection Sampling a Sphere



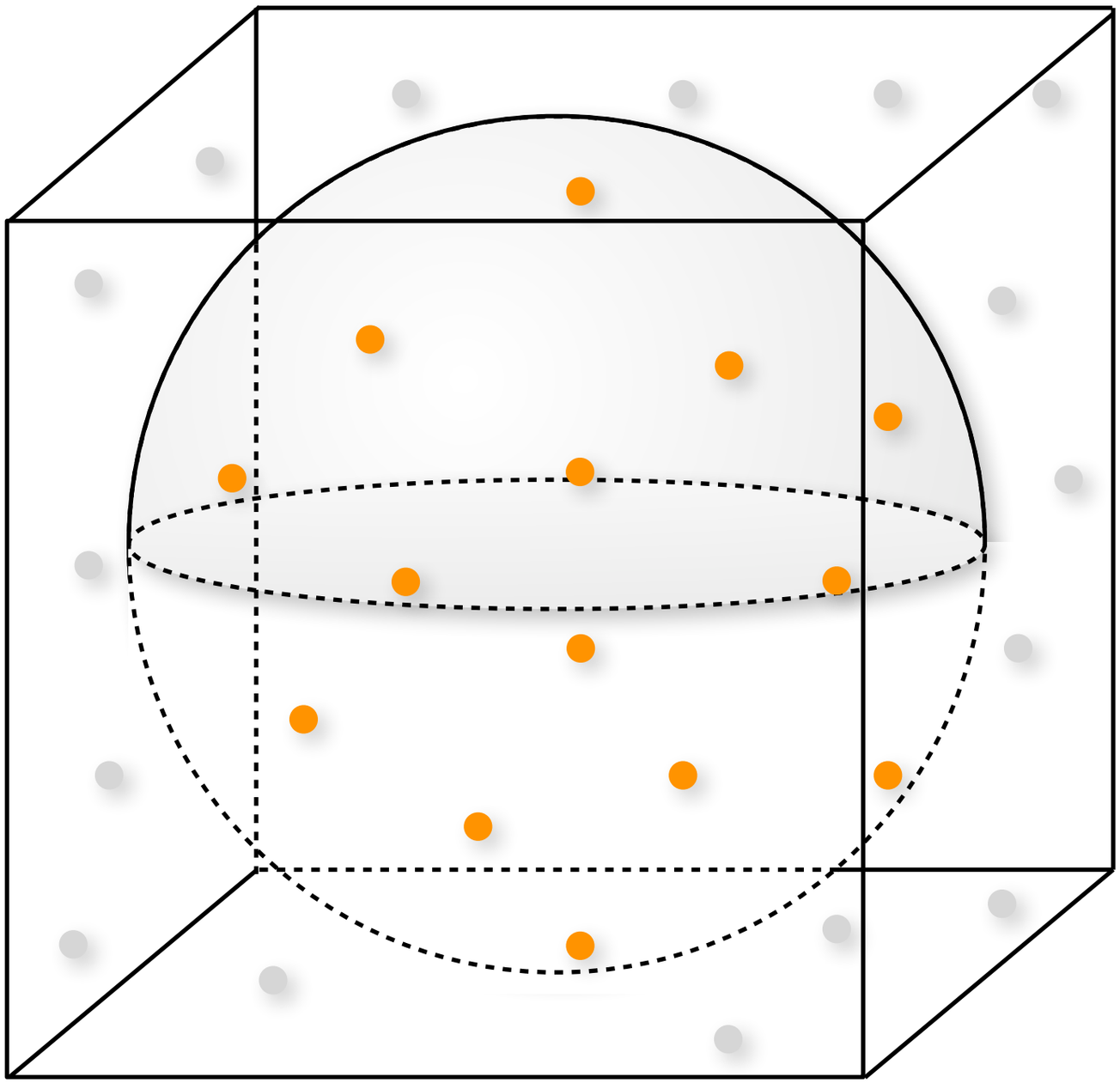
```
Vector3D v;
do
{
    v.x = 1-2*randf();
    v.y = 1-2*randf();
    v.z = 1-2*randf();
} while(v.length²() > 1)
```

Rejection Sampling a Sphere



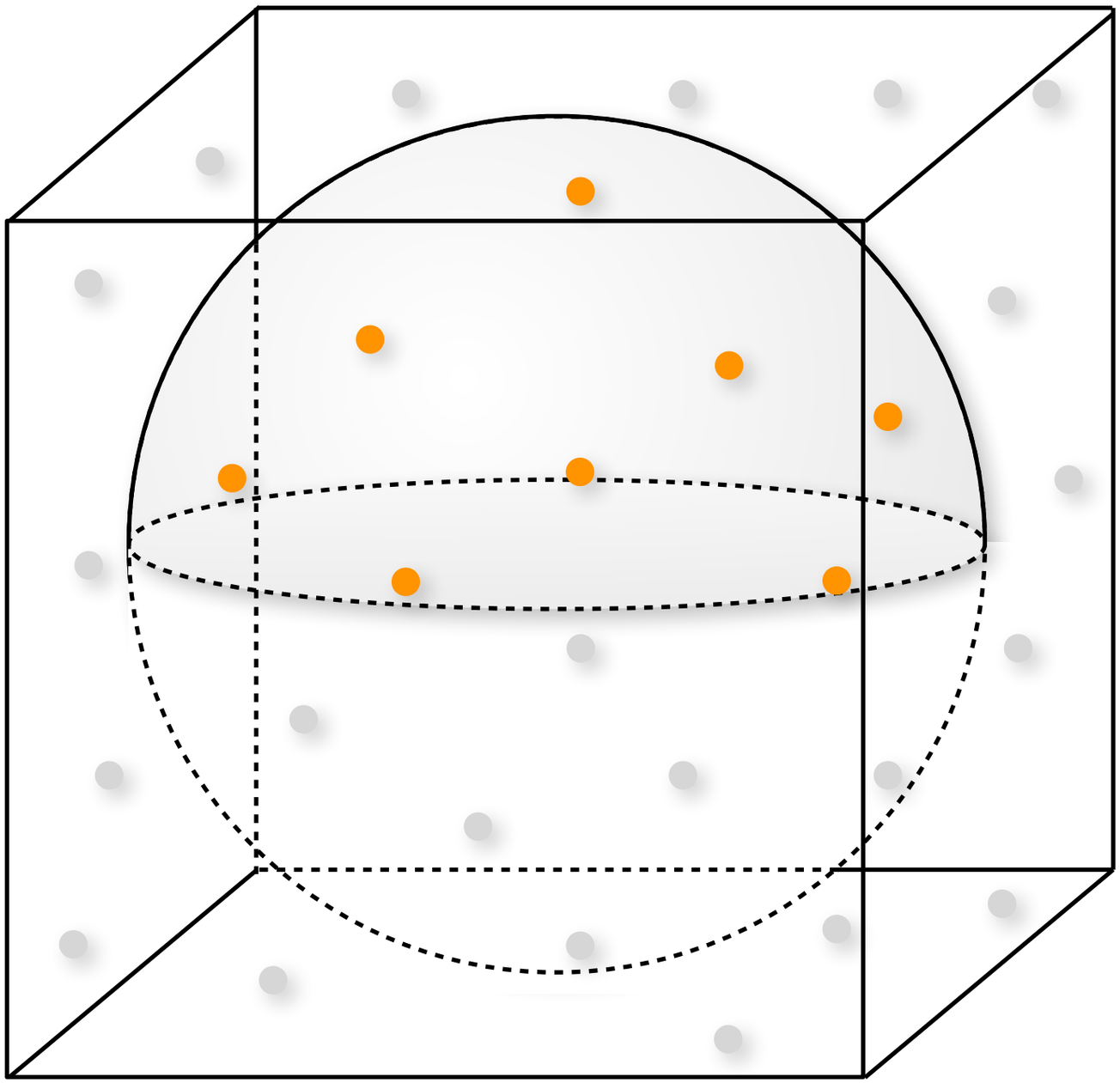
```
Vector3D v;
do
{
    v.x = 1-2*randf();
    v.y = 1-2*randf();
    v.z = 1-2*randf();
} while(v.length²() > 1)

// Project onto sphere
v /= v.length();
```

Rejection Sampling a Hemisphere



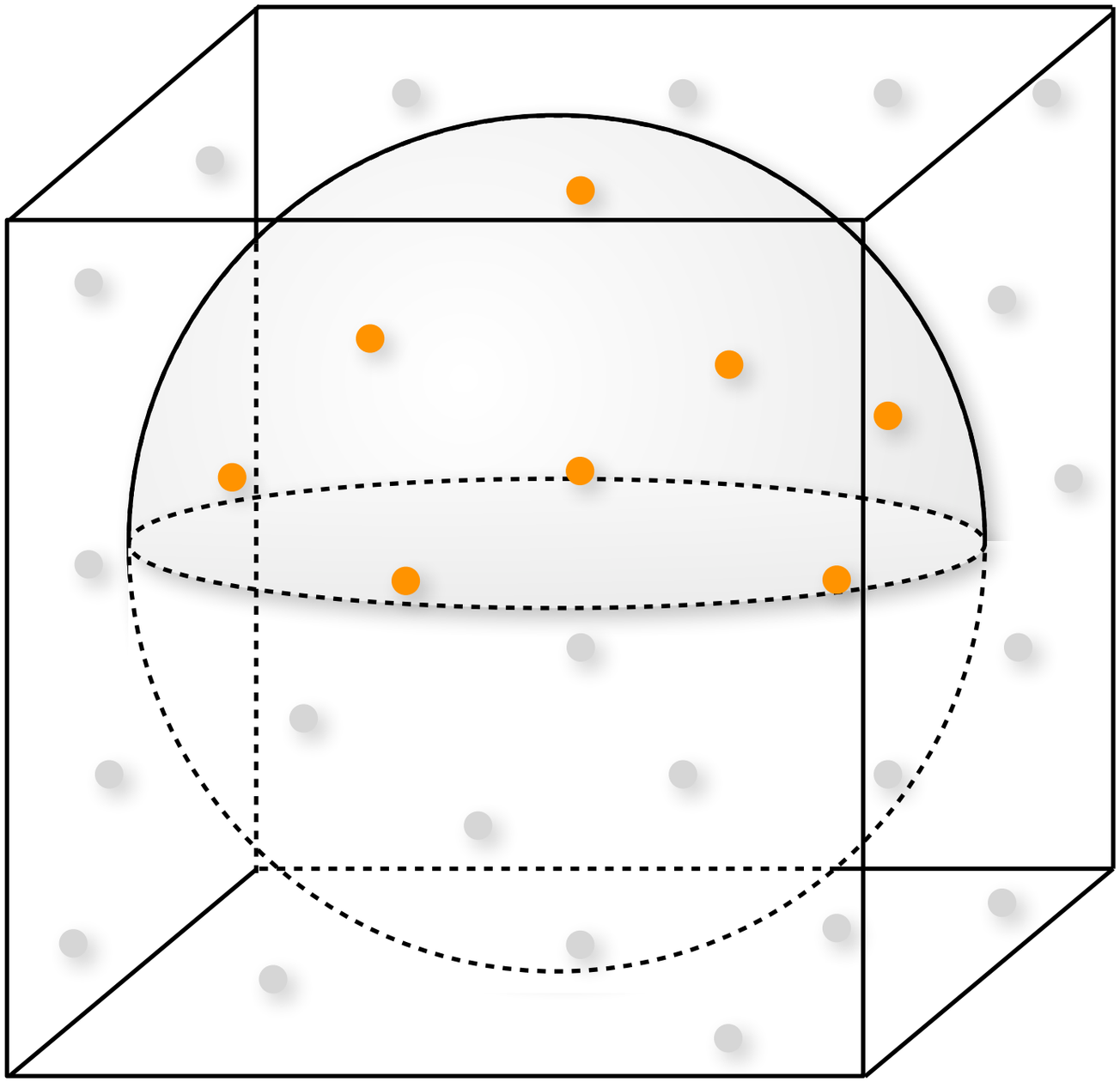
```
Vector3D v;
do
{
    v.x = 1-2*randf();
    v.y = 1-2*randf();
    v.z = 1-2*randf();
} while(v.length²() > 1)
```

Rejection Sampling a Hemisphere



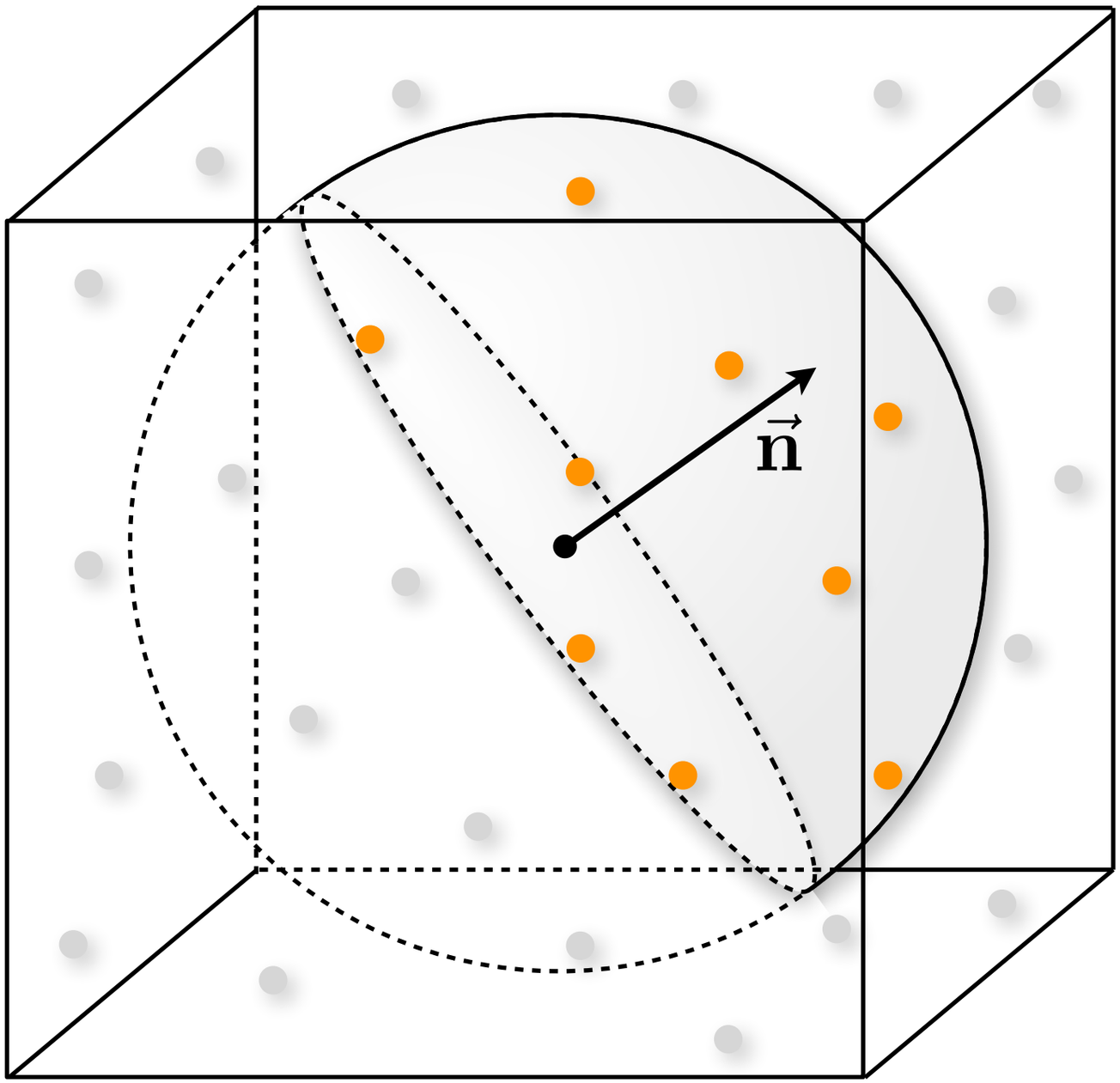
```
Vector3D v;
do
{
    v.x = 1-2*randf();
    v.y = 1-2*randf();
    v.z = 1-2*randf();
} while(v.length²() > 1 ||
        v.z < 0)
```

Rejection Sampling a Hemisphere



```
Vector3D v;
do
{
    v.x = 1-2*randf();
    v.y = 1-2*randf();
    v.z = 1-2*randf();
} while(v.length²() > 1 ||
        v.z < 0)

// Project onto hemisphere
v /= v.length();
```

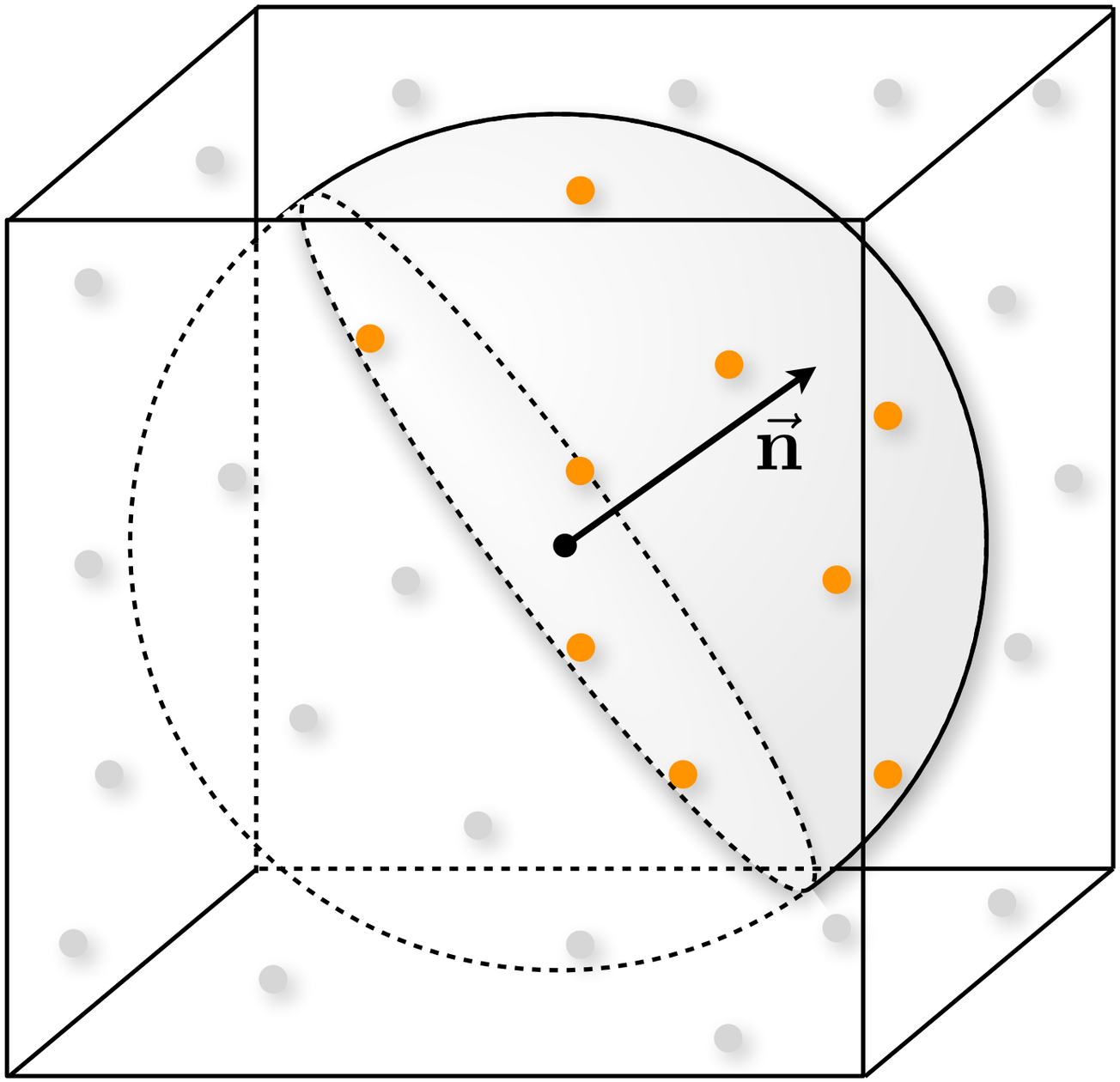
Rejection Sampling a Hemisphere




```
Vector3D v;
do
{
    v.x = 1-2*randf();
    v.y = 1-2*randf();
    v.z = 1-2*randf();
} while(v.length²() > 1 ||
        v.z < 0)

// Project onto hemisphere
v /= v.length();
```

- Arbitrary orientation?

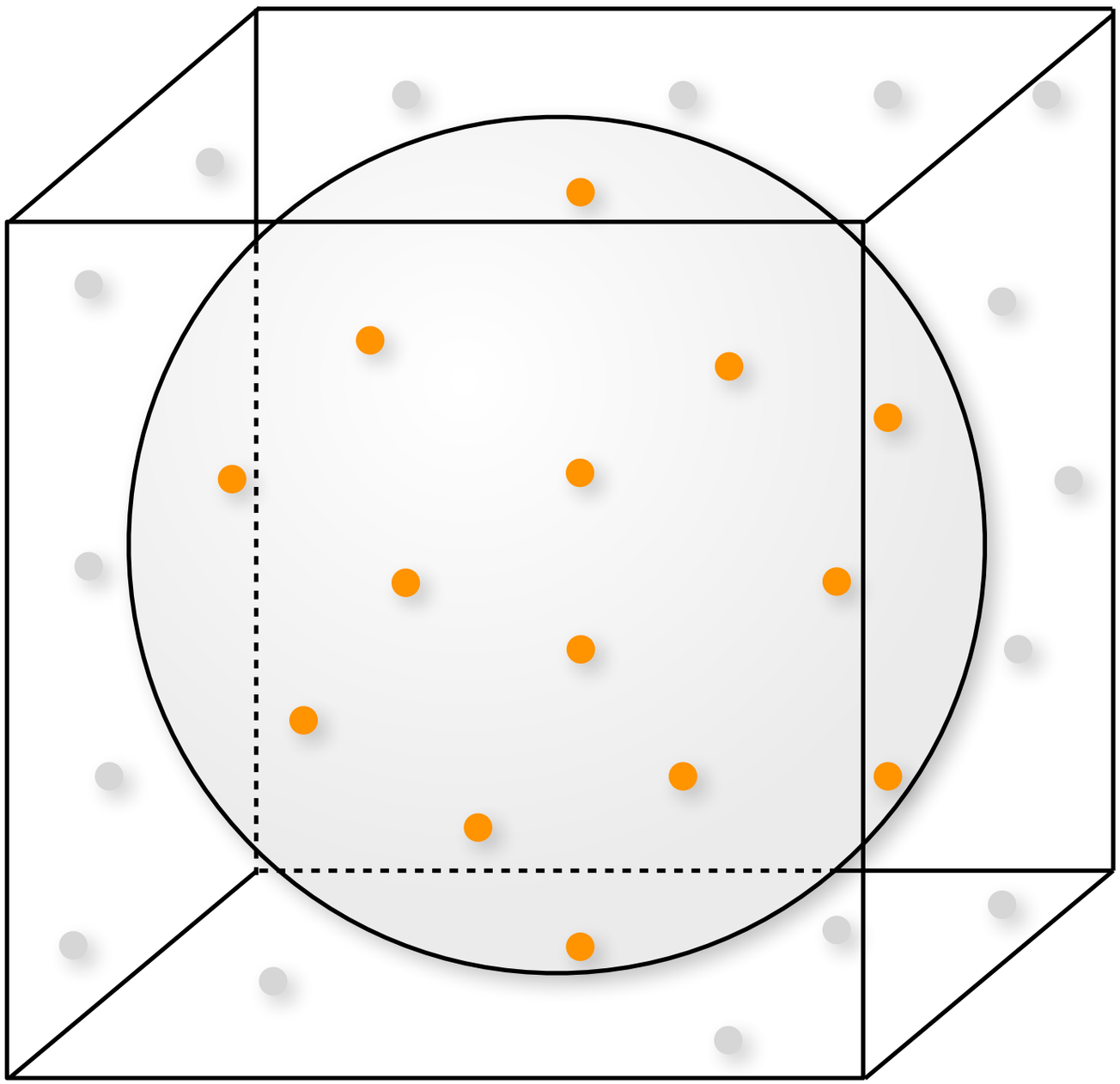
Rejection Sampling a Hemisphere




```
Vector3D v;
do
{
    v.x = 1-2*randf();
    v.y = 1-2*randf();
    v.z = 1-2*randf();
} while(v.length²() > 1 ||
        dot(v,n) < 0)

// Project onto hemisphere
v /= v.length();
```

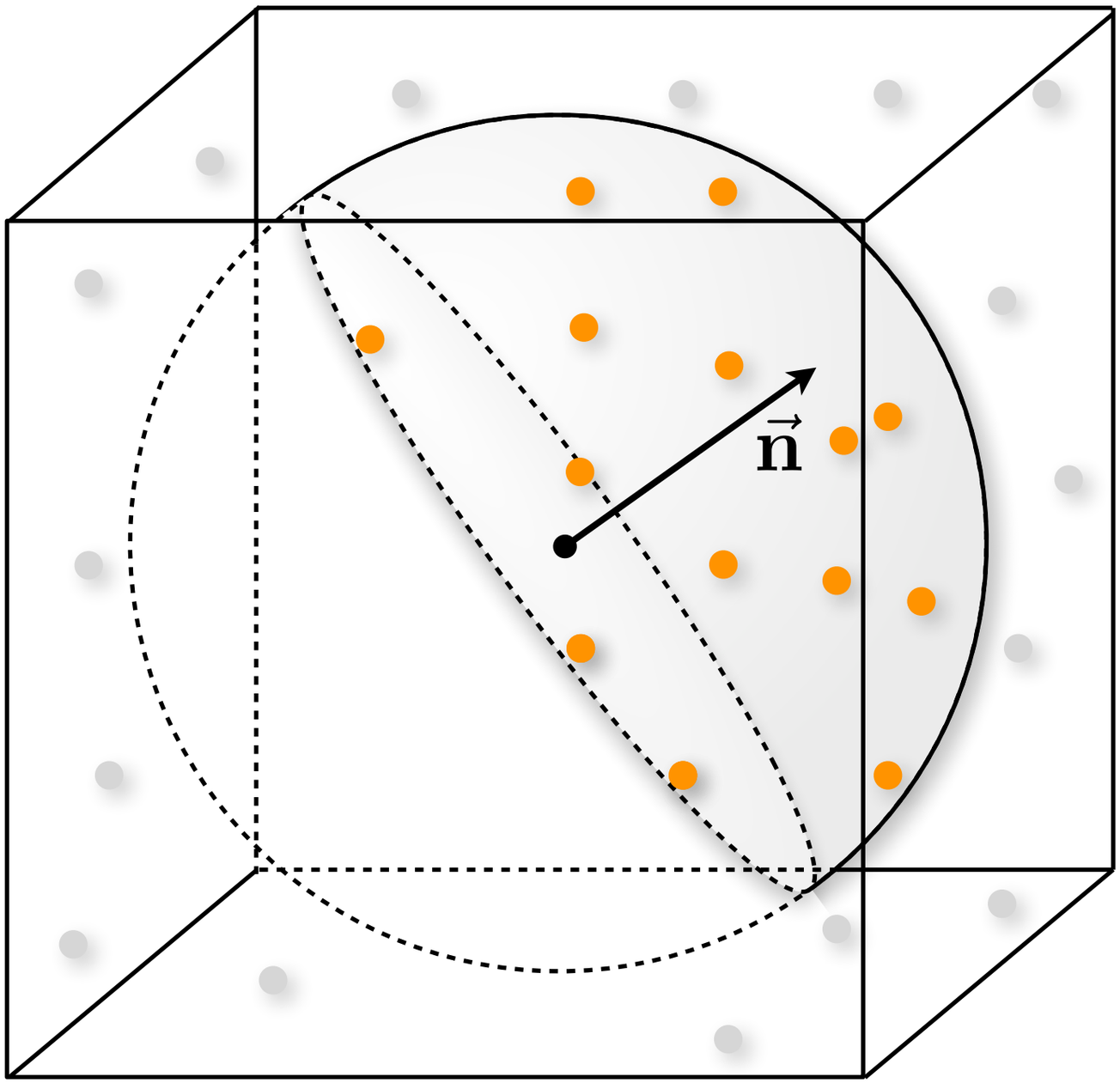
- Arbitrary orientation?
- **Inefficient!**

Rejection Sampling a Hemisphere



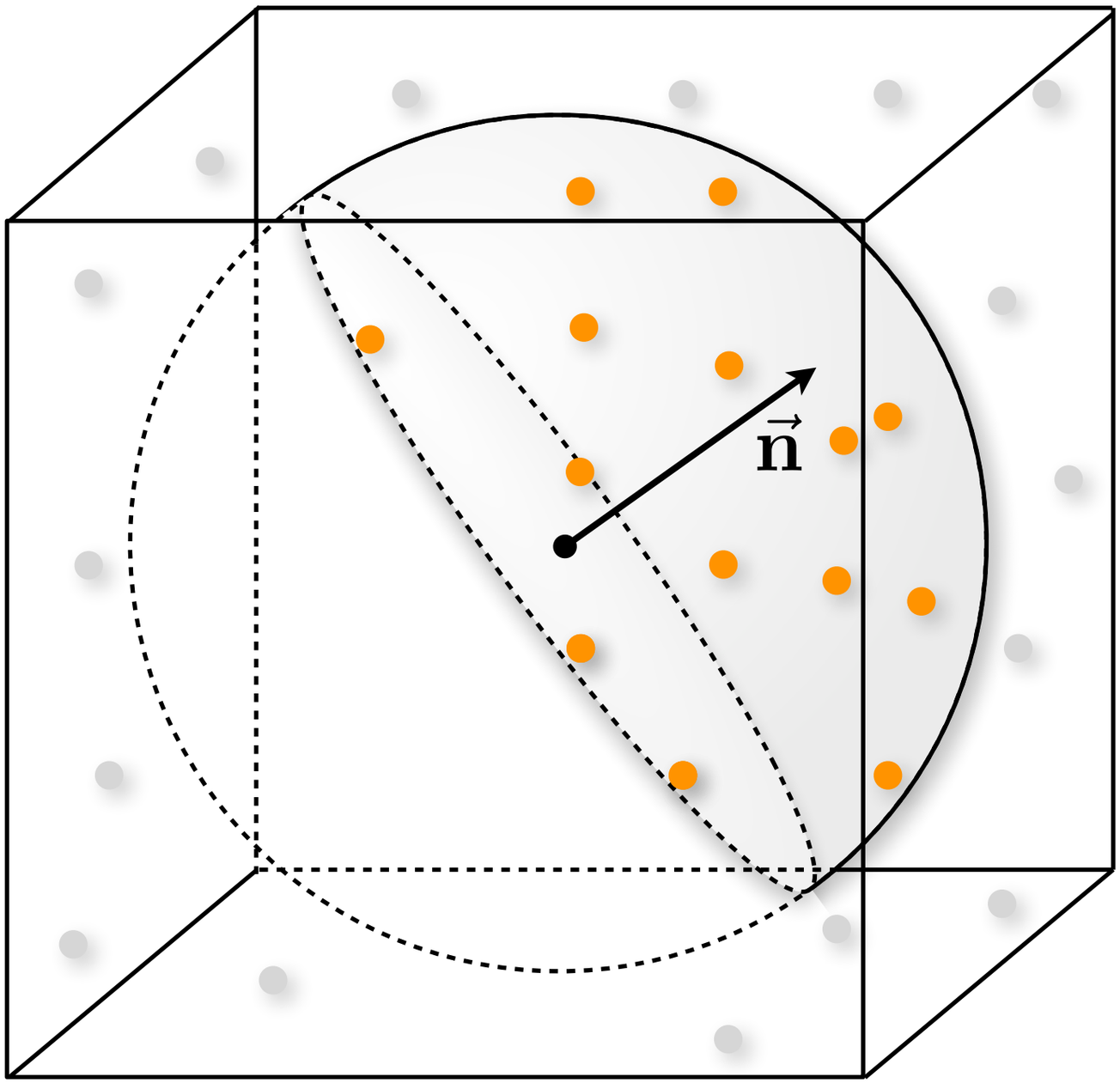
```
Vector3D v;
do
{
    v.x = 1-2*randf();
    v.y = 1-2*randf();
    v.z = 1-2*randf();
} while(v.length²() > 1)
```

Rejection Sampling a Hemisphere




```
Vector3D v;
do
{
    v.x = 1-2*randf();
    v.y = 1-2*randf();
    v.z = 1-2*randf();
} while(v.length²() > 1)

// flip to proper hemisphere
if (dot(v,n) < 0)
    v = -v;
```

Rejection Sampling a Hemisphere

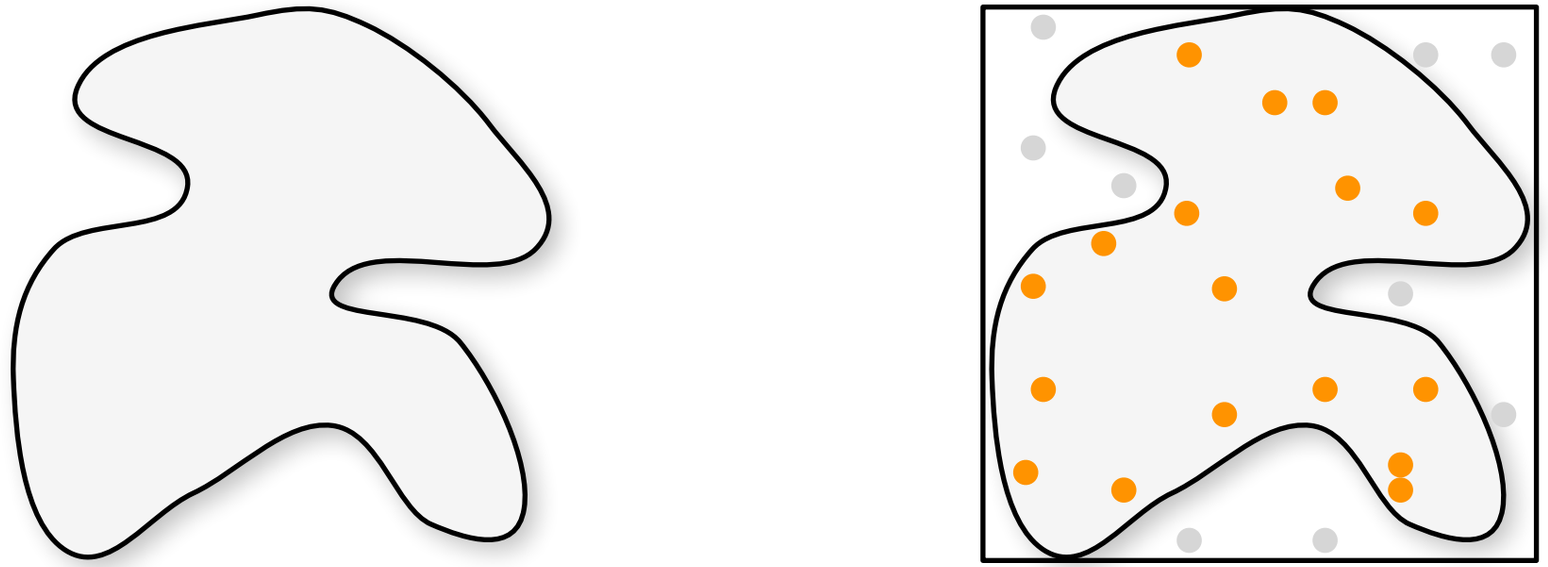



```
Vector3D v;
do
{
    v.x = 1-2*randf();
    v.y = 1-2*randf();
    v.z = 1-2*randf();
} while(v.length²() > 1)

// flip to proper hemisphere
if (dot(v,n) < 0)
    v = -v;

// Project onto hemisphere
v /= v.length();
```

- Or, generate in canonical orientation, and then rotate

Rejection Sampling

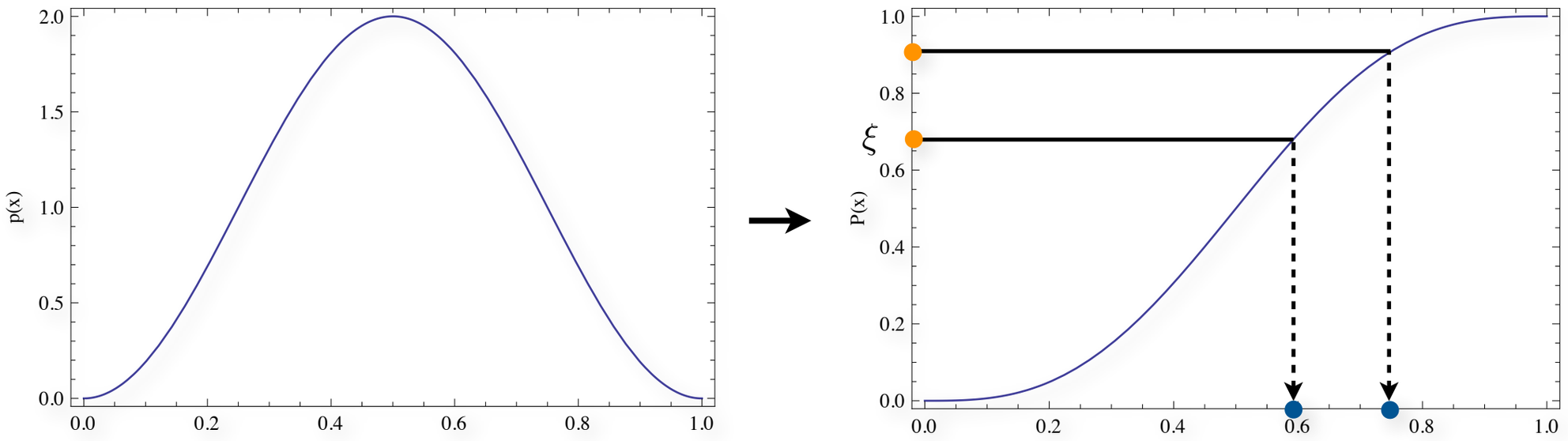
- More complex shapes



- Pros:
  - + Flexible
- Cons:
  - – Inefficient
  - – Difficult to combine with stratification or quasi-Monte Carlo

Sampling Arbitrary Distributions

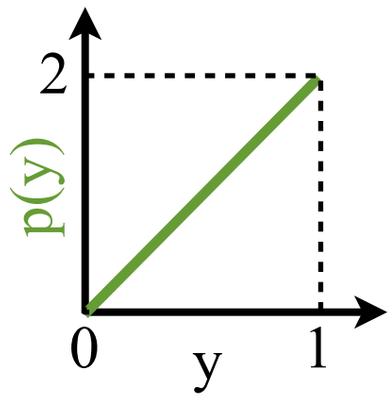
- The inversion method:
  1. Compute the CDF $P(x) = \int_0^x p(x')\, dx'$
  2. Compute its inverse $P^{-1}(x)$
  3. Obtain a uniformly distributed random number ξ
  4. Compute $X_i = P^{-1}(\xi)$



Example – Sampling a Linear Ramp

- Goal: sample with PDF

$$p(y) = 2y$$



$$P(y) = y^2$$

$$P^{-1}(y) = \sqrt{y}$$

$$Y_i = \sqrt{\xi}$$

Sampling 2D Distributions

- Draw samples (X, Y) from a 2D distribution $p(x, y)$
- If $p(x, y)$ is separable, i.e., $p(x,y) = p_x(x)\, p_y(y)$, we can independently sample $p_x(x)$, and $p_y(y)$
- Otherwise, compute the marginal density function:

$$p(x) = \int p(x, y)\, dy$$

- and, the conditional density:

$$p(y|x) = \frac{p(x, y)}{p(x)}$$

- Procedure: first sample $X_i \sim p(x)$, then $Y_i \sim p(y|x)$

Visual Break