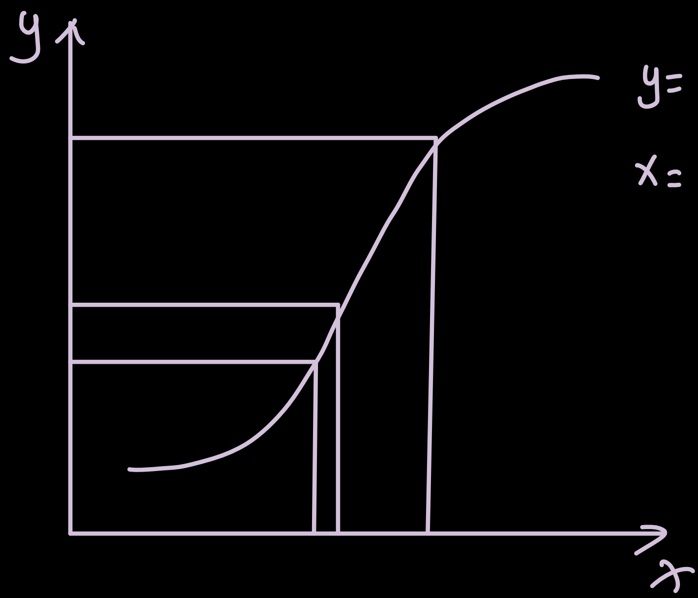
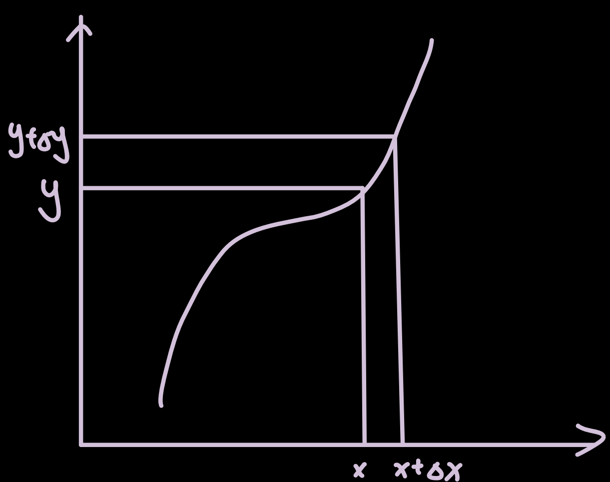
Transformation



$y = h(x)$

$x = h^{-1}(y) = g(y)$



$P(X \in (x, x+\Delta x))$

$= P(Y \in (y, y+\Delta y))$

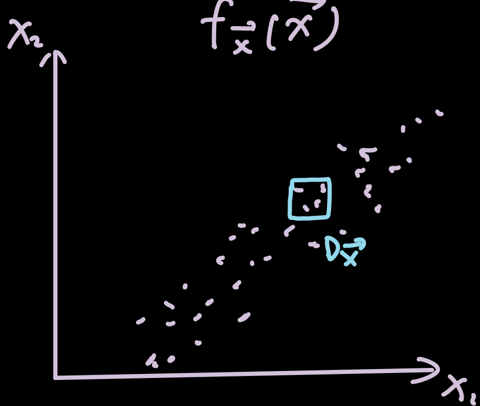
$f_X(x)\Delta x = f_Y(y)\Delta y$

$\frac{\Delta x}{\Delta y} = |g'(y)|$

Multivariate

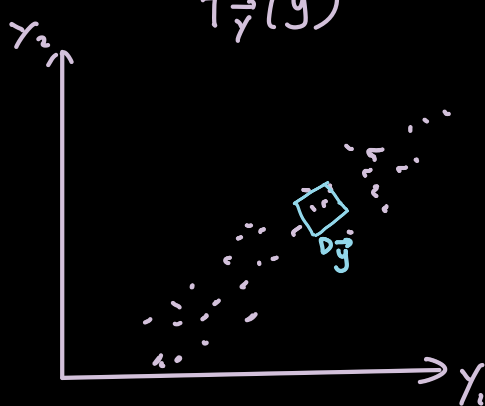
$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ $\vec{y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$

$f_{\vec{x}}(\vec{x})$



$\xrightarrow[\vec{x} = g(\vec{y})]{\vec{y} = h(\vec{x})}$

$f_{\vec{y}}(\vec{y})$



$$P(\vec{x} \in D_{\vec{x}}) = P(\vec{y} \in D_{\vec{y}})$$

$$f_{\vec{x}}(\vec{x})\,|D_{\vec{x}}| = f_{\vec{y}}(\vec{y})\,|D_{\vec{y}}|$$

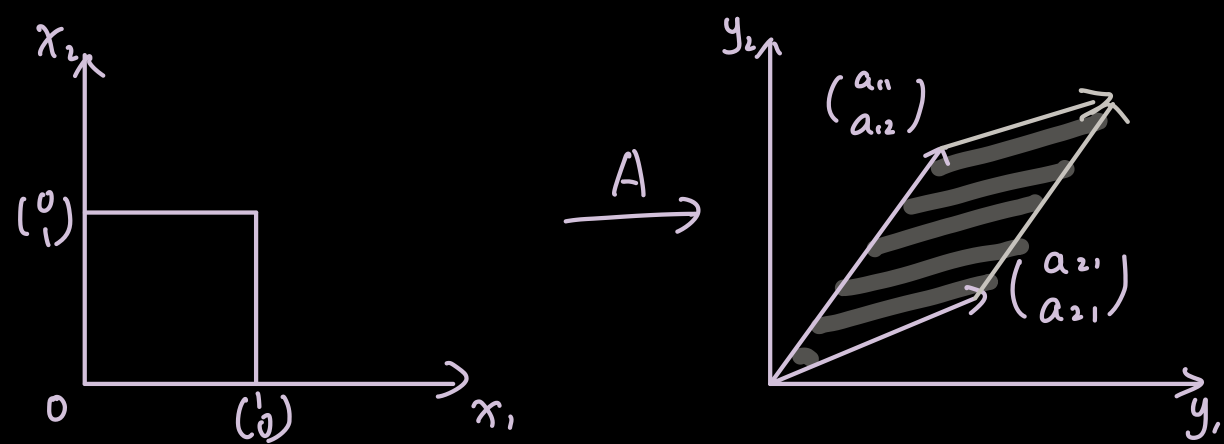
$\downarrow$ size

$\dfrac{|D_{\vec{x}}|}{|D_{\vec{y}}|}$ is Calculated from Jacobian

Linear Transformation

$$\vec{y} = A\vec{x}$$

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$



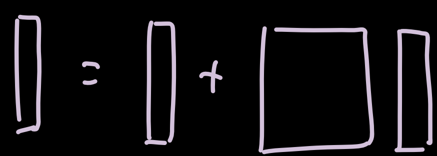
area = 1 space warping area = det(A)

Non-linear Transformation

$\vec{y} = h(\vec{x})$

First-order Taylor Expansion

$h(\vec{x} + \Delta\vec{x}) = h(\vec{x}) + h(\vec{x})\,\Delta\vec{x}$



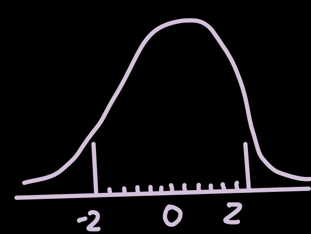
$$\boxed{J} = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} \end{pmatrix}$$

$$\frac{|D\vec{y}|}{|D\vec{x}|} = |\det(J)|$$

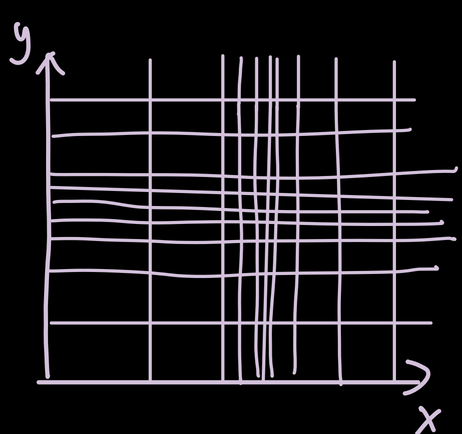
Polar Method

more details

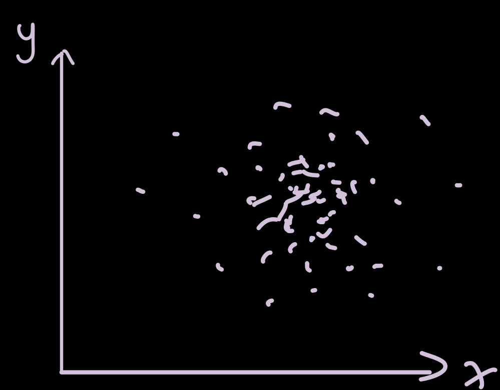
$(X, Y) \sim N(0, 1)$ independent



N points



N^2 points in 2D



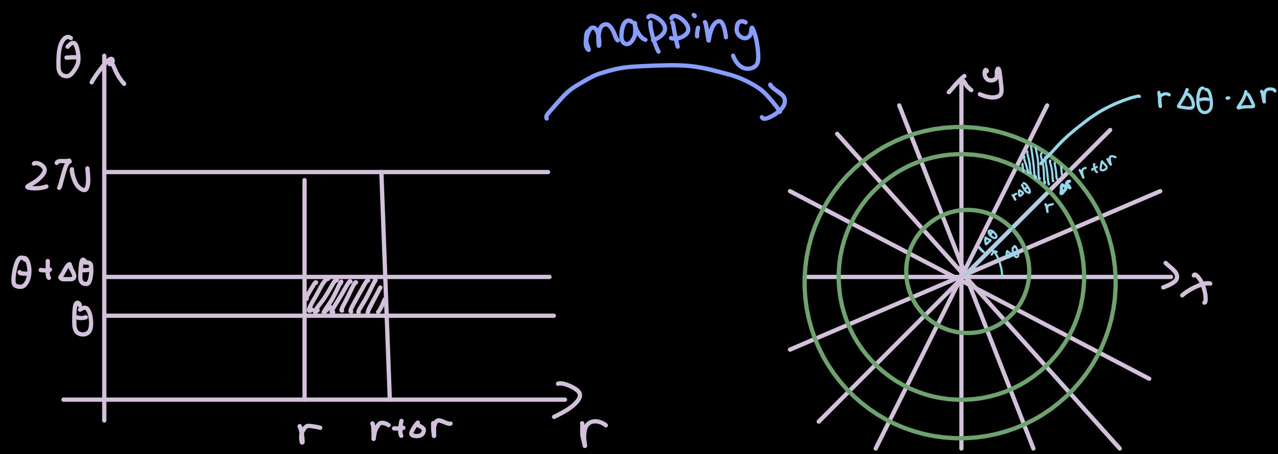
$f(x, y)$

Application: Wave function in electron

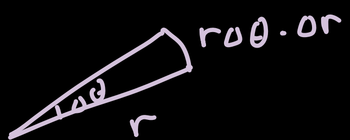
$\varphi(x, y)$ $\qquad f(x,y) = |\varphi(x,y)|^2$

$$f(x, y) = f_x(X)\, f_y(y) = \frac{1}{2\pi} e^{-\frac{x^2+y^2}{2}} \xrightarrow[\text{transformation}]{\text{Polar}} \begin{cases} x = r\cos\theta \\ y = r\sin\theta \end{cases}$$

mapping



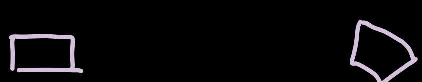
arc length : $\Delta r \Delta\theta$



Algebraic

$$J = \begin{pmatrix} \frac{\partial x}{\partial y} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{pmatrix}$$

$$\det(J) = r$$



$$f_{R\theta}(r, \theta)\, \Delta r\, \Delta\theta = f_{XY}(x, y)\, \Delta x\, \Delta y$$

$$f(r, \theta) = f(x,y) \frac{\Delta x \Delta y}{\Delta r \Delta\theta}$$

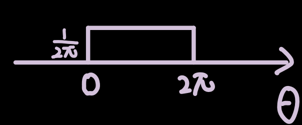
$$= f(x, y)\, r$$

$$= \frac{1}{2\pi} e^{-\frac{r^2}{2}} r$$

both can be easily generated

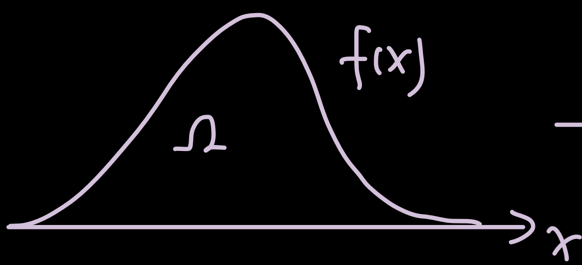
proportion of points in an area

density of both x and y is equal to 1.

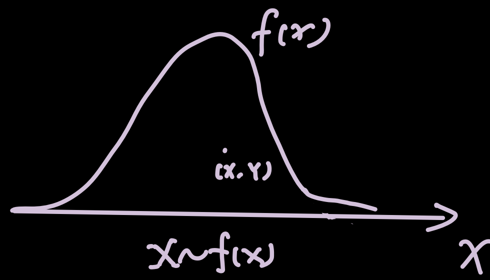


Acceptance-Rejection Sampling

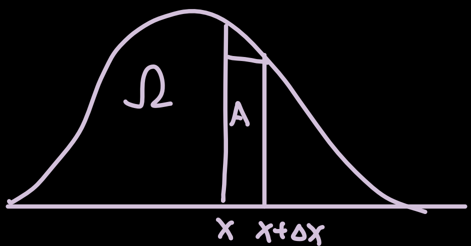
$X \sim f(x)$



$\xrightarrow[\text{sampling}]{\text{random}}$



population of 2D probability



$$P(X \in (x, x+\Delta x)) = P(A)$$

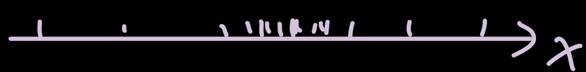
$$= \frac{|A|}{|\Omega|}$$

$$= \frac{f(x)\Delta x}{1}$$

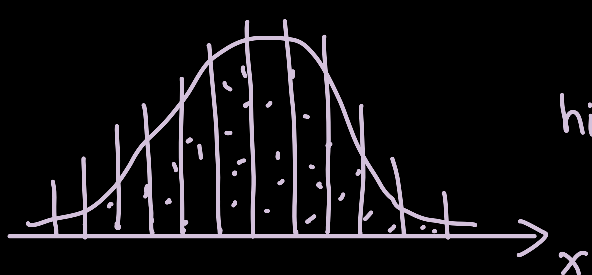
$$= f(x)\Delta x$$

$$f(x) = \frac{P(X \in (x, x+\Delta x))}{\Delta x}$$

Connection to previous point of view



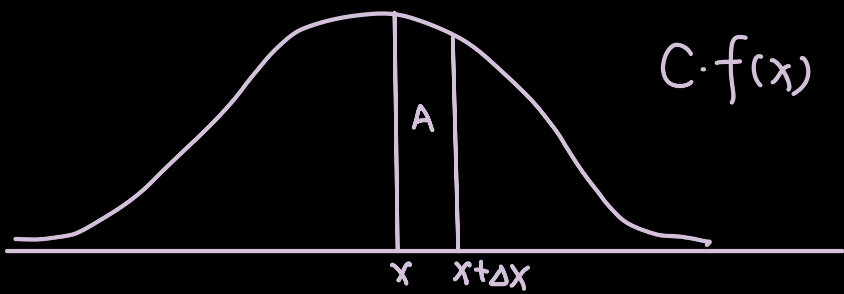
each point (dimensionless) to a small ball



histogram

transformation changes bin sizes

More generally.



$C \cdot f(x)$ C is a constant

$$P(X \in (x, x+\Delta x)) = P(A) = \frac{|A|}{|\Omega|}$$

$$= \frac{C f(x) \Delta x}{C}$$

$$= f(x) \Delta x$$

Rejection Sampling

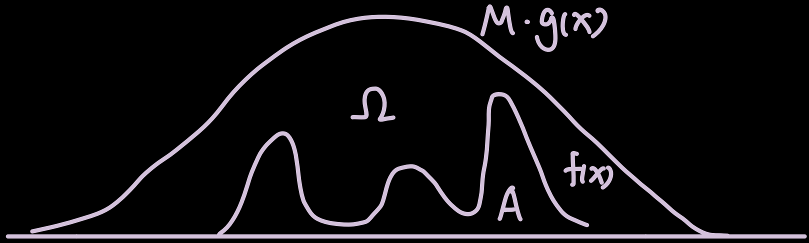
want $X \sim f(x)$

Conditions:

① Can sample $g(x)$ easily.

② $Mg(x)$ envelopes $f(x)$

$$f(x) \le Mg(x)$$



Sample $(X, Y) \sim \mathrm{Unif}($ 

$)$

$X \sim g(x)$

$Y \sim \mathrm{Unif}[0, Mg(X)]$

If $(X, Y) \in A$, accept, return X.

...... $\notin A$, reject, does not return anything

How do we determine if $(X, Y) \in A$?

$$(X, Y) \in A$$

$$\Longleftrightarrow \quad Y \leq f(X)$$

$$\Longleftrightarrow \quad Mg(x)\,U \leq f(X)$$

$$\Longleftrightarrow \quad U \leq \frac{f(X)}{Mg(X)}$$

Algorithm

Generate $X \sim g(x)$

If $U \leq \dfrac{f(X)}{Mg(X)}$, return X

Else reject, return nothing

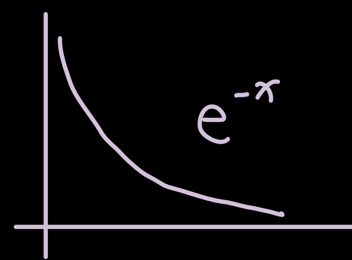
Probability of accepting $X = \dfrac{f(X)}{Mg(X)}$

Overall probability of acceptance $= \dfrac{|A|}{|\Omega|} = \dfrac{1}{M}$

make M as small as possible to max prob.

$f(x) = \sqrt{\frac{2}{\pi}}\, e^{-\frac{x^2}{2}}$

$g(x) = e^{-x}$



$\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$

$X = -\log U$

$$f(x) \le M g(x) \quad \forall x$$

$$\frac{f(x)}{g(x)} \le M \quad \forall x$$

$\Rightarrow$ want: $\displaystyle\max_{x} \frac{f(x)}{g(x)}$

$$\frac{f(x)}{g(x)} = \sqrt{\frac{2}{\pi}}\, e^{-\frac{x^2}{2}+x}$$

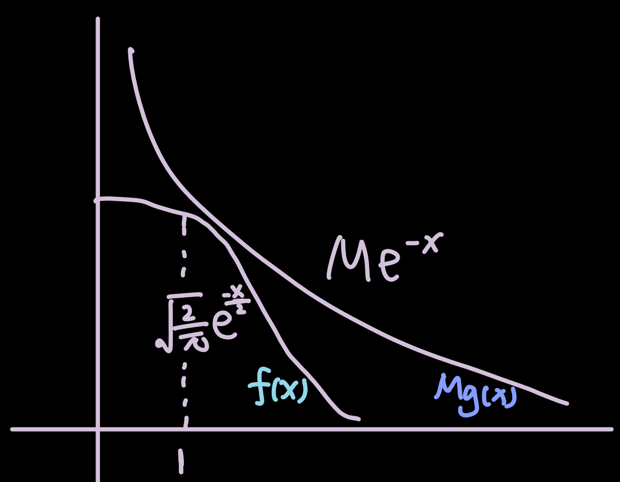
$$= \sqrt{\frac{2}{\pi}}\, e^{-\frac{1}{2}(x^2-2x)}$$

$$= \sqrt{\frac{2}{\pi}}\, e^{-\frac{1}{2}((x-1)^2-1)}$$

When $x=1$, $\frac{f(x)}{g(x)}$ is maximized

so $\max\limits_{x} \frac{f(x)}{g(x)} = \sqrt{\frac{2}{\pi}}\, e^{\frac{1}{2}} = M$

The 3rd method to generate normal random variable — rejection sampling



Application : Generative Modeling

1 million face images

$\Downarrow$ learning

$\vec{x} = g(\vec{z})$ → neural network

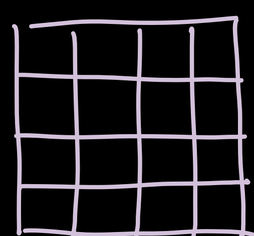
based on transforming normal random varibles to face images

$$\vec{z} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_k \\ \vdots \\ z_d \end{pmatrix}, \quad z_k \sim N(0,1) \text{ , indep.}$$

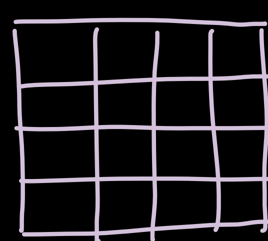
$d = 100$

Denoising diffusion probability model

each training image $\vec{x}$



noise image $\vec{\varepsilon}$



$\downarrow$

each element $\sim N(0,1)$

$$\vec{x} + \vec{\varepsilon} \xrightarrow[\substack{\text{denoising} \\ \text{at different } \sigma}]{\text{mappind}} \vec{x}$$

Generative process:

start from $\sigma N(0, 1)$ noise with a big σ, repeatedly do denoising, to gradually reduce σ.