Introdução à Física Computacional II (4300318)

Prof. André Vieira
apvieira@if.usp.br
Sala 3120 – Edifício Principal

Aula 9

Números aleatórios não uniformes
Números aleatórios gaussianos

ATENÇÃO!

Reforçando o pedido da Comissão de Graduação do IFUSP, peço que preencham um formulário que busca avaliar o andamento dos cursos online durante a pandemia. O formulário está disponível **neste link**.

Distribuições uniformes

- A função `random()` do pacote `random` do Python produz números pseudoaleatórios no intervalo $[0,1)$. Esses números correspondem, em tese, a uma distribuição de probabilidades uniforme:

$$p(x)=\begin{cases} 0 \text{ se } x<0 \text{ ou } x\geq 1, \\ 1 \text{ se } 0\leq x<1. \end{cases}$$

$$P(x_1\leq x\leq x_2)=\int_{x_1}^{x_2} p(x)\,dx=x_2-x_1, \quad \text{se} \quad x_1, x_2\in[0,1)$$

Distribuições não uniformes

- E se desejarmos produzir números aleatórios que sigam outras distribuições?
- Embora o pacote `random` possua funções que servem a esse propósito para algumas distribuições específicas (mais sobre isso adiante), vamos discutir um método de aplicabilidade mais geral, o **método da transformação**.

Preâmbulo: outras distribuições uniformes

- Uma distribuição de probabilidades uniforme no intervalo $[a,b)$ é definida por

$$p(x)=\begin{cases} 0 \quad \text{se } x<a \text{ ou } x\geq b, \\ 1/(b-a) \quad \text{se } a\leq x<b. \end{cases}$$

- Mostra-se que a média e a variância de x são

$$\mu\equiv\bar{x}=\int_{-\infty}^{+\infty} x\,p(x)\,dx=\frac{a+b}{2},$$

$$\sigma^2\equiv\overline{(x-\mu)^2}=\int_{-\infty}^{+\infty}(x-\mu)^2\,p(x)\,dx=\frac{(b-a)^2}{12}.$$

Preâmbulo: outras distribuições uniformes

- Se quisermos escolher um número aleatório x uniformemente no intervalo $[a,b)$, é intuitivo que basta escolher um número aleatório z no intervalo $[0,1)$ e transformá-lo linearmente:

$$x(z)=a+(b-a)z\,.$$

- Note que isso equivale a resolver a equação

$$\int_{-\infty}^{x(z)} p(x')dx'=\int_{a}^{x(z)} \frac{dx'}{b-a}=\int_{0}^{z} dz'=z$$

com $0\leq z<1$.

O método da transformação

- Suponha que $x = x(z)$ seja uma função de uma variável aleatória z, associada a uma distribuição de probabilidades $q(z)$. A probabilidade de sortear a partir de $q(z)$ um número entre z e z + dz é

$$q(z)dz.$$

- Como função de uma variável aleatória, x também é uma variável aleatória, e se denotarmos por $p(x)$ a distribuição associada, a probabilidade de obter por um sorteio de z um número $x = x(z)$ entre x e x + dx é

$$p(x)dx = q(z)dz.$$

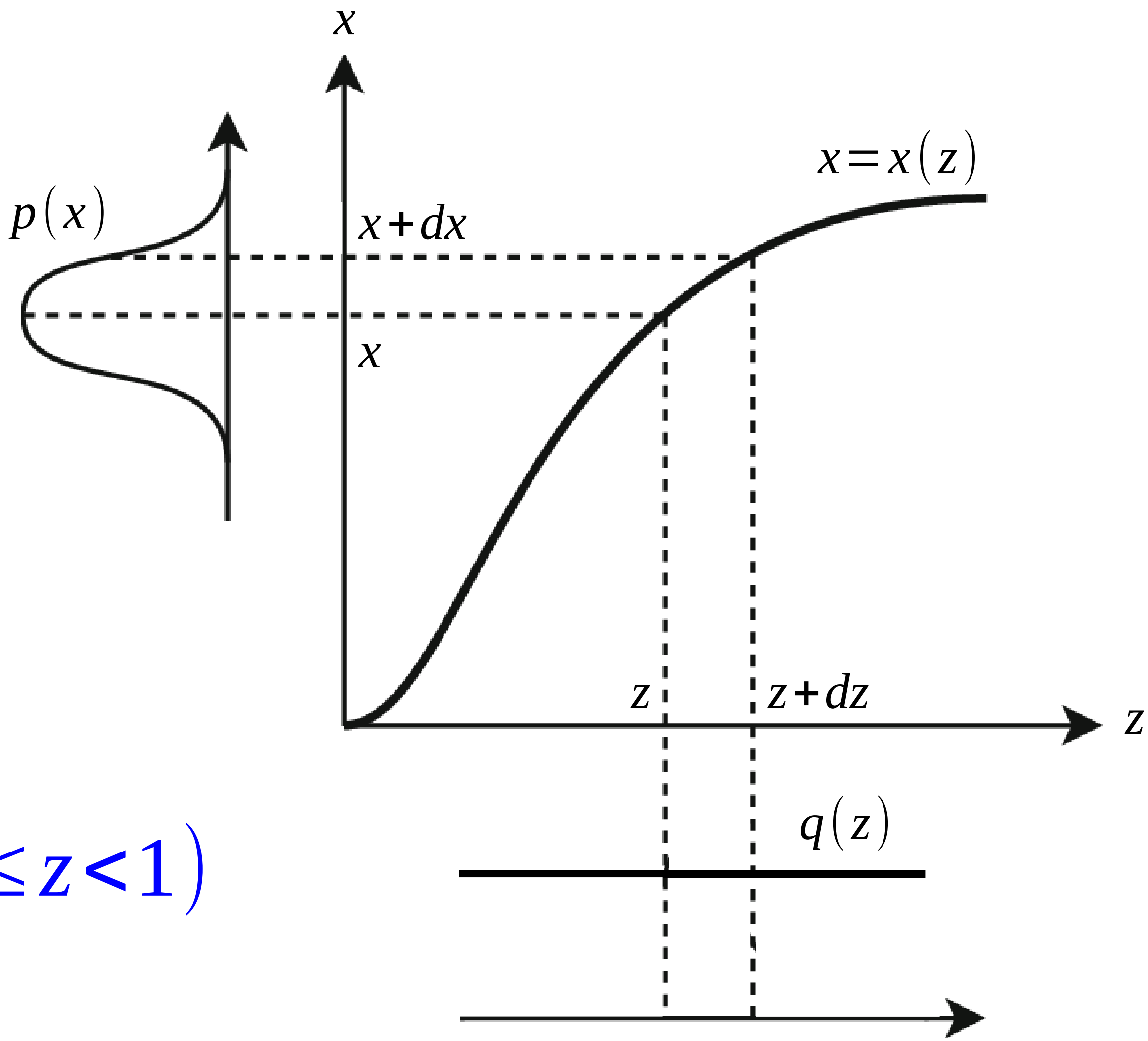
O método da transformação

$$p(x)\,dx = q(z)\,dz$$

Se $q(z)$ é uniforme entre 0 e 1:

$$\int_{-\infty}^{x} p(x')\,dx' = \int_{0}^{z} q(z')\,dz'$$

$$\boxed{\int_{-\infty}^{x} p(x')\,dx' = z} \qquad (0 \le z < 1)$$



Para obter x segundo $p(x)$, sorteia-se z uniforme. A solução da equação destacada fornece a relação $x(z)$. Compare com o slide 5.

Exemplo 1

- Vamos voltar ao caso do decaimento radioativo. Na aula passada, trabalhamos com a probabilidade de que um átomo decaia durante um certo intervalo de tempo. Podemos também analisar o problema de um outro ponto de vista, pensando na distribuição de probabilidades do tempo de decaimento.
- Vamos recordar que chamamos de τ a meia vida do isótopo radioativo.

Exemplo 1

- A probabilidade de que um átomo decaia durante um intervalo de tempo $\Delta t \ll \tau$ é

$$p = 1 - 2^{-\Delta t/\tau} = 1 - \exp\left(-\frac{\Delta t}{\tau}\ln 2\right) \simeq \frac{\Delta t}{\tau}\ln 2 .$$

- No limite em que Δt se torna infinitesimal, a probabilidade de que um átomo decaia entre os instantes t e t + dt é igual ao produto da probabilidade de que o átomo não decaia até o instante t pela probabilidade de que o átomo decaia durante o intervalo dt, ou seja,

$$P(t)\,dt = 2^{-t/\tau}\frac{dt}{\tau}\ln 2 = \exp\left(-\frac{t}{\tau}\ln 2\right)\frac{\ln 2}{\tau}dt .$$

Exemplo 1

- Para simular o processo de decaimento através dessa abordagem, sorteamos para cada átomo o instante de decaimento através da distribuição exponencial

$$P(t)=\begin{cases} 0 \ \text{ se } \ t<0, \\ \mu \exp(-\mu t) \ \text{ se } \ 0 \le t < \infty \end{cases} \qquad \mu=\frac{\ln 2}{\tau}.$$

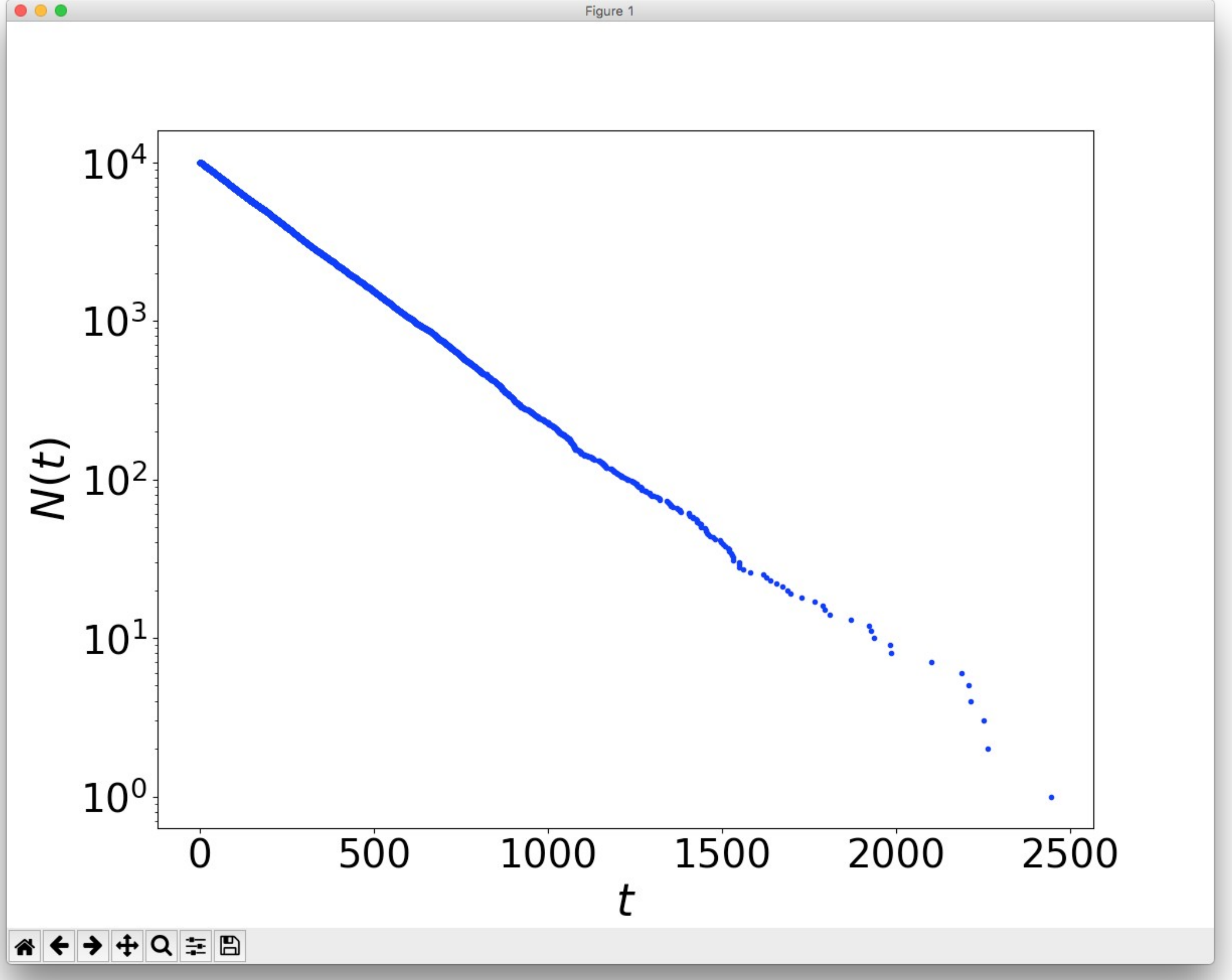
- Para gerar um número aleatório t segundo essa distribuição, sorteamos um número aleatório uniforme z e determinamos t através de

$$\int_{-\infty}^{t} P(t')dt'=z \ \Rightarrow \ 1-\exp(-\mu t)=z \ \Rightarrow \ t=-\frac{1}{\mu}\ln(1-z).$$

Exemplo 1

```
from random import random
from math import log
from numpy import sort,arange
import matplotlib.pyplot as plt

# Constantes
NTl = 10**4          # Número inicial de átomos de tálio
tau = 3.053*60       # Meia-vida do tálio em segundos

mu = log(2)/tau      # Constante da distribuição exponencial

# Sorteando os tempos de decaimento de cada átomo
t_lista = []
for i in range(NTl):
    t_lista.append(-log(1-random())/mu)

# Vamos fazer um gráfico, em função de t, do número de átomos que não
# decaíram até um certo instante t. Para isso, é útil reordenar a lista
# de tempos do menor para o maior.
t_ordenado = sort(t_lista)
N_ordenado = arange(NTl,0,-1)

# Parâmetros da exibição dos gráficos
plt.rcParams['xtick.labelsize'] = 28
plt.rcParams['ytick.labelsize'] = 28
plt.rcParams['axes.labelsize'] = 32

# Traçando o gráfico
plt.figure(figsize=(12,9))
plt.semilogy(t_ordenado,N_ordenado,"b.")
plt.show()
```

Exemplo 1



Como na aula passada, obtemos um número de átomos que decai exponencialmente com o tempo. Mas sorteamos muito menos números aleatórios do que com o método anterior.

Números aleatórios gaussianos

- O método da transformação, como formulamos, requer que saibamos inverter analiticamente a integral da distribuição $p(x)$. Esse não é o caso imediato da distribuição gaussiana,

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

uma vez que sua integral é

$$\int_{-\infty}^{x} p(x')\,dx' = \frac{1}{2} + \frac{1}{2}\operatorname{erf}\left(\frac{x-\mu}{\sqrt{2\sigma^2}}\right)$$

e não há expressão simples para a inversa da função erro em termos de funções elementares.

Números aleatórios gaussianos

Vamos discutir a seguir quatro métodos para gerar números aleatórios gaussianos, e comparar sua eficiência.

Números aleatórios gaussianos

- O primeiro método é baseado no **teorema central do limite**. Segundo o teorema, sendo $\{z_i\}$ um conjunto de números independentes extraídos de uma distribuição de probabilidades com média μ e variância σ^2, a quantidade X definida por

$$X = \mu + \lim_{n \to \infty} \frac{1}{\sqrt{n}} \sum_{i=1}^{n} (z_i - \mu)$$

  é descrita por uma distribuição de probabilidades gaussiana com média μ e variância σ^2.
- Fazendo n suficientemente grande, podemos gerar números aleatórios gaussianos X a partir de uma distribuição uniforme.

Exemplo 2

$$\mu = 2$$

$$\sigma^2 = 3$$

$$n = 1000$$

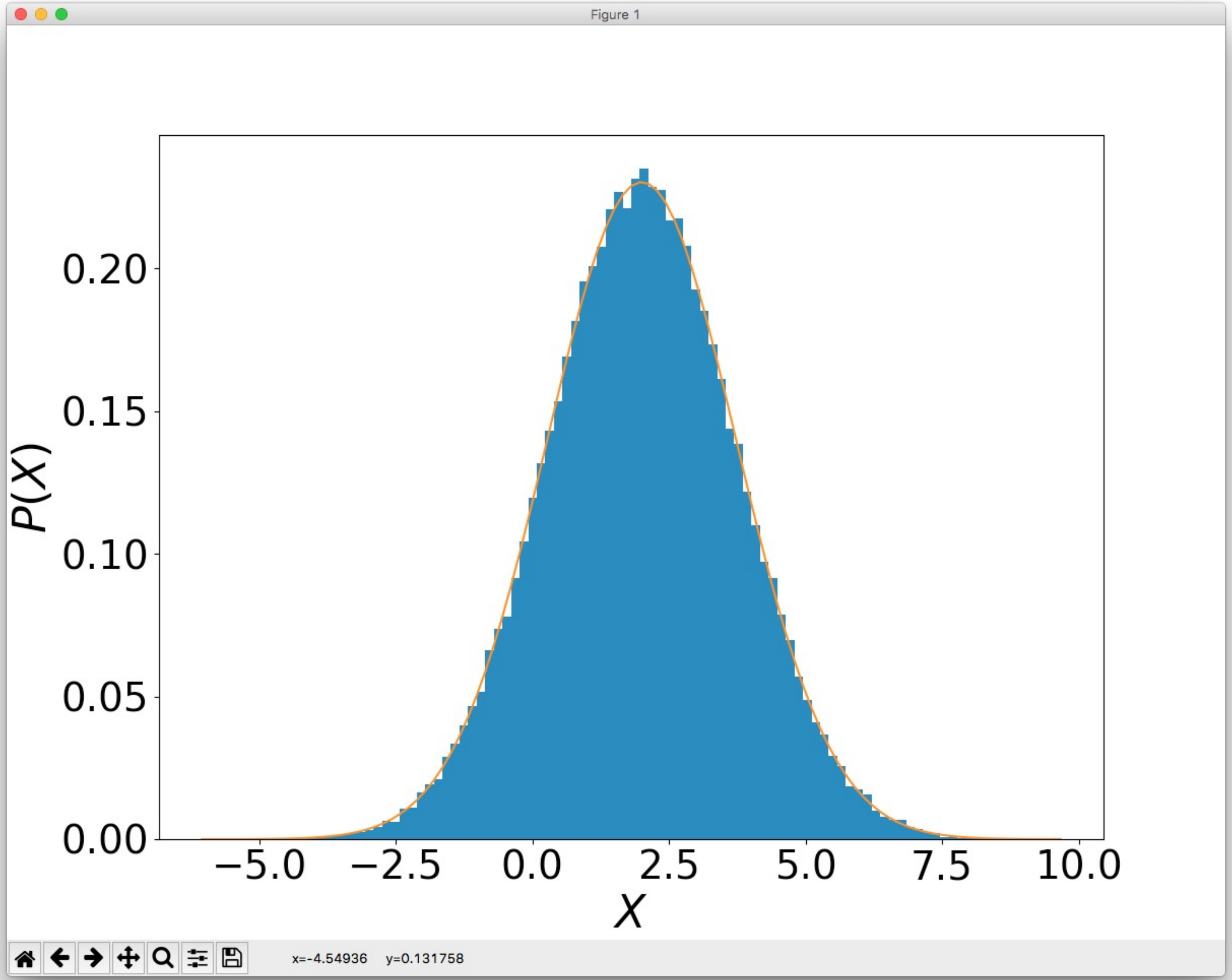
$$N = 10^5$$

```
# Constantes
mu = 2.0        # Valor médio desejado das variáveis aleatórias gaussianas
sigma2 = 3.0     # Valor desejado da variância da gaussiana
n = 10**3       # Número de variáveis z a sortear para definir cada X
N = 10**5       # Número de variáveis X a obter
Bins = 10**2    # Número de caixas do histograma das variáveis X

# Parâmetros da distribuição uniforme correspondente
a, b = mu - sqrt(3*sigma2), mu + sqrt(3*sigma2)

# Vamos produzir as N variáveis X, cada uma envolvendo a soma de n
# variáveis z distribuídas uniformemente entre a e b correspondendo
# à média mu e à variância sigma2. Depois vamos produzir um gráfico em
# histograma, para visualizar a distribuição dos X obtidos.
X_lista = []
for i in range(N):
    S = 0.0
    for j in range(n):
        S += (b-a)*random() + a
    X_lista.append(mu + sqrt(n)*(S/n-mu))

# Vamos também produzir o gráfico de uma distribuição gaussiana com
# média mu e variância sigma2, para comparação.
XX_min, XX_max = min(X_lista), max(X_lista)
h = (XX_max - XX_min)/Bins
XX_lista, Gauss = arange(XX_min,XX_max,h), []
Gauss[:] = [1/sqrt(2*pi*sigma2)*exp(-(X-mu)**2/2/sigma2) for X in XX_lista]

# Parâmetros da exibição dos gráficos
plt.rcParams['xtick.labelsize'] = 28
plt.rcParams['ytick.labelsize'] = 28
plt.rcParams['axes.labelsize'] = 32

# Traçando os gráficos
plt.figure(figsize=(12,9))
plt.hist(X_lista,bins=Bins,density=True) # Produzindo um histograma normalizado
plt.plot(XX_lista,Gauss)
plt.xlabel("$X$")
plt.ylabel("$P(X)$")
plt.show()
```

Exemplo 2

$$\mu = 2$$
$$\sigma^2 = 3$$
$$n = 1000$$

$$N = 10^5$$



O histograma em **azul** é o resultado dos números aleatórios gaussianos produzidos, enquanto a curva **laranja** é uma distribuição gaussiana com mesmos μ e σ. O tempo de execução do programa foi de 37 segundos.

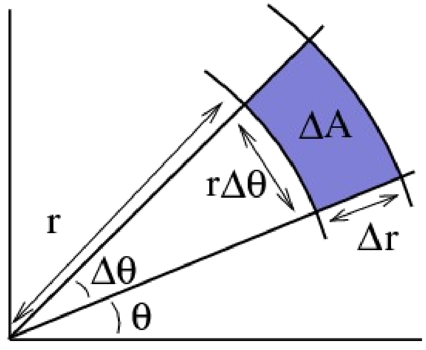
Números aleatórios gaussianos

- O segundo método é uma extensão do método da transformação. Se $p(x)$ é uma distribuição de probabilidades gaussiana com média zero, vale

$$p(x)\,dx \times p(y)\,dy = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)dx\,dy\,.$$

- Podemos interpretar x e y como coordenadas de um ponto em 2D. Descrevendo o mesmo ponto em coordenadas polares r e θ, temos

$$\begin{cases} x = r\cos\theta \\ y = r\,\mathrm{sen}\,\theta \end{cases} \Rightarrow \begin{cases} x^2+y^2 = r^2 \\ dx\,dy = r\,dr\,d\theta \end{cases}$$



Números aleatórios gaussianos

- Podemos portanto escrever

$$p(x)\,dx \times p(y)\,dy = p_r(r)\,dr \times p_\theta(\theta)\,d\theta,$$

em que

$$p_r(r) = \frac{r}{\sigma^2}\exp\left(-\frac{r^2}{2\sigma^2}\right), \qquad p_\theta(\theta) = \frac{1}{2\pi}.$$

- Logo, escolhendo dois números independentes r (entre 0 e ∞) e θ (entre 0 e 2π) segundo as distribuições acima e convertendo de volta para coordenadas cartesianas, obtemos dois números independentes x e y que seguem distribuições gaussianas.

Números aleatórios gaussianos

- Note que a distribuição para θ é uniforme entre 0 e 2π, enquanto para gerar o número r a partir de um número z escolhido uniformemente entre 0 e 1 temos que resolver a equação

$$\int_0^r p_r(r')\,dr' = z \;\Rightarrow\; \int_0^{r/\sigma} u\,e^{-u^2/2}\,du = \int_0^{r^2/2\sigma^2} e^{-w}\,dw = z$$

$$1 - e^{-r^2/2\sigma^2} = z \;\Rightarrow\; r = \sqrt{-2\sigma^2 \ln(1-z)}.$$

- O próximo exemplo utiliza esse método para gerar números aleatórios gaussianos.

Exemplo 3

$$\mu = 2$$

$$\sigma^2 = 3$$

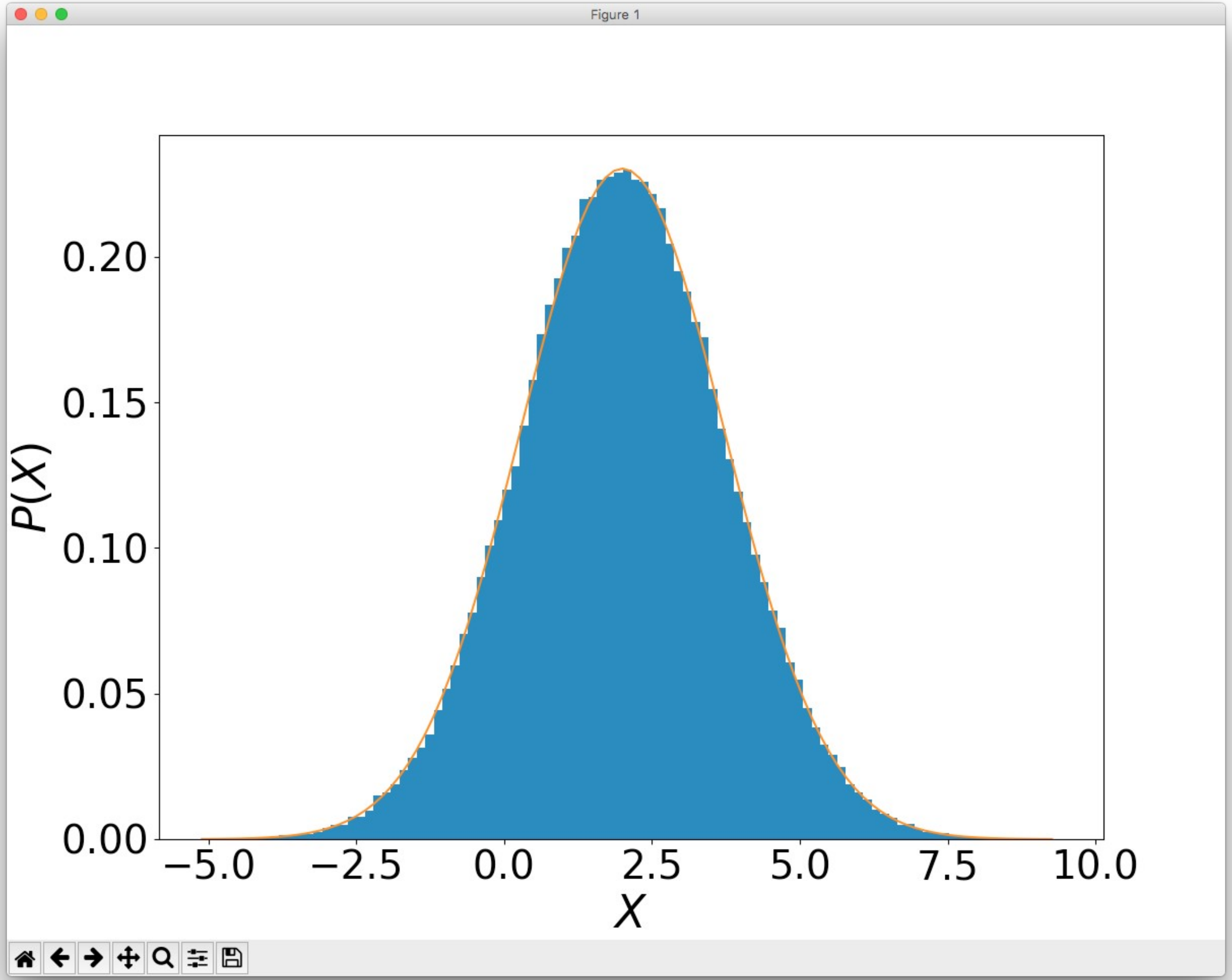
$$N = 10^5$$

```
# Constantes
mu = 2.0      # Valor médio desejado das variáveis aleatórias gaussianas
sigma2 = 3.0  # Valor desejado da variância da gaussiana
N = 10**5     # Número de variáveis aleatórias gaussianas a obter
Bins = 10**2  # Número de caixas do histograma das variáveis gaussianas

# O laço abaixo produz N números aleatórios gaussianos pelo método da
# transformação estendido. Note que produzimos um par de números a cada
# passo do laço. Perceba também o uso do método 'extend' para acrescentar
# mais de um elemento à lista que contém os números aleatórios gaussianos.
X_lista = []
for i in range(N//2):
    r = sqrt(-2*sigma2*log(1-random()))
    theta = 2*pi*random()
    X_lista.extend([mu+r*cos(theta),mu+r*sin(theta)])
```

Exemplo 3

$$\mu = 2$$
$$\sigma^2 = 3$$

$$N = 10^5$$



O histograma em **azul** é o resultado dos números aleatórios gaussianos produzidos, enquanto a curva **laranja** é uma distribuição gaussiana com mesmos μ e σ. O tempo de execução do programa foi de 0.1 segundo.

Números aleatórios gaussianos

- O terceiro método faz uso da função erro inversa implementada pelo pacote `scipy`. De posse dessa função, e a partir de um número aleatório uniforme z, resolvemos

$$\int_{-\infty}^{x} p(x')\,dx' = \frac{1}{2} + \frac{1}{2}\operatorname{erf}\left(\frac{x-\mu}{\sqrt{2\sigma^2}}\right) = z$$

$$\frac{x-\mu}{\sqrt{2\sigma^2}} = \operatorname{erfinv}(2z-1) \;\Rightarrow\; x = \mu + \sqrt{2\sigma^2}\operatorname{erfinv}(2z-1)$$

- O próximo exemplo utiliza esse método para gerar números aleatórios gaussianos.

Exemplo 4

$$\mu = 2$$

$$\sigma^2 = 3$$

$$N = 10^5$$

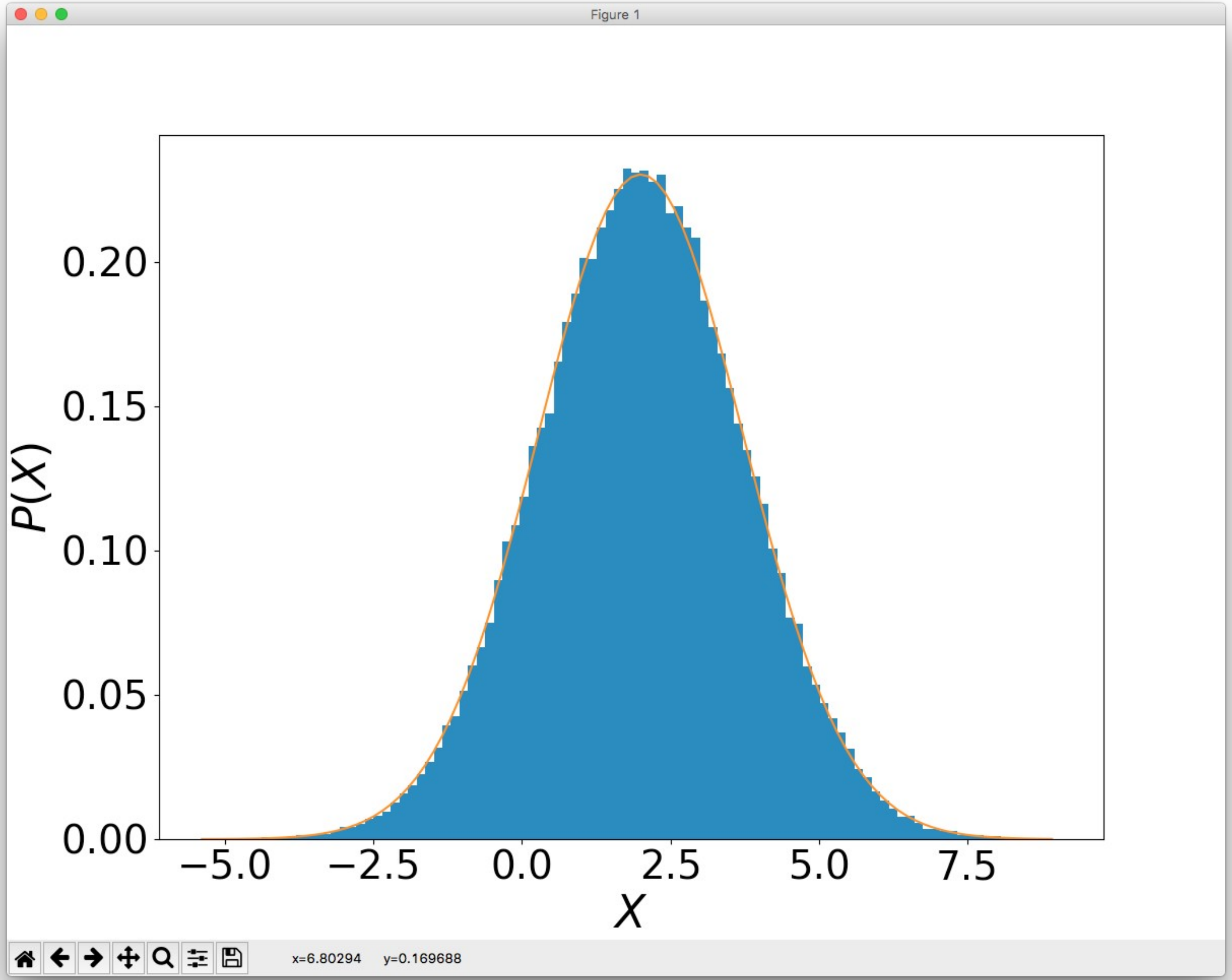
```
from random import random
from math import sqrt,pi,exp
from numpy import arange
from scipy.special import erfinv
import matplotlib.pyplot as plt

# Constantes
mu = 2.0        # Valor médio desejado das variáveis aleatórias gaussianas
sigma2 = 3.0    # Valor desejado da variância da gaussiana
N = 10**5       # Número de variáveis aleatórias gaussianas a obter
Bins = 10**2    # Número de caixas do histograma das variáveis gaussianas

# O laço abaixo produz N números aleatórios gaussianos utilizando a
# função erro inversa do pacote 'scipy'.
X_lista = []
for i in range(N):
    X_lista.append(mu + sqrt(2*sigma2)*erfinv(2*random()-1))
```

Exemplo 4

$\mu = 2$

$\sigma^2 = 3$

$N = 10^5$



O histograma em **azul** é o resultado dos números aleatórios gaussianos produzidos, enquanto a curva **laranja** é uma distribuição gaussiana com mesmos μ e σ. O tempo de execução do programa foi de 0.7 segundo.

Números aleatórios gaussianos

- Finalmente, o quarto método faz uso de funções implementadas pelo próprio pacote `random`, cujos argumentos são a média e a variância da distribuição desejada.
- A função `gauss` (‘exemplo5a.py’, na pasta de arquivos da aula no moodle) implementa a extensão do método da transformação que implementamos, e o tempo de execução para os parâmetros que utilizamos nos exemplos anteriores é de 0.13 segundos.

Números aleatórios gaussianos

- Finalmente, o quarto método faz uso de funções implementadas pelo próprio pacote `random`, cujos argumentos são a média e a variância da distribuição desejada.
- Já a função `normalvariate` (‘exemplo5b.py’, na pasta de arquivos da aula no moodle) implementa um algoritmo distinto, devido a Kinderman e Monahan, e o tempo de execução para os parâmetros que utilizamos nos exemplos anteriores é de 0.16 segundos.
- Nossa implementação “caseira” é mais rápida que as implementações do pacote `random`!

Exemplo 6

- Vamos voltar ao problema do decaimento radioativo em um outro contexto. O méson pi, ou píon, é uma partícula instável, cuja meia vida, em seu referencial de repouso, é de 2.6×10^{-8} s, e cuja massa é de 140 MeV/c^2. Vamos imaginar um feixe de píons com energias cinéticas descritas por uma distribuição gaussiana com média de 200 MeV e desvio-padrão de 30 MeV. Qual é a fração de píons que sobreviverá após um deslocamento de 20 m?

Exemplo 6

- A primeira estimativa que podemos fazer é que sobreviverá uma fração de menos de 1/3 dos píons. Isso porque mesmo um fóton percorre apenas $\left(2.6\times10^{-8}\right)\times\left(3.0\times10^{8}\right)=7.8$ m durante a meia-vida de um píon.
- O que há de errado com essa estimativa?

Exemplo 6

- Para um píon que se desloca com altíssimas velocidades, o tempo passa mais devagar. Se v é a velocidade do píon no referencial do laboratório, um processo que no laboratório leva um tempo Δt corresponde, para o píon, a um tempo

$$\Delta t_0 = \frac{\Delta t}{\gamma} = \sqrt{1 - (v/c)^2}\,\Delta t.$$

- Se a velocidade do píon corresponder a uma fração significativa da velocidade da luz, o efeito de dilatação temporal será substancial.

Exemplo 6

- No referencial do laboratório, o tempo para que um píon com velocidade v percorra uma distância L é

$$\Delta t = L/v\,,$$

de modo que

$$\Delta t_0 = \sqrt{1 - \left(\frac{v}{c}\right)^2}\,\Delta t = \sqrt{1 - \left(\frac{v}{c}\right)^2}\,\frac{L}{v}\,.$$

- Por outro lado, a energia cinética do píon é a diferença entre a energia do píon à velocidade v e a energia de repouso:

$$K = \gamma mc^2 - mc^2\,.$$

Exemplo 6

- O fator relativístico γ depende de v, e em termos da energia cinética obtemos

$$\frac{v}{c}=\sqrt{1-\frac{\left(mc^2\right)^2}{\left(mc^2+K\right)^2}}.$$

  É possível mostrar que essa relação se reduz à relação clássica $v=\left(2K/m\right)^{1/2}$ quando $v\ll c$.

- Em suma: conhecendo a energia cinética de cada píon, determinamos sua velocidade e, em seguida, o tempo próprio de percurso. Com a razão entre esse tempo e o tempo próprio de decaimento determinamos a probabilidade de que o píon sobreviva ao percurso.

Exemplo 6

```
# Constantes
N = 10**6                   # Número inicial de píons
L = 20.0                    # Distância que os píons percorrem
tau = 2.6e-8                # Meia-vida do píon no repouso, em segundos
c = 3e8                     # Velocidade da luz, em metros por segundo
mc2 = 140                   # Energia de repouso de um píon, em MeV
K_medio = 200               # Energia cinética média dos píons, em MeV
dK = 30                     # Desvio-padrão da energia cinética dos píons

# Função que gera pares de números aleatórios gaussianos
mu = K_medio
sigma = dK
doispi = 2.0*pi
doissigma2 = 2.0*sigma*sigma
def gaussian():
    r = sqrt(-doissigma2*log(1-random()))
    theta = doispi*random()
    return mu+r*cos(theta), mu+r*sin(theta)

# Vamos sortear as energias de cada píon e delas calcular a velocidade
# (em unidades de c).
v_lista = zeros(N)
for n in range(N//2):
    K0, K1 = gaussian()
    v_lista[2*n] = sqrt(1-1/(1+K0/mc2)**2)*c
    v_lista[2*n+1] = sqrt(1-1/(1+K1/mc2)**2)*c

# Para cada píon, determinamos o tempo próprio de percurso delta_t
# e a probabilidade p de que o píon sobreviva ao percurso.
sobreviventes = 0
for n in range(N):
    delta_t = sqrt(1-(v_lista[n]/c)**2)*L/v_lista[n]
    p = 2**(-delta_t/tau)
    if random() < p:
        sobreviventes +=1

print("Fração de píons sobreviventes =",sobreviventes/N)
```

O resultado é uma fração de sobreviventes de cerca de 44%. Como esse resultado mudaria com o aumento da energia cinética média?

Exercício no moodle

Há um único exercício, explorando o conteúdo da aula de hoje, e que pode ser feito com base nos programas dos exemplos desta aula e da aula anterior, disponíveis no moodle.

O exercício vale até 2 pontos de bônus para o EP4 e a data para entrega é o dia **20 de maio**.