Advanced Monte Carlo algorithms

Bad Honnef Physics School
Computational Physics of Complex and Disordered Systemss

Werner Krauth

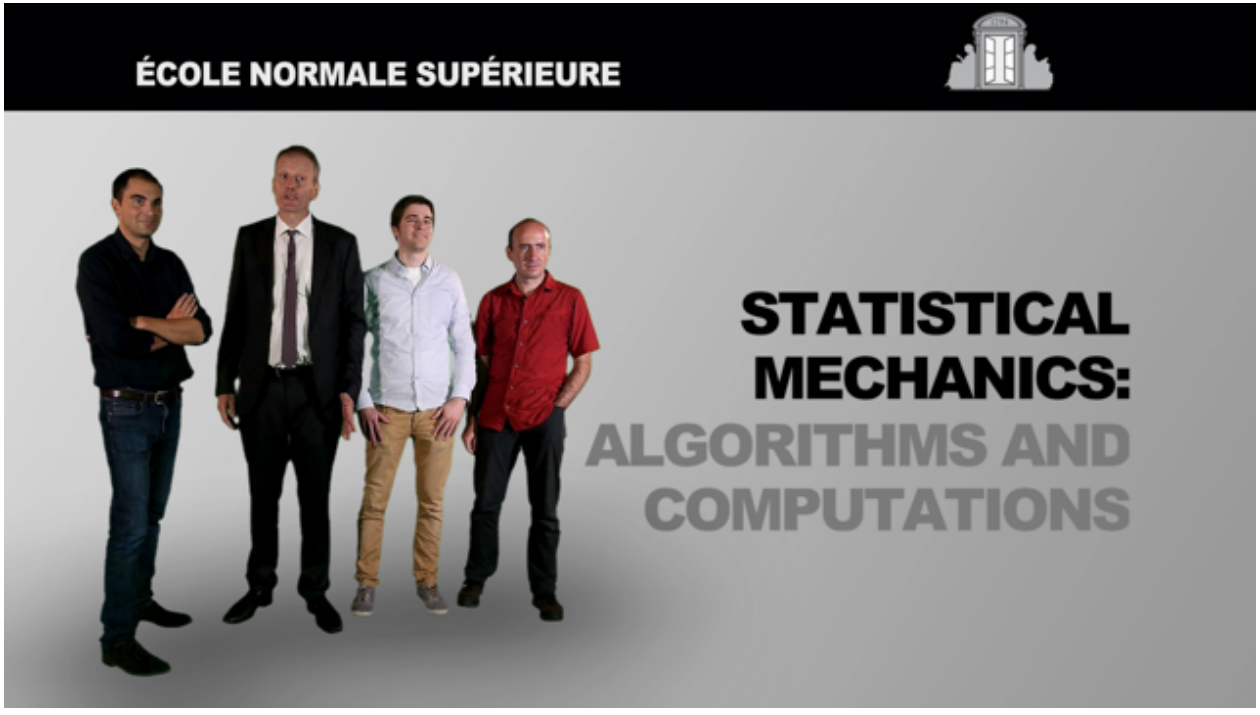
Département de Physique
Ecole Normale Supérieure, Paris, France

23 September 2015

- Monte Carlo algorithms - basic notions.
- Hard disks: From detailed balance to global balance and to cluster algorithms.
- Integration and Sampling: From Gaussians to Maxwell and Boltzmann.
- Cluster algorithms for spin models: Ising, XY, Spin glasses.

Département de Physique
École normale supérieure

MOOC Statistical Mechanics: Algorithms and Computations



- see also: Werner Krauth *Statistical Mechanics: Algorithms and Computations* (Oxford, 2006)

Département de Physique
École normale supérieure

Gaussian integral

$$I = \int \frac{dx}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$$

Département de Physique
École normale supérieure

Gaussian integral

$$I = \int \frac{dx}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$$

$$I^2 = \left(\int \frac{dx}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} \right)^2$$

Département de Physique
École normale supérieure

Integration and sampling (Gaussians)

$$\left[\int_{-\infty}^{\infty} \frac{\mathrm{d}x}{\sqrt{2\pi}} \exp -x^2/2\right]^2 = \int_{-\infty}^{\infty} \frac{\mathrm{d}x}{\sqrt{2\pi}} \mathrm{e}^{-x^2/2} \int_{-\infty}^{\infty} \frac{\mathrm{d}y}{\sqrt{2\pi}} \exp -y^2/2$$

$$\ldots = \int_{-\infty}^{\infty} \frac{\mathrm{d}x\mathrm{d}y}{2\pi} \exp -(x^2+y^2)/2,$$

polar coordinates ($\mathrm{d}x\mathrm{d}y = r\mathrm{d}r\mathrm{d}\phi$),

$$\ldots = \int_0^{2\pi} \frac{\mathrm{d}\phi}{2\pi} \int_0^{\infty} \mathrm{d}r \ \exp -r^2/2$$

substitutions $r^2/2 = \Upsilon$ ($r\mathrm{d}r = \mathrm{d}\Upsilon$) and $\exp -\Upsilon = \Psi$

$$\ldots = \int_0^{2\pi} \frac{\mathrm{d}\phi}{2\pi} \int_0^{\infty} \mathrm{d}\Upsilon \ \exp -\Upsilon$$

$$\ldots = \int_0^{2\pi} \frac{\mathrm{d}\phi}{2\pi} \int_0^{1} \mathrm{d}\Psi \ = 1$$

Département de Physique
École normale supérieure

gauss-test.py

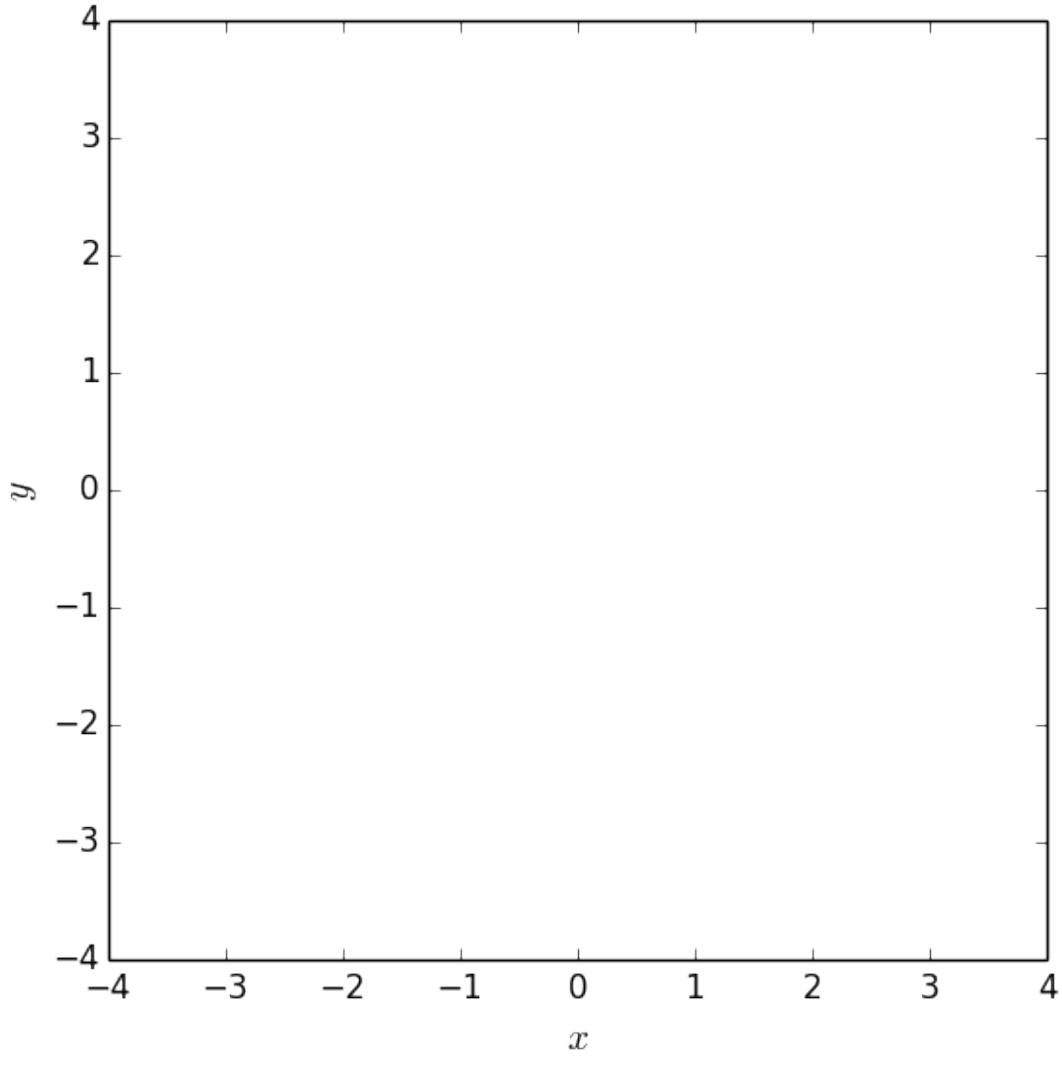
```
import random, math

def gauss_test(sigma):
    phi = random.uniform(0.0, 2.0 * math.pi)
    Upsilon = random.uniform(0.0, 1.0)
    Psi = - math.log(Upsilon)
    r = sigma * math.sqrt(2.0 * Psi)
    x = r * math.cos(phi)
    y = r * math.sin(phi)
    return [x, y]

nsamples = 50
for sample in range(nsamples):
    [x, y] = gauss_test(1.0)
    print x, y
```

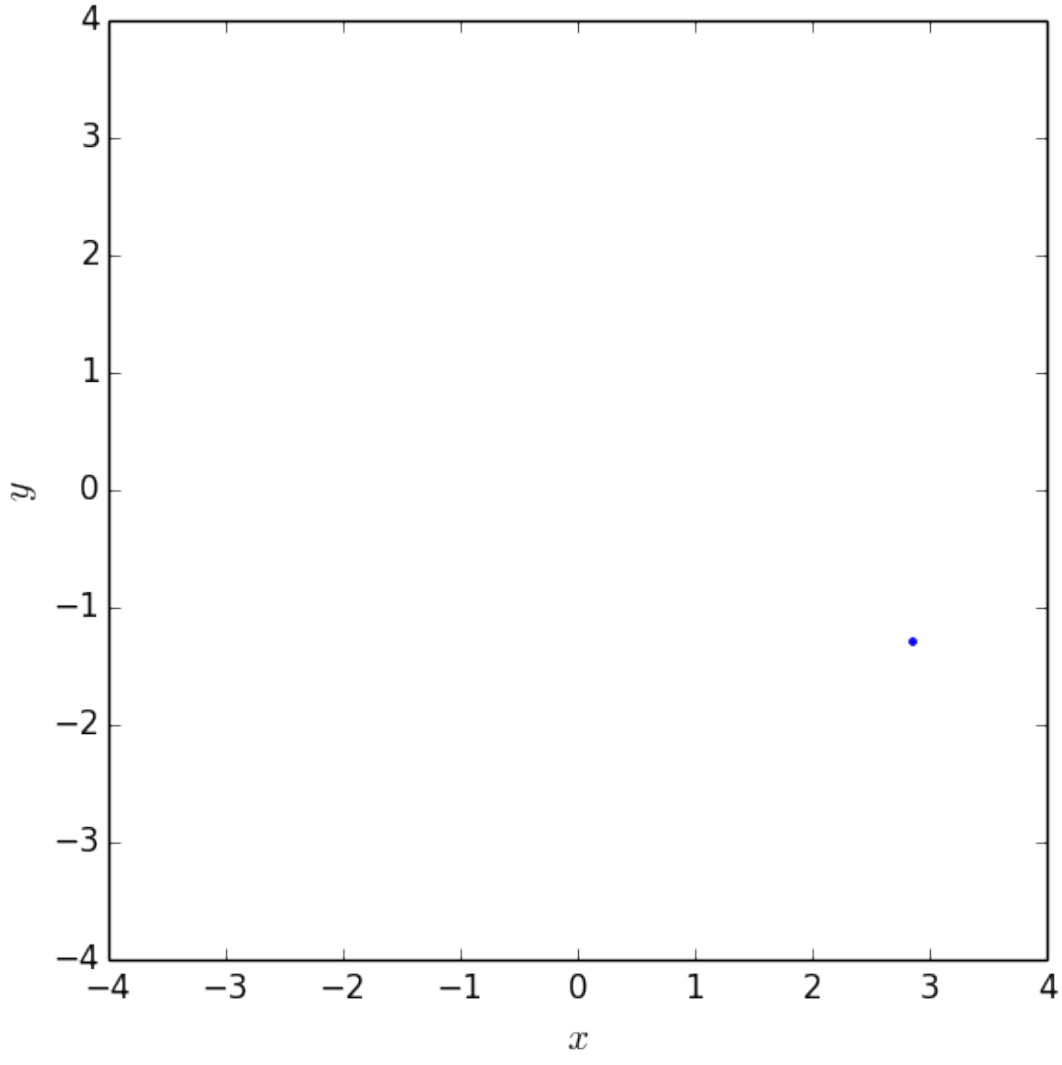
Département de Physique
École normale supérieure

output of gauss-test.py



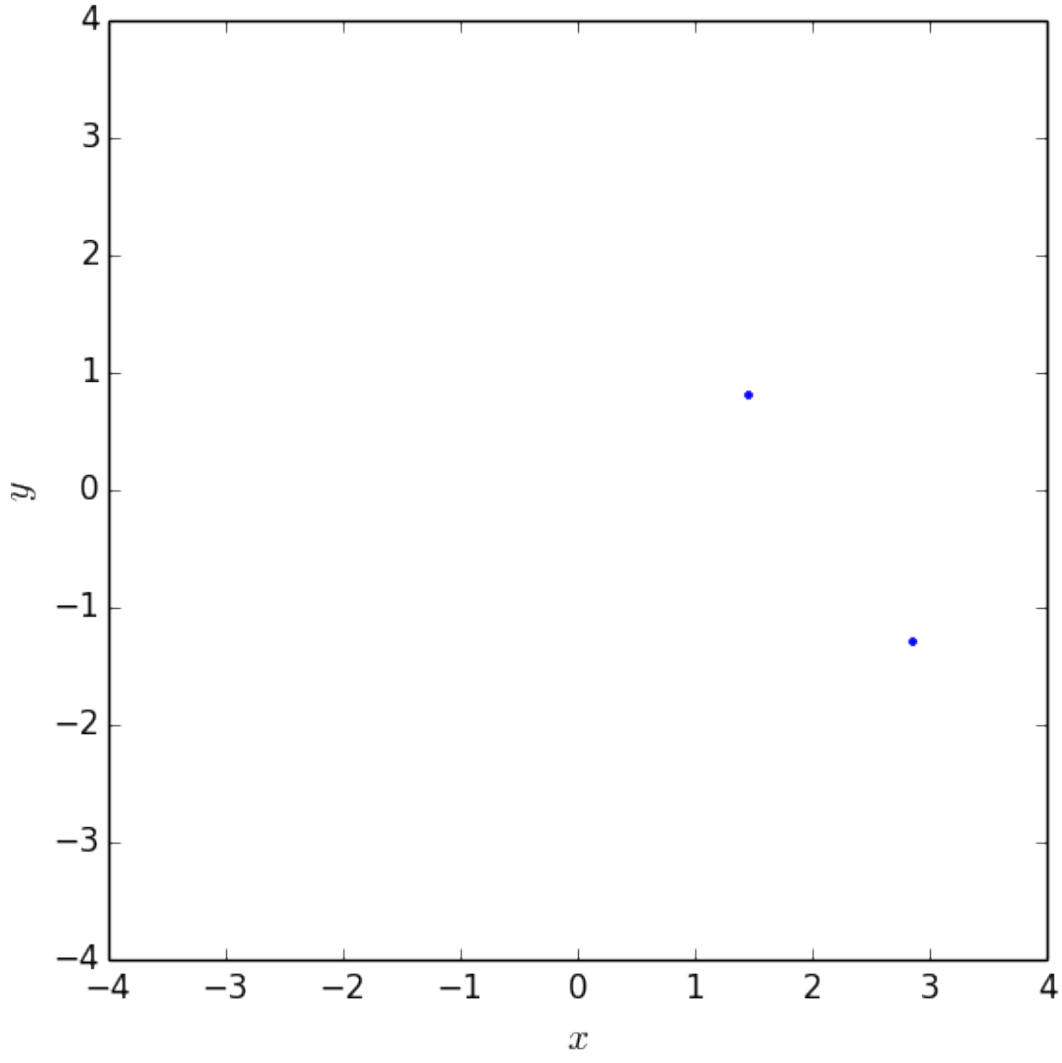
Département de Physique
École normale supérieure

output of gauss-test.py



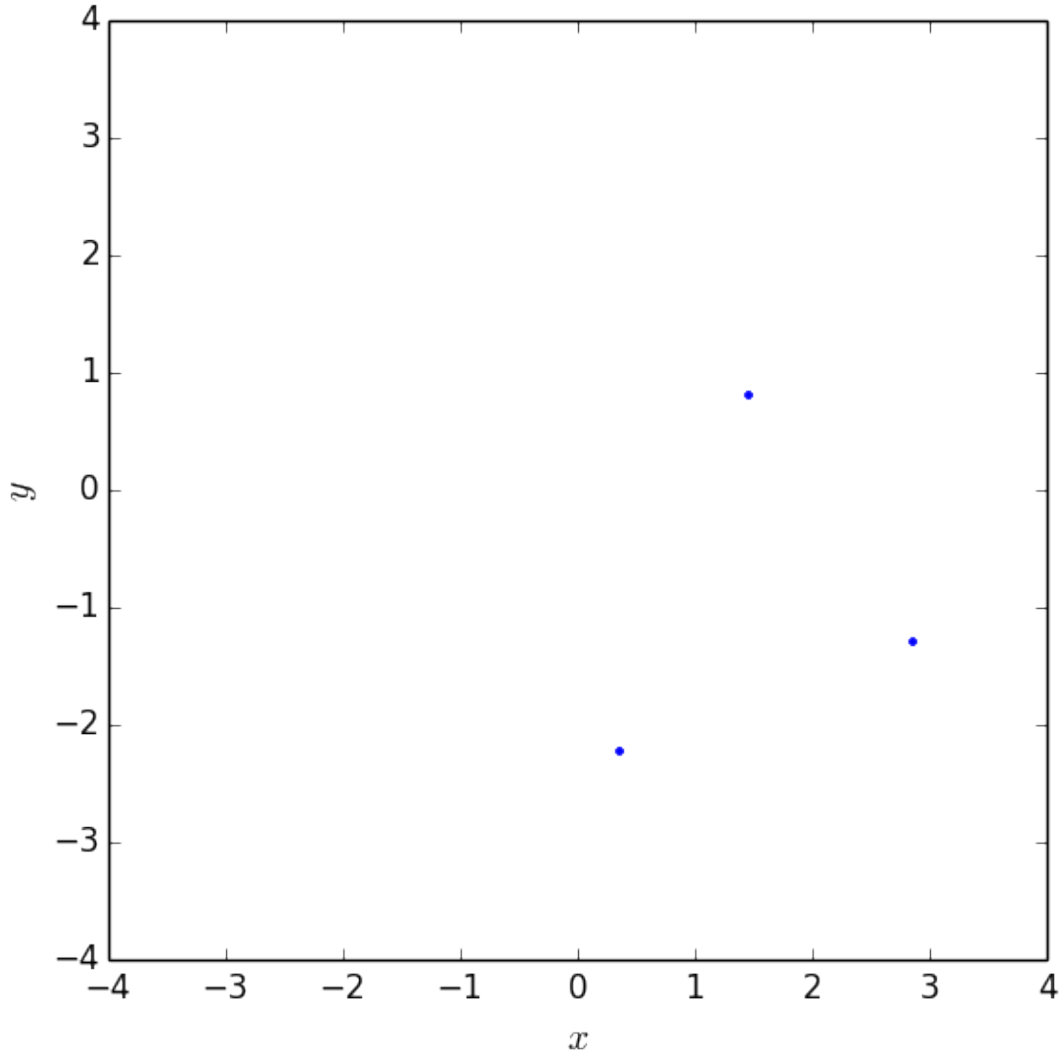
Département de Physique
École normale supérieure

output of gauss-test.py



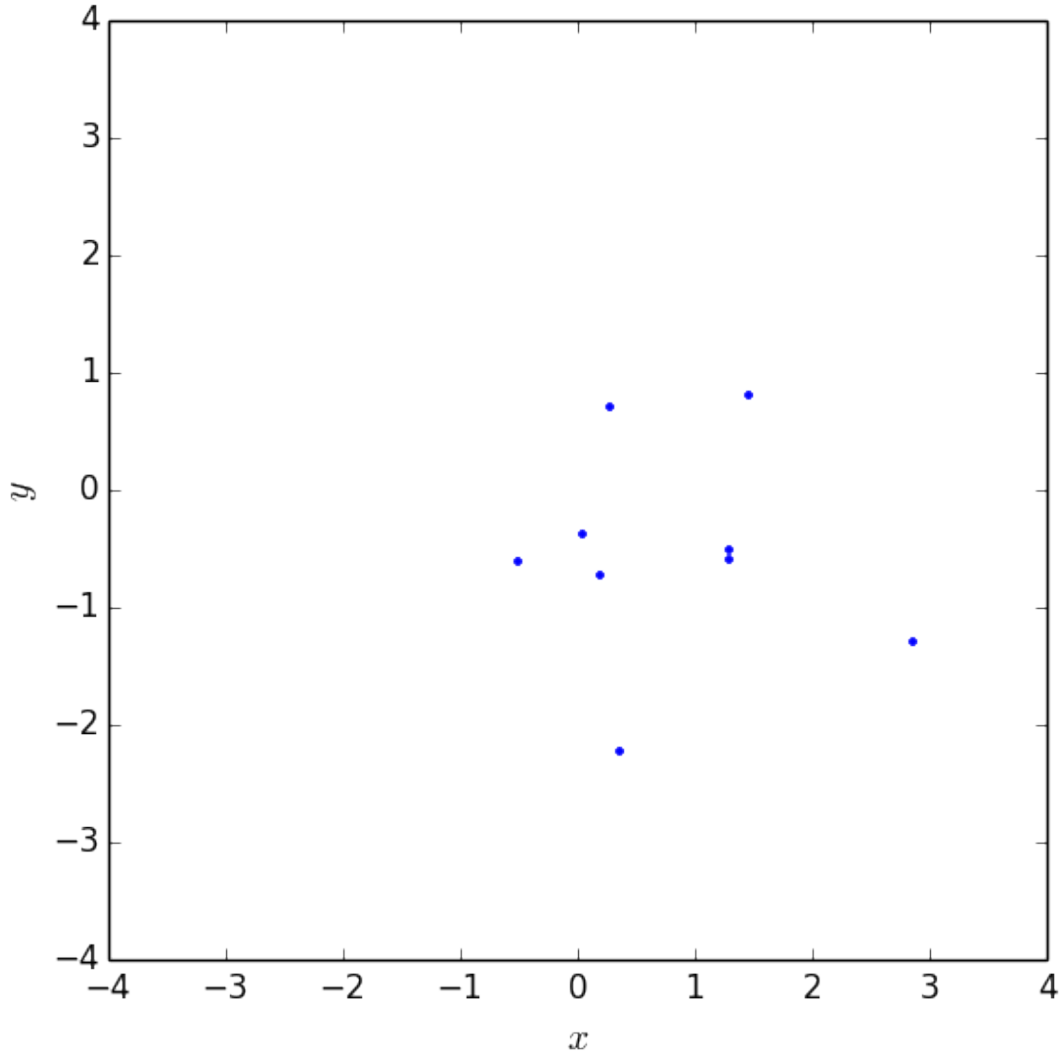
Département de Physique
École normale supérieure

output of gauss-test.py



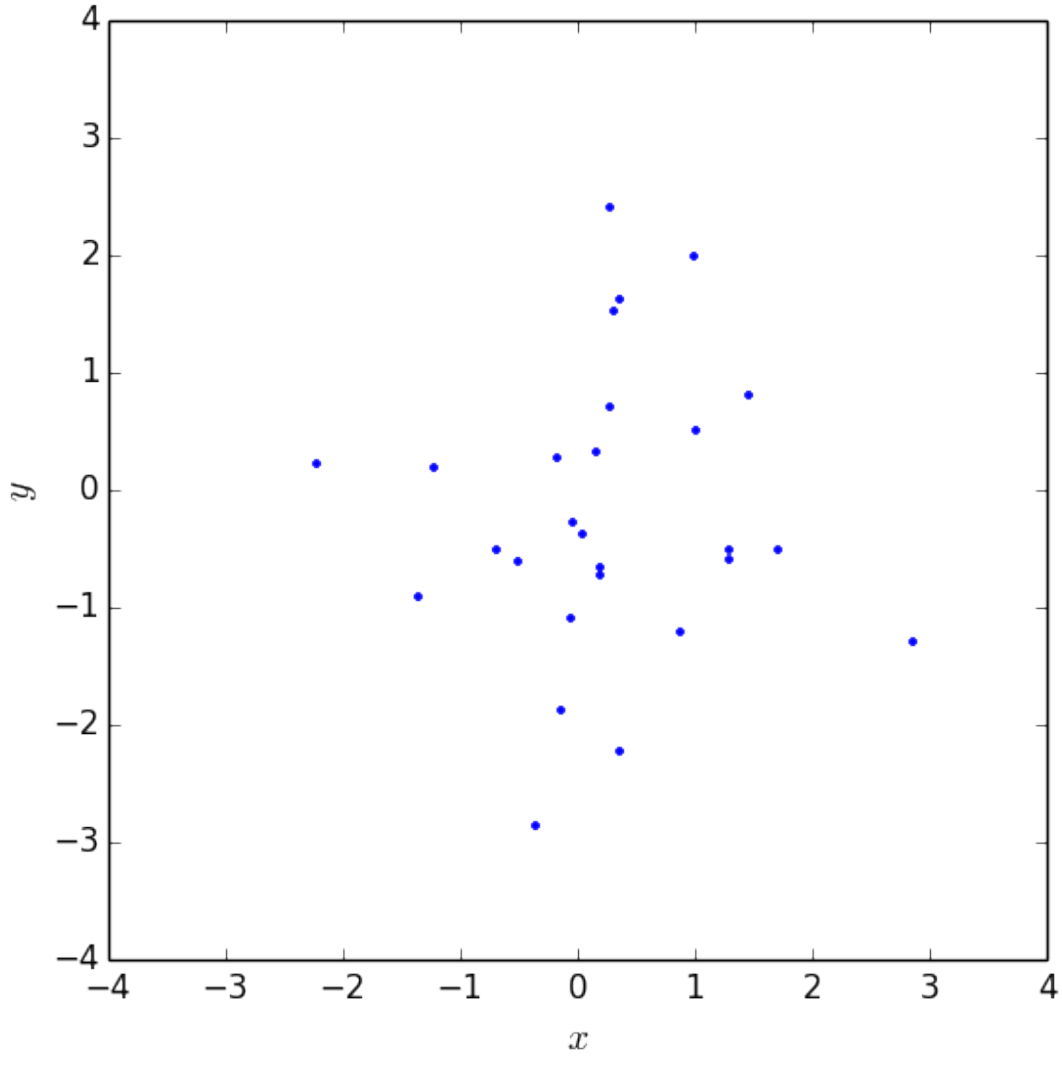
Département de Physique
École normale supérieure

output of gauss-test.py

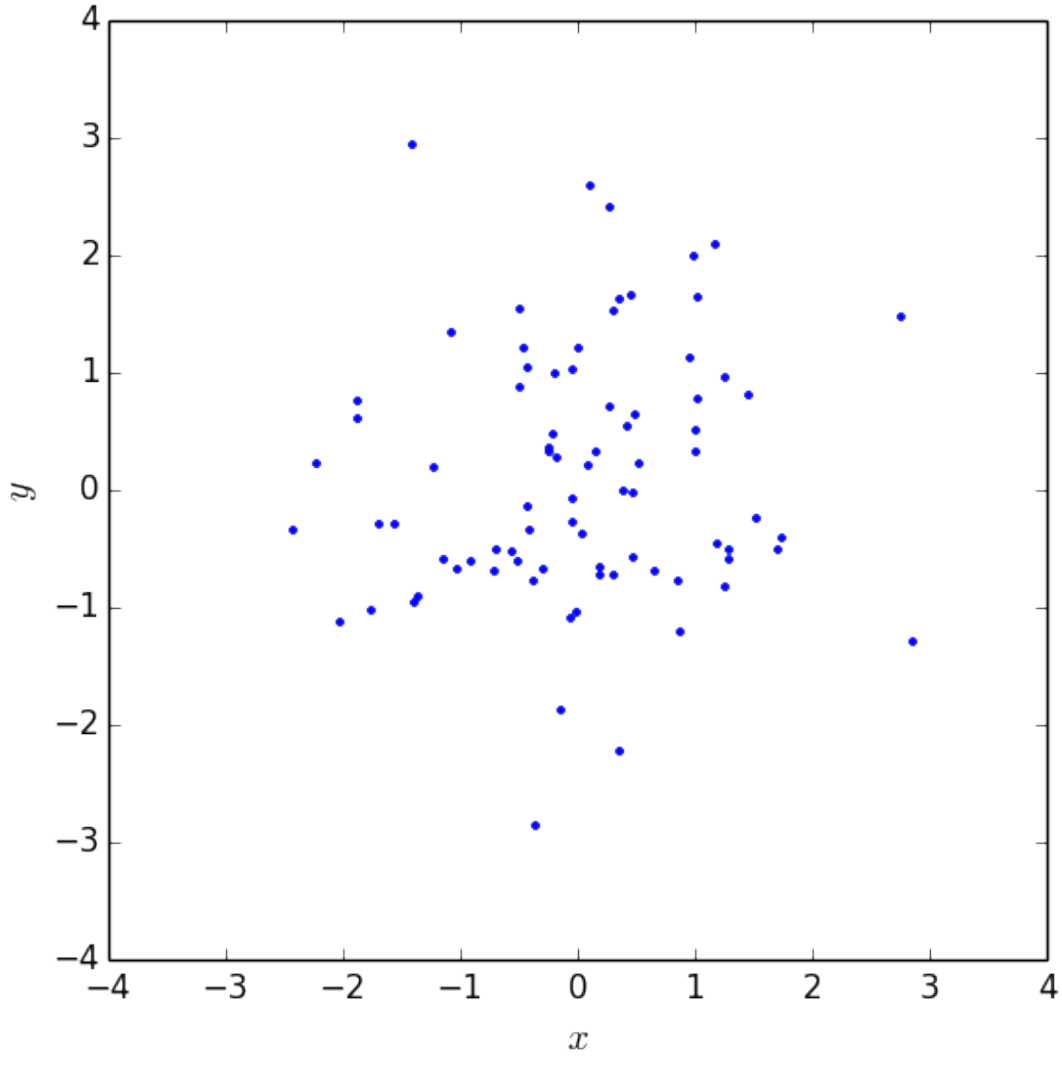


Département de Physique
École normale supérieure

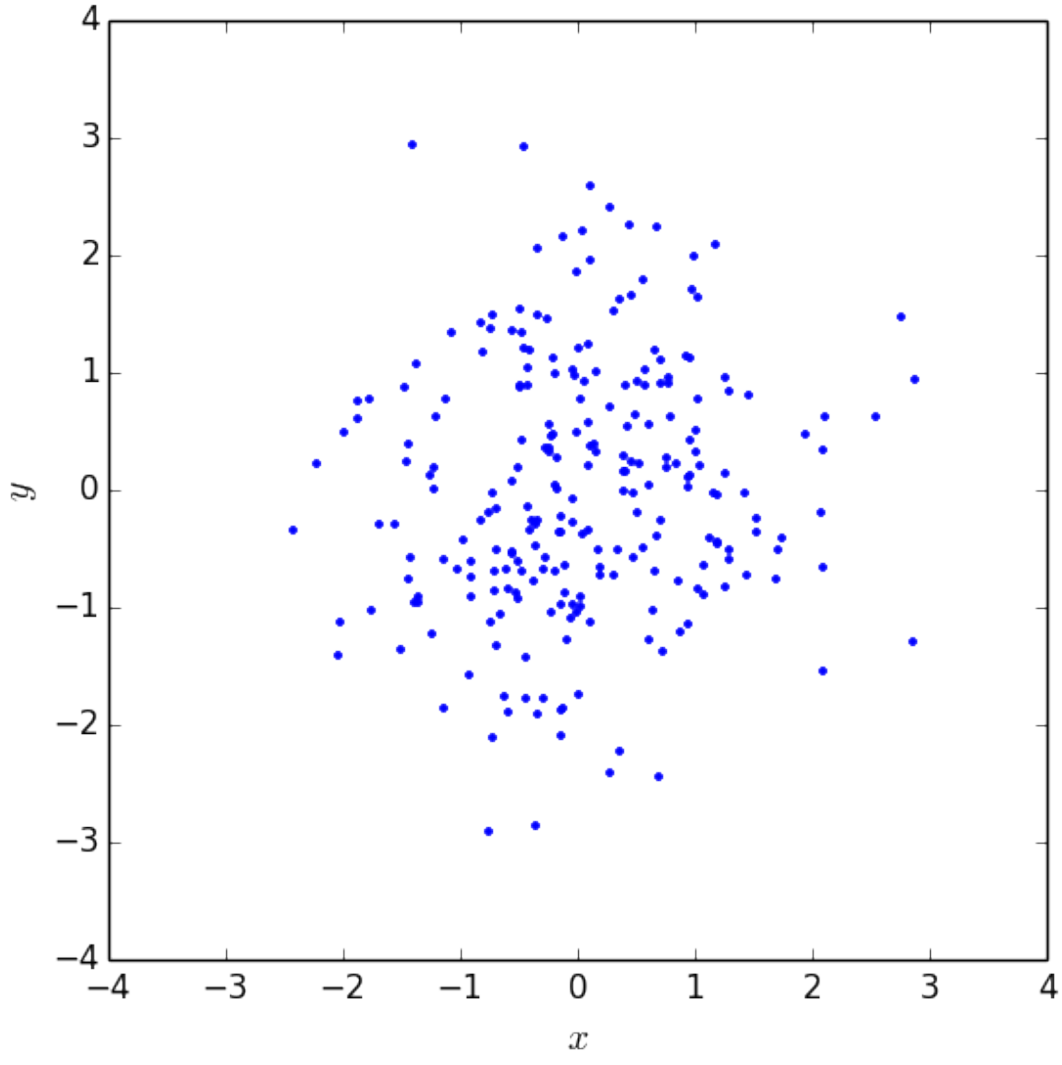
output of gauss-test.py



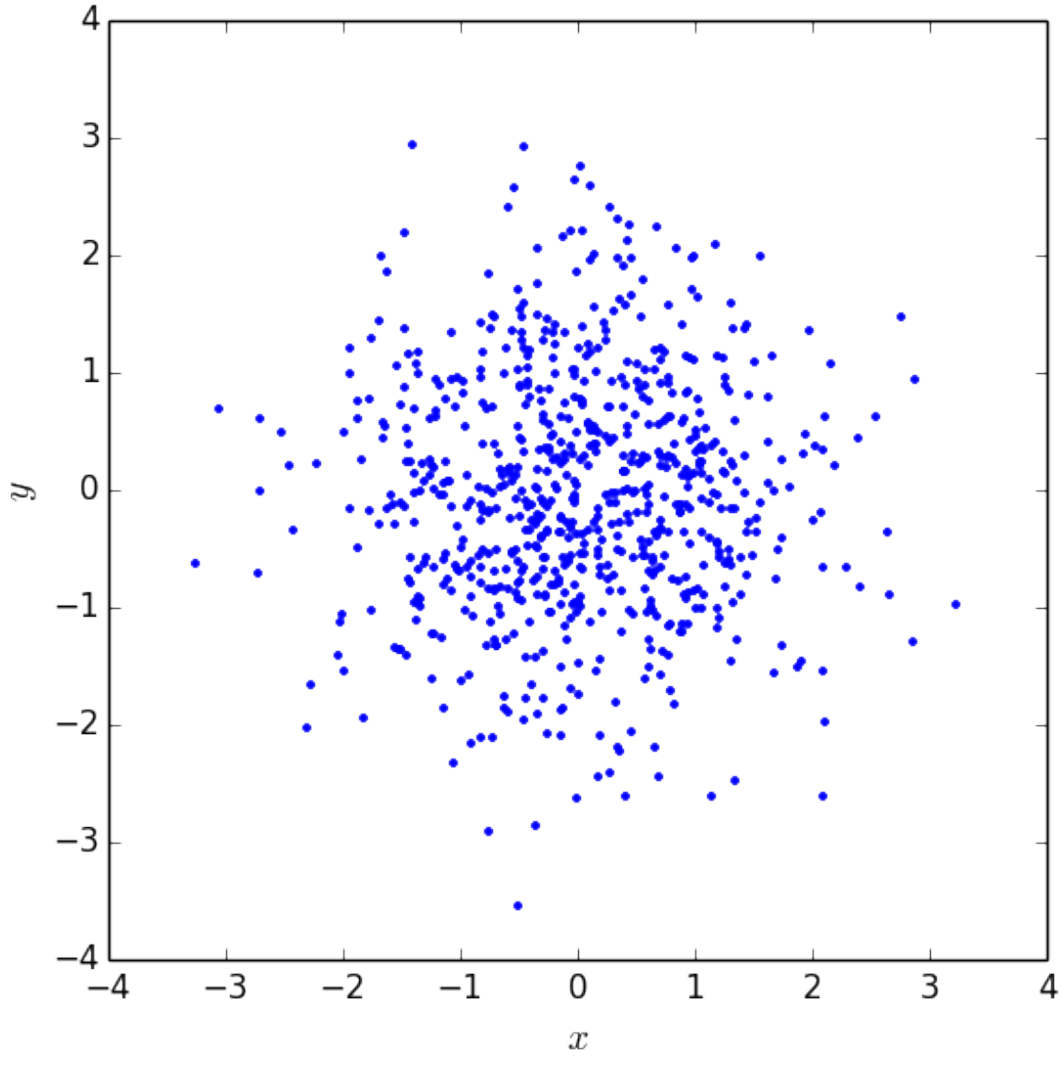
output of gauss-test.py



output of gauss-test.py

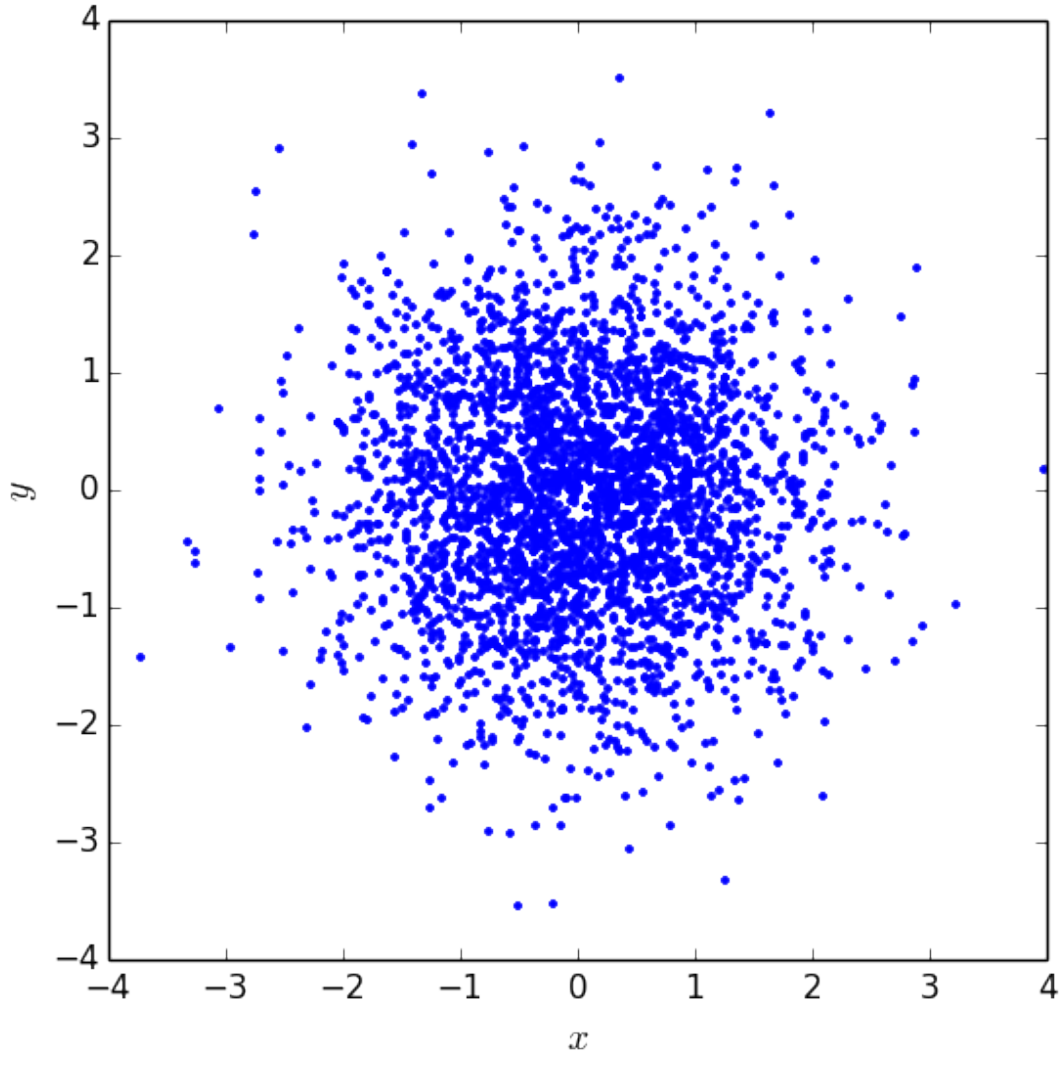


output of gauss-test.py



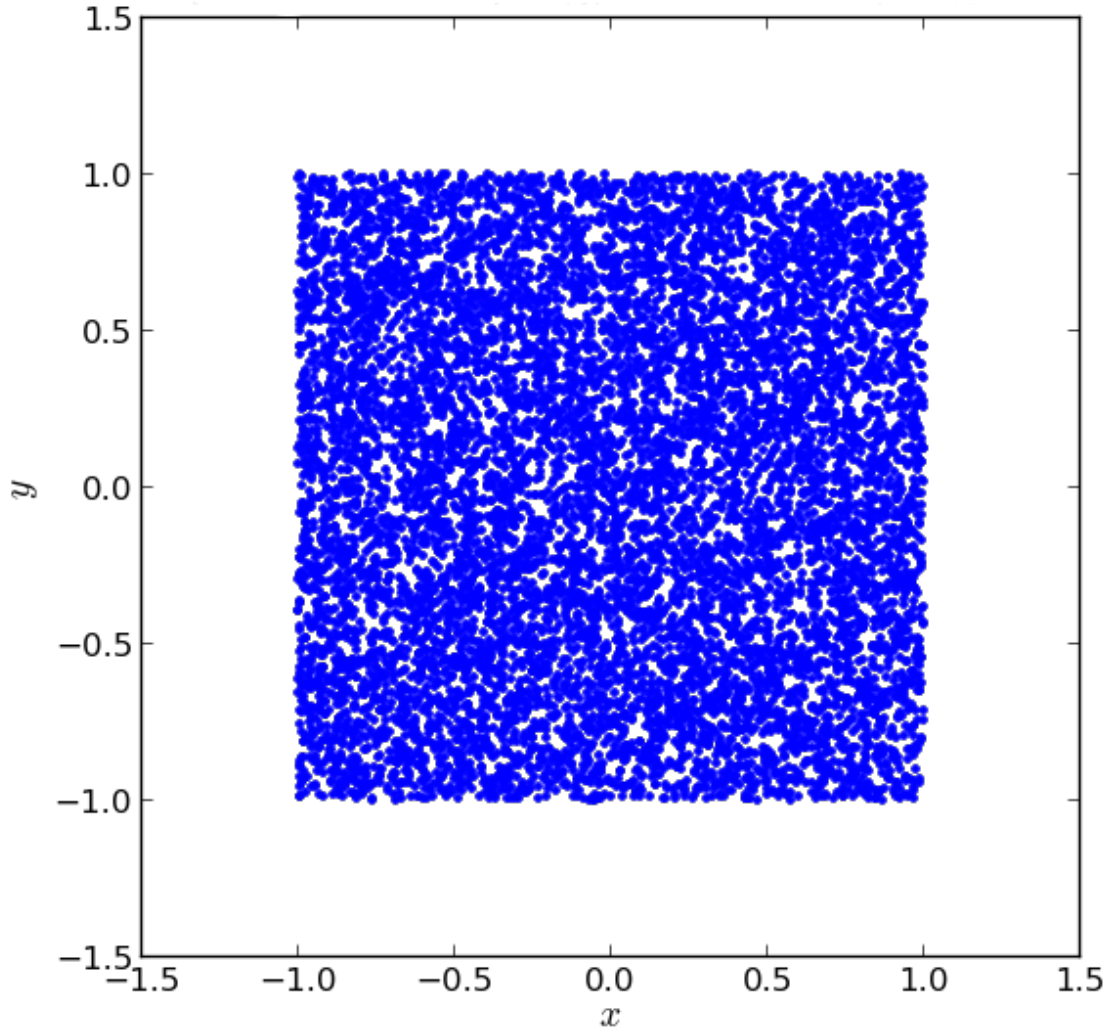
Département de Physique
École normale supérieure

output of gauss-test.py



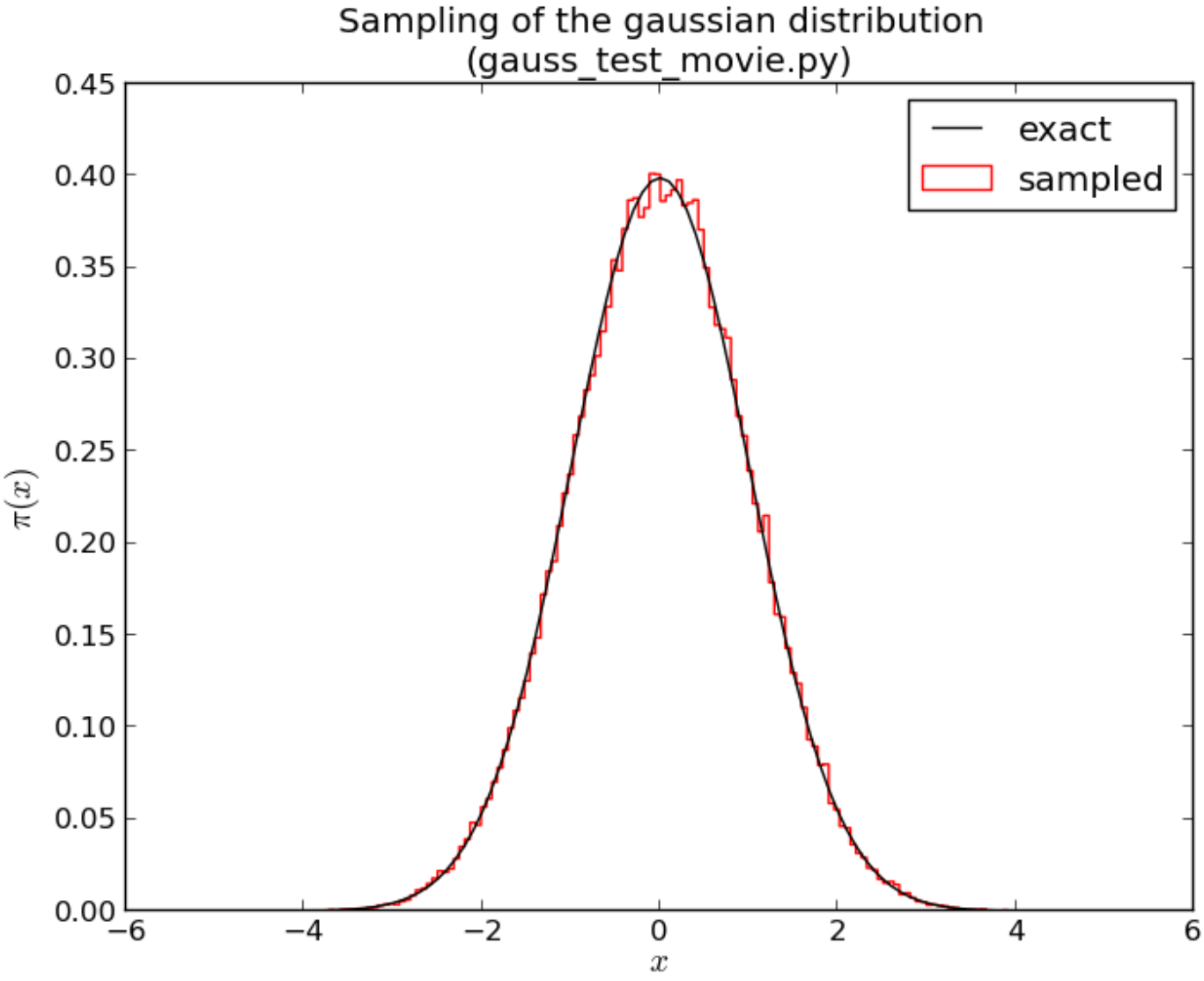
Département de Physique
École normale supérieure

Isotropy is exceptional



Département de Physique
École normale supérieure

test of gauss-test.py



Gaussians in three dimensions

- Isotropy of Gaussians in two dimensions can be checked in `gauss-test.py`.
- Let's check three dimensions (using `random.gauss` from Python).

Gauss-3d.py

```
import random, math

nsamples = 100
for sample in xrange(nsamples):
    x, y, z = (random.gauss(0.0, 1.0),
               random.gauss(0.0, 1.0),
               random.gauss(0.0, 1.0))
    print x, y, z
```

Département de Physique
École normale supérieure

Movie output of Gauss-3d

Haircut - random points on the surface of a sphere

Département de Physique
École normale supérieure

direct-surface-3d.py

```
import random, math

nsamples = 50
for sample in xrange(nsamples):
    x, y, z = (random.gauss(0.0, 1.0),
               random.gauss(0.0, 1.0),
               random.gauss(0.0, 1.0))
    radius = math.sqrt(x ** 2 + y ** 2 + z ** 2)
    print x / radius, y / radius, z / radius
```

direct-surface.py

Département de Physique
École normale supérieure

analyzing gauss-d-dim.py

$$1 = \left(\frac{1}{\sqrt{2\pi}}\right)^d \int \dots \int dx_0 \dots dx_{d-1}\, e^{-\frac{1}{2}(x_0^2+\dots+x_{d-1}^2)}$$

Département de Physique
École normale supérieure

analyzing gauss-d-dim.py

$$1 = \left(\frac{1}{\sqrt{2\pi}}\right)^d \int \dots \int dx_0 \dots dx_{d-1}\, e^{-\frac{1}{2}(x_0^2+\dots+x_{d-1}^2)}$$

$$(x_0, \dots, x_{d-1}) \longrightarrow (r, \Omega)$$

Département de Physique
École normale supérieure

analyzing gauss-d-dim.py

$$1 = \left(\frac{1}{\sqrt{2\pi}}\right)^d \int \dots \int dx_0 \dots dx_{d-1} \, e^{-\frac{1}{2}(x_0^2+\dots+x_{d-1}^2)}$$

$$(x_0, \dots, x_{d-1}) \longrightarrow (r, \Omega)$$

Département de Physique
École normale supérieure

analyzing gauss-d-dim.py

$$1 = \left(\frac{1}{\sqrt{2\pi}}\right)^d \int \ldots \int dx_0 \ldots dx_{d-1}\, e^{-\frac{1}{2}(x_0^2+\ldots+x_{d-1}^2)}$$

$$(x_0, \ldots, x_{d-1}) \longrightarrow (r, \Omega)$$

$$x_0^2 + \ldots + x_{d-1}^2 = r^2$$

Département de Physique
École normale supérieure

analyzing gauss-d-dim.py

$$1 = \left(\frac{1}{\sqrt{2\pi}}\right)^d \int \ldots \int dx_0 \ldots dx_{d-1}\, e^{-\frac{1}{2}(x_0^2+\ldots+x_{d-1}^2)}$$

$$(x_0, \ldots, x_{d-1}) \longrightarrow (r, \Omega)$$

$$x_0^2 + \ldots + x_{d-1}^2 = r^2$$

$$dx_0 \ldots dx_{d-1} = r^{d-1} dr d\Omega$$

Département de Physique
École normale supérieure

analyzing gauss-d-dim.py

$$1 = \left(\frac{1}{\sqrt{2\pi}}\right)^d \int \ldots \int dx_0 \ldots dx_{d-1}\, e^{-\frac{1}{2}(x_0^2+\ldots+x_{d-1}^2)}$$

$$(x_0, \ldots, x_{d-1}) \longrightarrow (r, \Omega)$$

$$x_0^2 + \ldots + x_{d-1}^2 = r^2$$

$$dx_0 \ldots dx_{d-1} = r^{d-1} dr d\Omega$$

$$\ldots = \left(\frac{1}{\sqrt{2\pi}}\right)^d \int_0^\infty dr\, r^{d-1}\, e^{-\frac{1}{2}r^2} \int d\Omega$$

Département de Physique
École normale supérieure

analyzing gauss-d-dim.py

$$1 = \left(\frac{1}{\sqrt{2\pi}}\right)^d \int \dots \int dx_0 \dots dx_{d-1}\, e^{-\frac{1}{2}(x_0^2+\dots+x_{d-1}^2)}$$

$$(x_0, \dots, x_{d-1}) \longrightarrow (r, \Omega)$$

$$x_0^2 + \dots + x_{d-1}^2 = r^2$$

$$dx_0 \dots dx_{d-1} = r^{d-1} dr d\Omega$$

$$\dots = \left(\frac{1}{\sqrt{2\pi}}\right)^d \int_0^\infty dr\, r^{d-1}\, e^{-\frac{1}{2}r^2} \int d\Omega$$

Département de Physique
École normale supérieure

analyzing direct-surface.py

$$1 = \left(\frac{1}{\sqrt{2\pi}}\right)^d \boxed{\int \dots \int dx_0 \dots dx_{d-1}\, e^{-\frac{1}{2}(x_0^2+\dots+x_{d-1}^2)}}$$

$$(x_0, \dots, x_{d-1}) \longrightarrow (r, \Omega)$$

$$x_0^2 + \dots + x_{d-1}^2 = r^2$$

$$dx_0 \dots dx_{d-1} = r^{d-1} dr d\Omega$$

$$\dots = \left(\frac{1}{\sqrt{2\pi}}\right)^d \boxed{\int_0^\infty dr\, r^{d-1}\, e^{-\frac{1}{2}r^2} \int d\Omega}$$

- replacing $r \to 1$ preserves distribution of Ω (haircut)

Département de Physique
École normale supérieure

Uniform samples in sphere

$$1 = \left(\frac{1}{\sqrt{2\pi}}\right)^d \boxed{\int \dots \int dx_0 \dots dx_{d-1}\, e^{-\frac{1}{2}(x_0^2+\dots+x_{d-1}^2)}}$$

$$(x_0, \dots, x_{d-1}) \longrightarrow (r, \Omega)$$

$$x_0^2 + \dots + x_{d-1}^2 = r^2$$

$$dx_0 \dots dx_{d-1} = r^{d-1} dr d\Omega$$

$$\dots = \left(\frac{1}{\sqrt{2\pi}}\right)^d \boxed{\int_0^\infty dr\, r^{d-1}\, e^{-\frac{1}{2}r^2} \int d\Omega}$$

replacing the r distribution:

$$\pi(r) = r^{d-1} \quad (0 < r < 1)$$

...sampled through $(\texttt{random.uniform(0.0, 1.0)})^{(1/d)}$

Département de Physique
École normale supérieure

direct-sphere-3d.py

```
import random, math

nsamples = 100
for sample in xrange(nsamples):
    x, y, z = (random.gauss(0.0, 1.0),
               random.gauss(0.0, 1.0),
               random.gauss(0.0, 1.0))
    length = random.uniform(0.0, 1.0) ** (1.0 / 3.0) \
                    / math.sqrt(x ** 2 + y ** 2 + z ** 2)
    print x * length, y * length, z * length
```

Output of direct-sphere-3d.py

Newton vs. Boltzmann

- Equiprobability:

$$\pi^{\text{Newton}}(\mathcal{C}) = \pi^{\text{Boltzmann}}(\mathcal{C}) = \begin{cases} \text{constant} & \text{for legal } \mathcal{C} \\ 0 & \text{else} \end{cases}$$

- Newton = Boltzmann, but something missing on the right.

Département de Physique
École normale supérieure

Conservation of kinetic energy

$$E_{\mathrm{kin}} = \frac{1}{2} m \left(\mathbf{v}_0^2 + \ldots + \mathbf{v}_{N-1}^2 \right)$$

Conservation of kinetic energy

$$E_{\mathrm{kin}} = \frac{1}{2} m \left(\mathbf{v}_0^2 + \ldots + \mathbf{v}_{N-1}^2 \right)$$

$$\mathbf{v}_k = \begin{pmatrix} v_k^x \\ v_k^y \end{pmatrix}$$

Conservation of kinetic energy

$$E_{\text{kin}} = \frac{1}{2} m \left(\mathbf{v}_0^2 + \ldots + \mathbf{v}_{N-1}^2 \right)$$

$$\mathbf{v}_k = \begin{pmatrix} v_k^x \\ v_k^y \end{pmatrix}$$

$$\mathbf{v}_k^2 = \left(v_k^x\right)^2 + \left(v_k^y\right)^2$$

Département de Physique
École normale supérieure

Maxwell distribution

$$\pi(\mathbf{v}_0, \ldots, \mathbf{v}_{N-1}) = \begin{cases} 1 & \text{if velocities are legal} \\ 0 & \text{otherwise} \end{cases}$$

- random points on the surface of a sphere in dN dimensions.

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

- haircut not necessary for $N \to \infty$.

Département de Physique
École normale supérieure

Conclusion

Concepts considered:

- Integration vs. Sampling
- Sample transformation
- Equiprobability inside a sphere
- Equiprobability on the surface of a sphere
- Maxwell distribution

Algorithms considered:

- gauss-test.py
- gauss-3d.py
- direct-surface.py
- direct-sphere.py