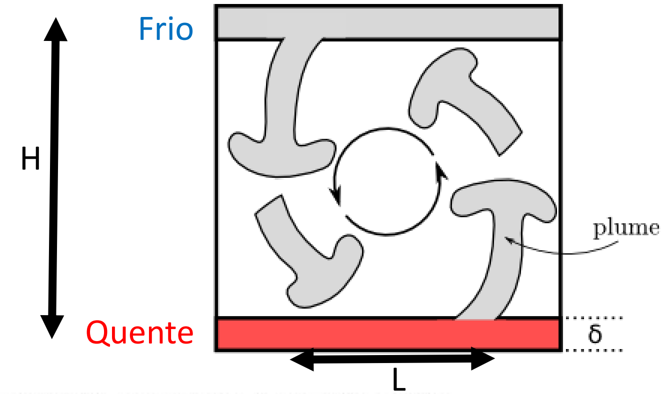
Aula 22

Visualização de Trajetórias.

Atrator de Lorenz.

Convecção de Rayleigh-Benard

Henri Bénard (1901) Les tourbillons cellulaires dans une nappe liquide. - Méthodes optiques d’observation et d’enregistrement. J. Phys. Theor. Appl., 1901, 10 (1), pp.254-266. <10.1051/jphystap:019010010025400>. <jpa-00240502>



Vista superior

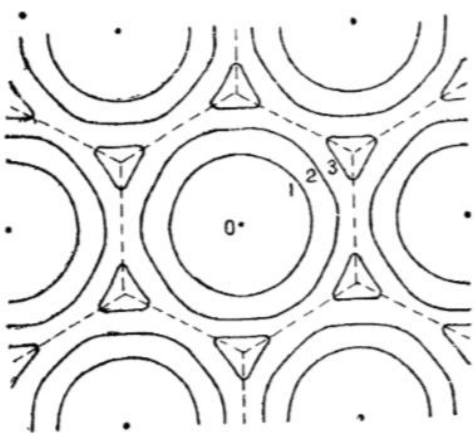
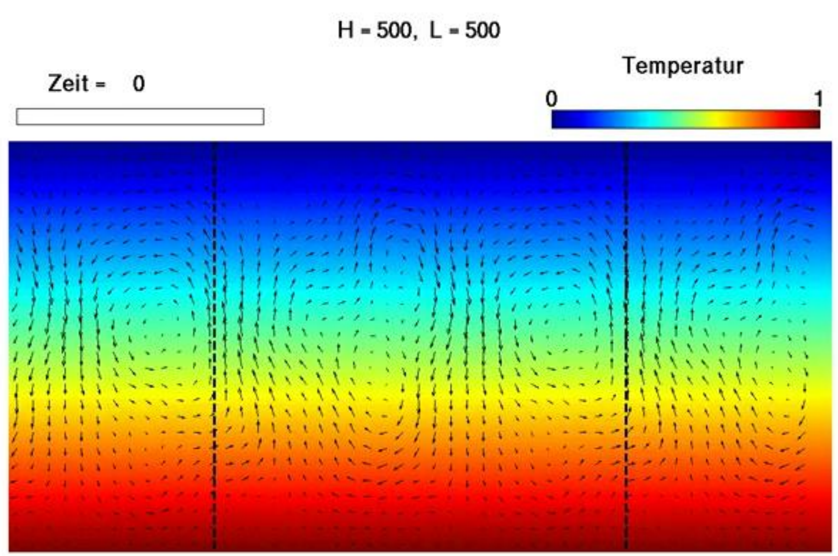


Fig. 1.



https://www.mis.mpg.de/applan/research/rayleigh.html

O processo de convecção de Rayleigh-Bénard

Cloud albedo calculated from large eddy simulation of closed and open cellular structures.

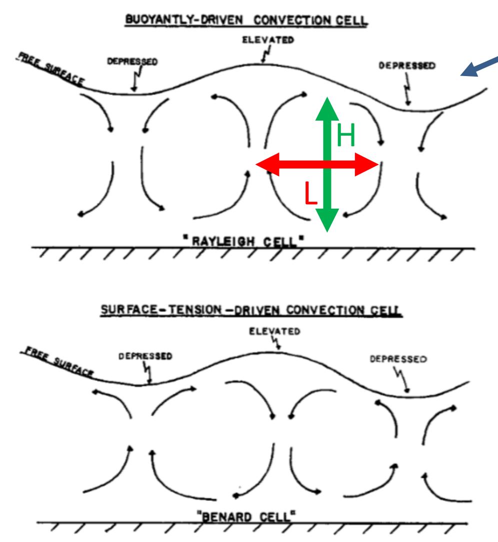
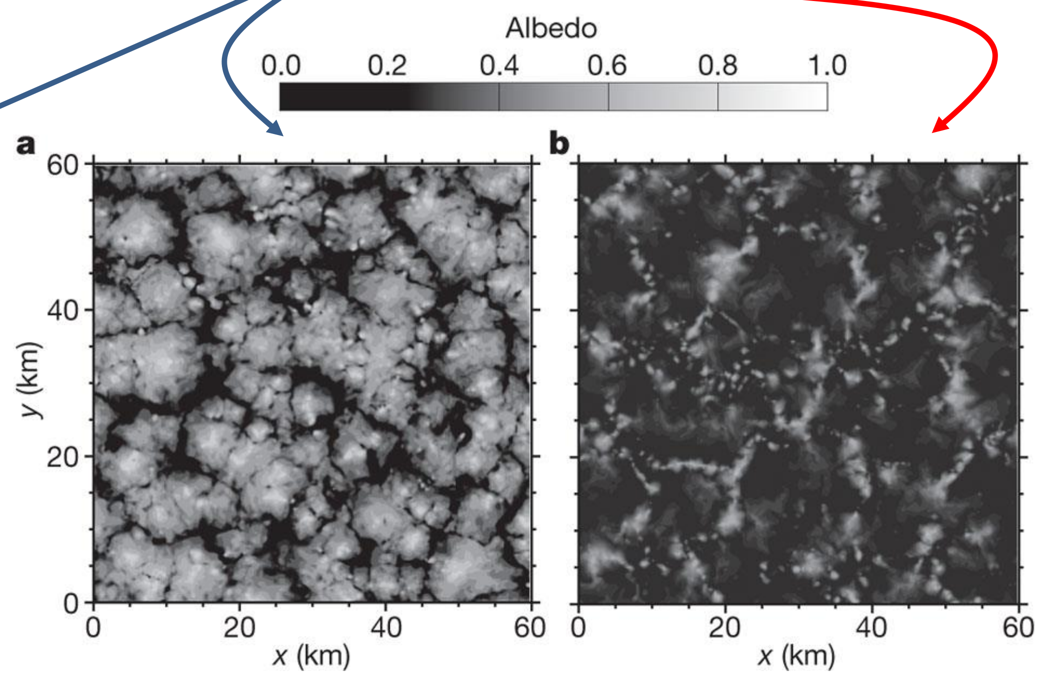


FIG. 4. Nature of free surface deformations for buoyantly-driven and surface-tension-driven convection cells.

Agee et al, BAMS, 1973



nature

G Feingold *et al. Nature* **466**, 849-852 (2010) doi:10.1038/nature09314

Instabilidade de Rayleigh-Bénard

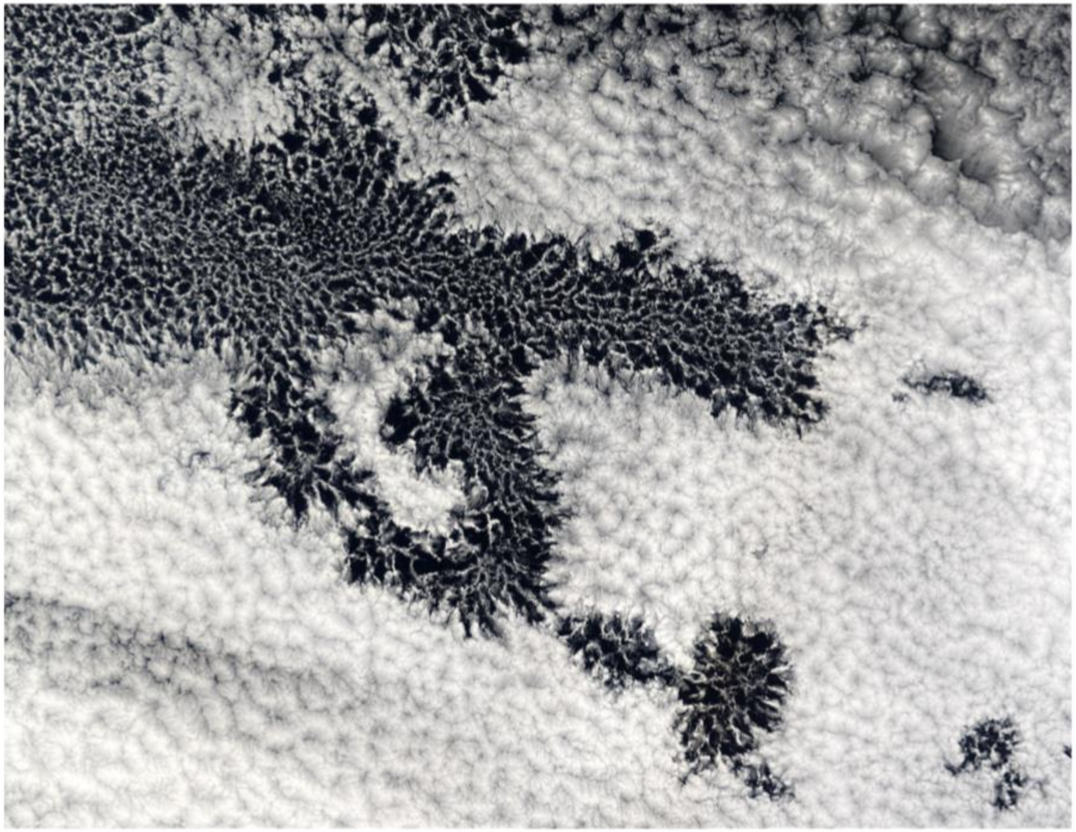
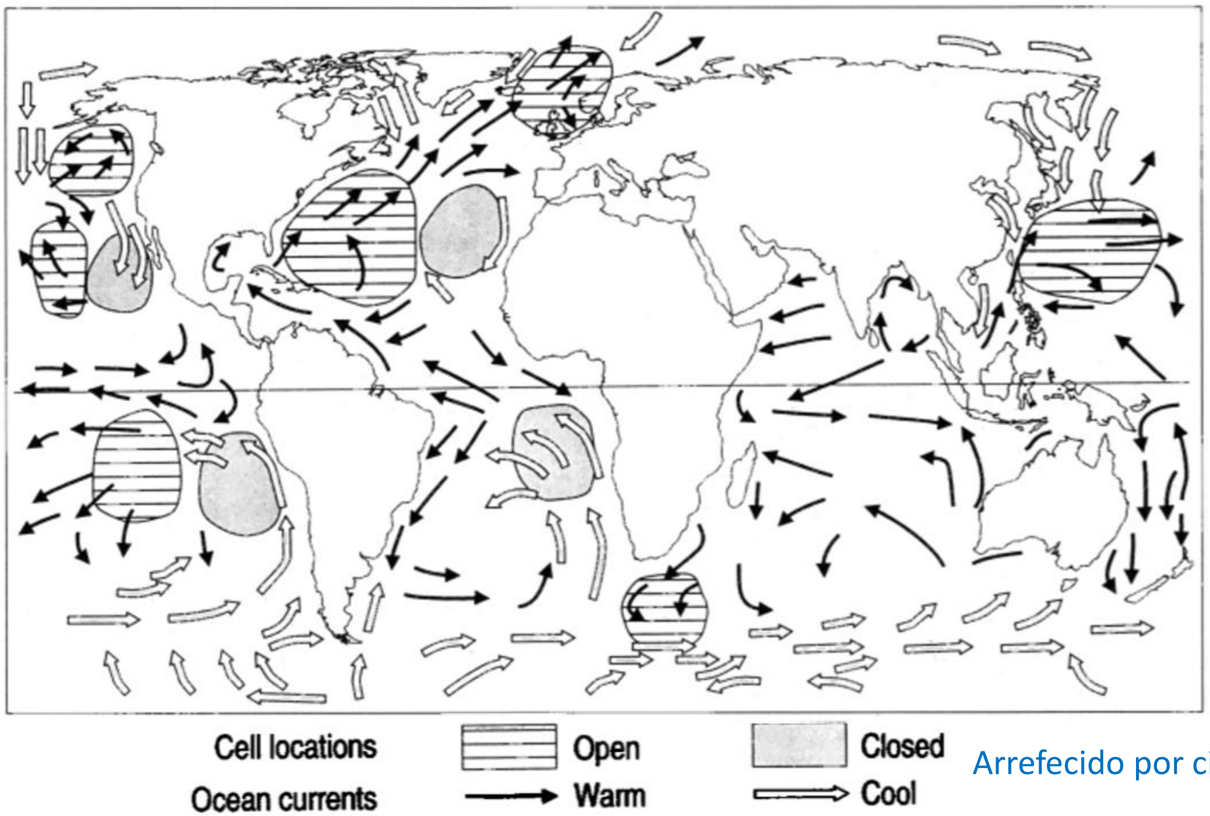


Figure 5: Example of a region of open cellular convection (dark cell interiors, with bright cell walls) embedded in a broader region of closed cellular convection (bright cells with darkened cell walls). Open cellular regions have been hypothesized to be envelopes where drizzle is more prevalent.

Stevens B (2005) Annu. Rev. Earth Planet. Sci



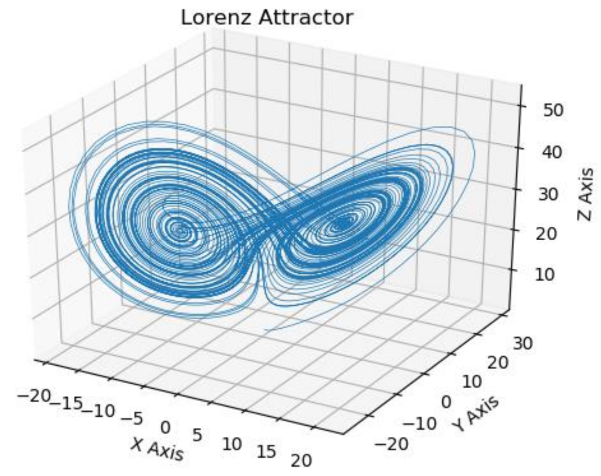
Arrefecido por cima

Aquecido por baixo

Figure 7. Global distribution of mesoscale cellular convection (MCC) [after *Agee*, 1987]. Reprinted with kind permission from Elsevier Science–NL, Sara Burgerhartstraat 25, 1055 KU Amsterdam, Netherlands.

O atractor de Lorenz

$$\begin{cases} \dfrac{dx}{dt} = \sigma(y - x) \\ \dfrac{dy}{dt} = x(\rho - z) - y, \\ \dfrac{dz}{dt} = xy - \beta z \end{cases} \begin{cases} \sigma = 10 \\ \rho = 28 \\ \beta = 2.667 \end{cases}$$



3 equações diferenciais acopladas
Termos não lineares

Variáveis:

x – intensidade da convecção

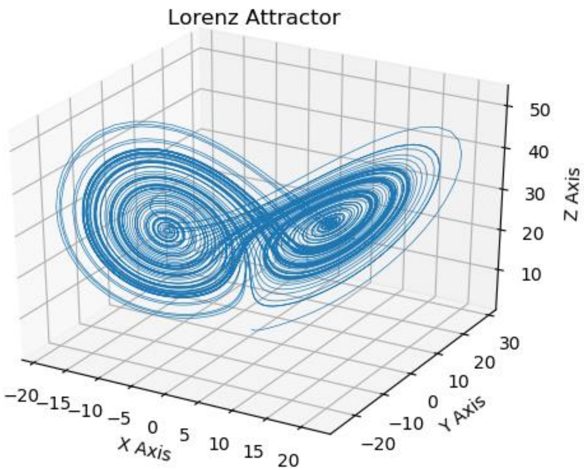
y – forçamento térmico (ΔT entre ascendente e descendente)

z – perturbação do gradiente vertical de temperatura

Parâmetros: σ, ρ, β (dependem da viscosidade, da difusividade térmica e da geometria)

$$\begin{cases} \dot{x} \equiv \dfrac{dx}{dt} = \sigma(y - x) \\ \dot{y} \equiv \dfrac{dy}{dt} = x(\rho - z) - y, \\ \dot{z} \equiv \dfrac{dz}{dt} = xy - \beta z \end{cases} \begin{cases} \sigma = 10 \\ \rho = 28 \\ \beta = 2.667 \end{cases}$$

```
import numpy as np;import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
def lorenz(x, y, z, s=10, r=28, b=2.667):
    x_dot = s*(y - x)
    y_dot = r*x - y - x*z
    z_dot = x*y - b*z
    return x_dot, y_dot, z_dot
```



Posição inicial das partículas

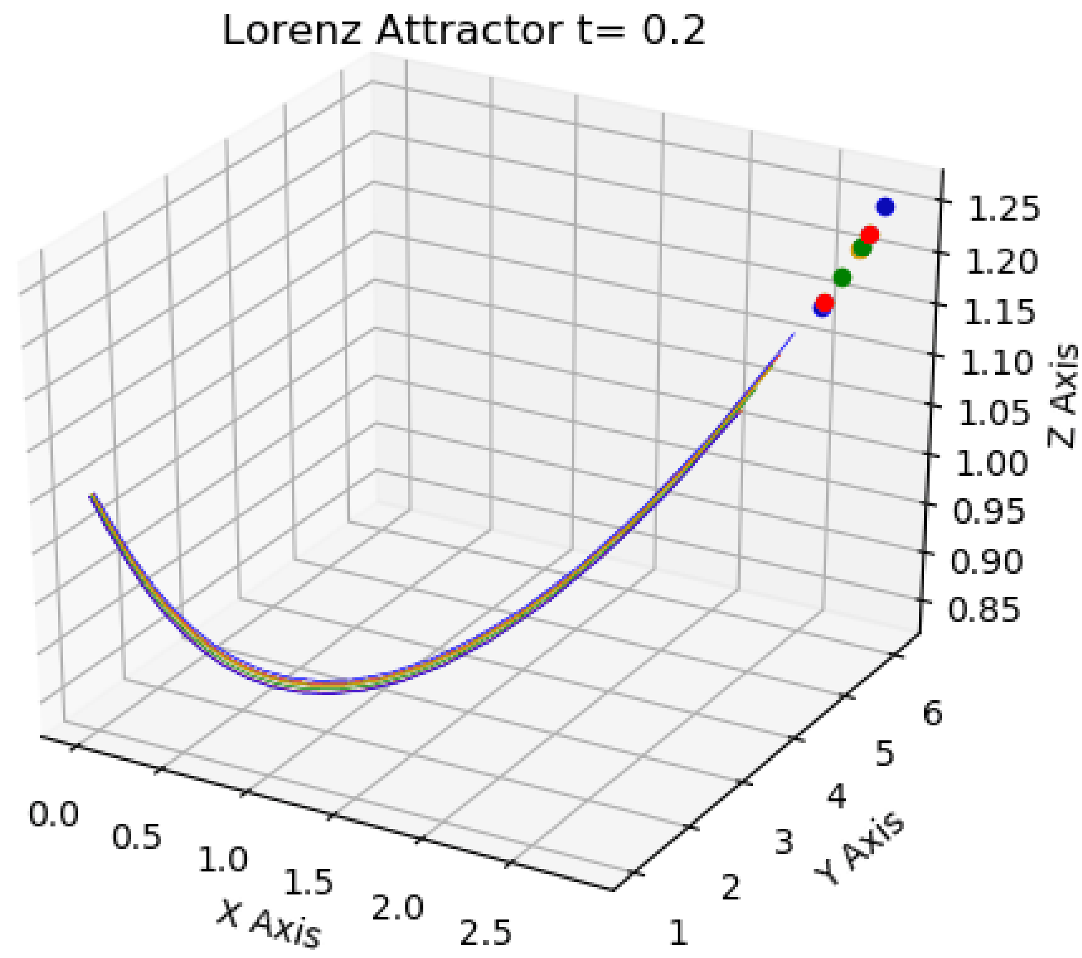
```
dt=0.01;stepCnt=2000
num=4 #n° de partículas
xs = np.empty((stepCnt + 1,num))
ys = np.empty((stepCnt + 1,num))
zs = np.empty((stepCnt + 1,num))
cor=['red','green','blue','orange']
for k in range(num):
    xs[0,k],ys[0,k],zs[0,k]=\
           (0.,1.+0.1*np.random.rand(), 1.05)
```

Integração temporal

```
for k in range(num): #trajetória

    for i in range(stepCnt): #tempo
        x_dot,y_dot,z_dot=\
            lorenz(xs[i,k], ys[i,k], zs[i,k])

        xs[i+1,k]=xs[i,k]+(x_dot * dt)
        ys[i+1,k]=ys[i,k]+(y_dot * dt)
        zs[i+1,k]=zs[i,k]+(z_dot * dt)
```


Lorenz Attractor t= 0.2
1.25
1.20
1.15
1.10
1.05
1.00
0.95
0.90
0.85
Z Axis
0.0
0.5
1.0
1.5
2.0
2.5
X Axis
1
2
3
4
5
6
Y Axis

Desenho das trajetórias e movie

```
for i in range(20,stepCnt,20): #frame de 20 em 20 passos
    fig = plt.figure(1)
    ax = fig.gca(projection='3d')
    ax.scatter(1+np.max(xs[:,0]),\
            1+np.max(ys[:,0]),np.max(zs[:,0])+5,alpha=0)

    for k in range(num):
        ax.plot(xs[0:i,k], ys[0:i,k], zs[0:i,k],\
                lw=0.5,color=cor[k])
        ax.scatter(xs[i,k],ys[i,k],zs[i,k],color=cor[k])
        ax.set_title("Lorenz Attractor t=%4.1f "%(i*dt))
    plt.pause(0.1)
    plt.show()
    if movie!='':
        frame=movie+str(i)+'.png'
        pngs.append(frame)
        plt.savefig(frame)
    plt.clf()
```

gif

```
if len(pngs)>0:
    import imageio
    import os
    images=[]
    for frame in pngs:
        images.append(imageio.imread(frame))
        os.remove(frame)
    imageio.mimsave('lorenz.gif’,\
        images,duration=0.1)
```

Seguindo 4 partículas que começam quase no mesmo ponto

Ao fim de pouco tempo, cada partícula segue trajetorias completamente diferentes:

CAOS