

# Sciences or One Science?



**Mgr. Jiří Kvita, Ph.D.**  
**Badatelský víkendový seminář 2023**  
**Společná laboratoř optiky UP**

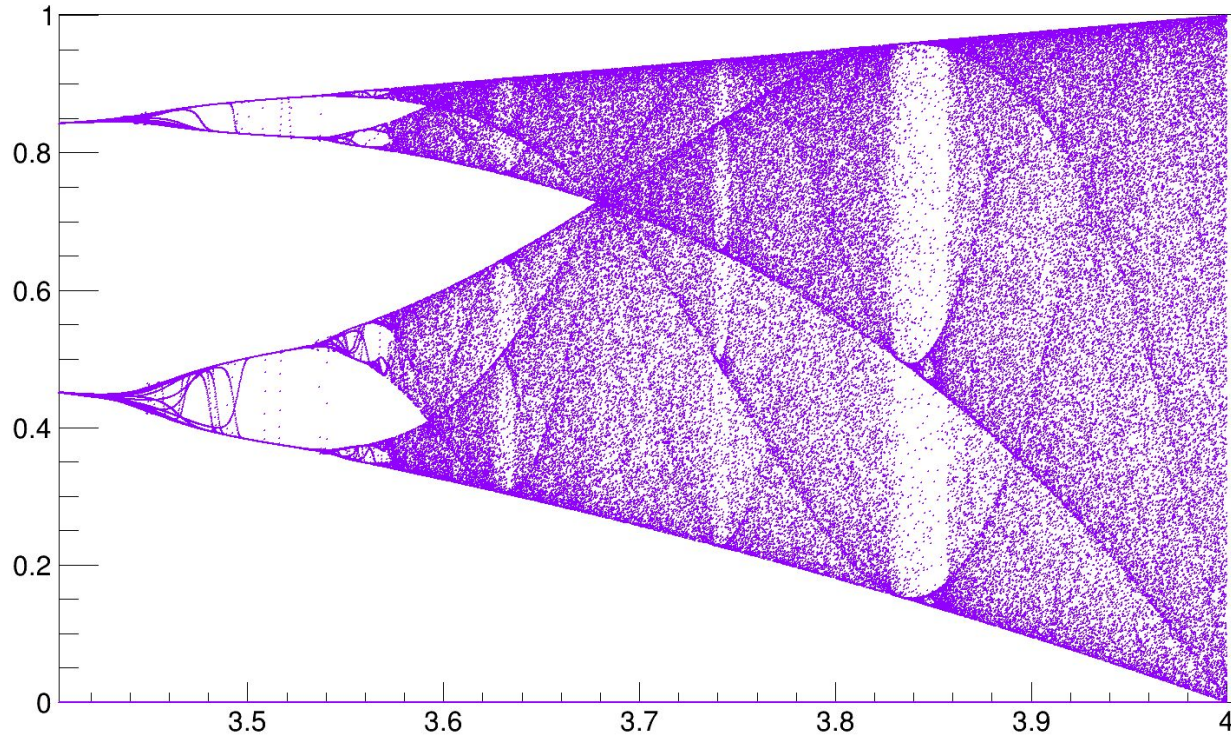


# Model



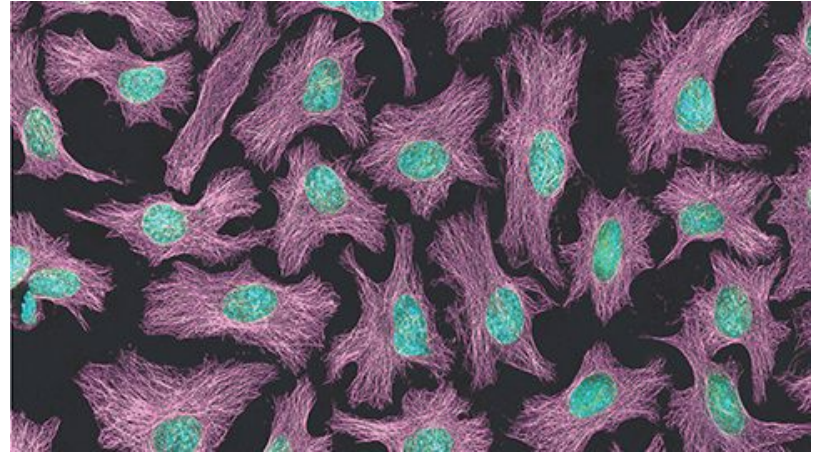
# Model

$$\frac{dN}{dt} = rN \left( 1 - \frac{N}{K} \right)$$



```
14
15 import ROOT
16 from math import sqrt, pow, log, exp
17 import os, sys, getopt
18
19 import random
20
21 cans = []
22 stuff = []
23
24 #####
25
26 def iterate(y0, r, cycles, getFullSeq = False):
27     y = 1.*y0
28     ys = []
29     if getFullSeq:
30         ys.append(y)
31     for i in range(0, cycles):
32         y = r*y*(1.-y)
33         if getFullSeq:
34             ys.append(1.*y)
35     return y, ys
36
37 #####
38
39 def MakeSeqGraphs(YSs):
40     Grs = []
41     for ys in YSs:
42         gr = ROOT.TGraph()
43         for i in range(0, len(ys)):
44             gr.SetPoint(i, i, ys[i])
45         Grs.append(gr)
46     return Grs
47
48 #####
49
50 def MakeGraph(y0, rmin, rmax, n, step, cycles):
51     gr = ROOT.TGraph()
52     rp = -1
53     Ys = []
54     for r in [rmin + step * j for j in range(0, n)]:
```

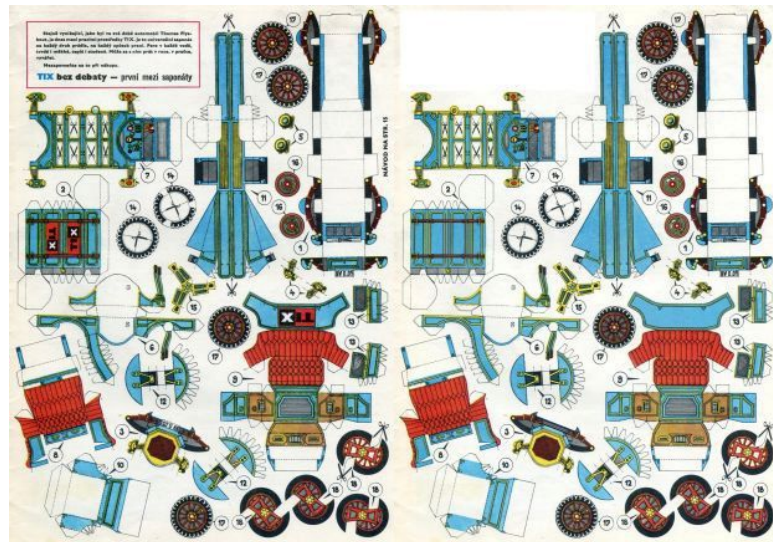
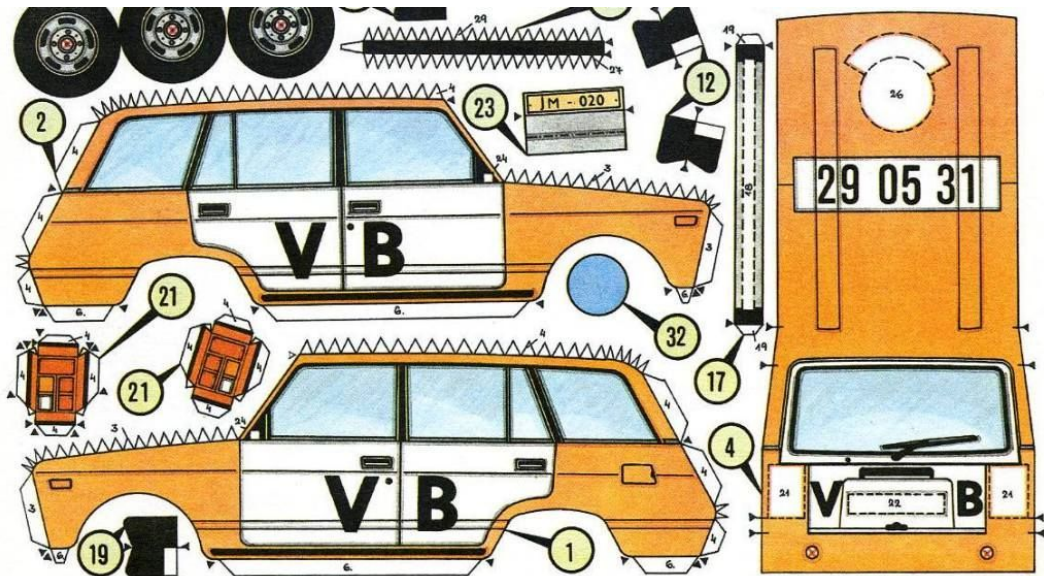
# Model



# Model

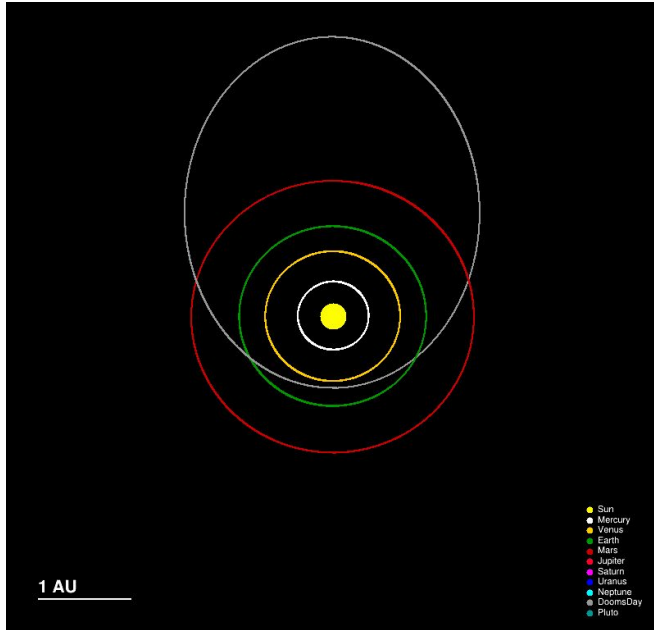
$$\begin{aligned}
 L[\pi, \vec{W}, B, h] = & -\frac{1}{2}\pi_a \square \pi_a - \frac{1}{2}h(\square + m_h^2)h - \lambda(\pi_a^2 + h^2)^2 \\
 & - 4\lambda v h(\pi_a^2 + h^2) - \frac{g}{2}\partial^\mu \pi_1(W_\mu^3 \pi_2 - W_\mu^2 \pi_3) \\
 & - \frac{g}{2}\partial^\mu \pi_2(W_\mu^1 \pi_3 - W_\mu^3 \pi_1) - \frac{g}{2}\partial^\mu \pi_3(W_\mu^2 \pi_1 - W_\mu^1 \pi_2) \\
 & + g\partial^\mu h(\vec{W}_\mu \cdot \vec{\pi}) - \frac{g'}{2}(\pi_1 \partial_\mu \pi_2 - \pi_2 \partial_\mu \pi_1)B^\mu - g\partial_\mu h \pi_3 B^\mu \\
 & + \frac{1}{2}m_W^2 \vec{W}_\mu \cdot \vec{W}^\mu + \frac{1}{2}m_B^2 B_\mu B^\mu - m_W m_B W_\mu^3 B^\mu \\
 & + \frac{g^2}{8}(\vec{W}_\mu \cdot \vec{\pi})(\vec{W}^\mu \cdot \vec{\pi}) + \frac{g'^2 v}{4}h B_\mu B^\mu \\
 & + \frac{g'^2}{8}H^2 B_\mu B^\mu - \frac{gg'}{4}h^2 W_\mu^3 B^\mu - \frac{gg'v}{2}h W_\mu^3 B^\mu \\
 & + \frac{g^2}{8}h^2 \vec{W}_\mu \cdot \vec{W}^\mu + \frac{g^2 v}{4}h \vec{W}_\mu \cdot \vec{W}^\mu + \frac{g'^2}{8}B_\mu B^\mu \vec{\pi} \cdot \vec{\pi} \\
 & + \frac{gg'}{4}W_\mu^3 B^\mu \vec{\pi} \cdot \vec{\pi} - \frac{gg'}{2}\pi_3 B_\mu (W_1^\mu \pi_1 + W_2^\mu \pi_2) \\
 & + g' m_W B_\mu (W_1^\mu \pi_2 - W_2^\mu \pi_1) + \frac{gg'}{2}B_\mu (W_1^\mu \pi_2 - W_2^\mu \pi_1)h \\
 & + L_{YM}. \tag{7}
 \end{aligned}$$

# Model



# Model

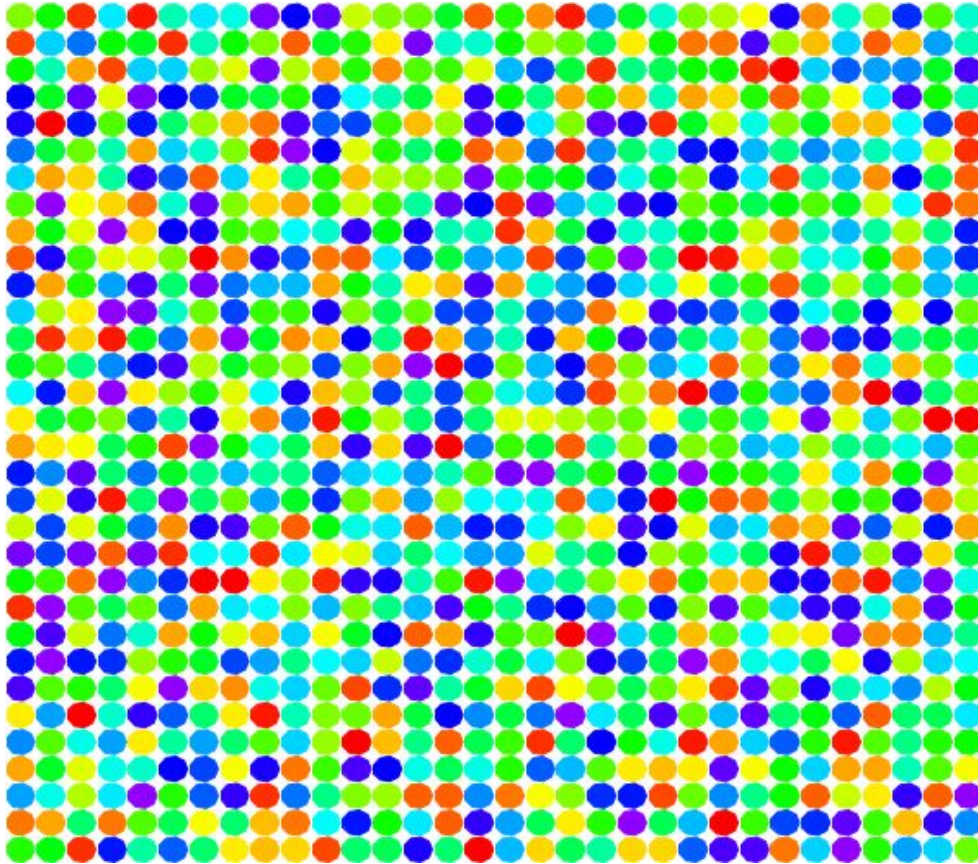
$$m \frac{d^2 \vec{r}}{dt^2} = - \frac{GMm}{r^2} \hat{r}$$



```
17
18 #####
19
20 kday = 24*3600. # seconds
21 kyear = 31556952. # seconds; http://www.kylesconverter.com/time/years-to-seconds
22 kAU = 149597871e3 # m
23 kEarthMass = 5.972e24 # kg
24 kSunMass = 1.989e30 # kgso
25 kSunName = 'Sun'
26 kms = 1000. # 1 km/s = 1000 m/s
27 # grav. const
28 kappa = 6.67408e-11 # m3 kg-1 s-2
29
30 # 3D world:)
31 kdim = 3
32 kepsilon = 1.e-2*kAU
33
34 #####
35
36 def GetDr(x,y):
37     dr = 0.
38     for xx,yy in zip(x,y):
39         dr = dr + pow(xx-yy, 2)
40     if dr > 0:
41         dr = sqrt(dr)
42     return dr
43
44
45 #####
46 def MakeDigitStr(i, digits = 4):
47     tag = str(i)
48     n = digits
49     try:
50         n = int(log10(i))
```

<http://jointlab.upol.cz/kvita/gravity.html>

# Model

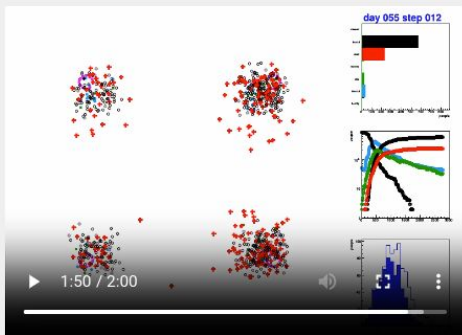


```
10 import numpy as np
11
12 import random
13
14 cans = []
15 stuff = []
16
17 #####
18
19 def ScaleHistosMaxima(hsx, hsy):
20     mmax = 1.2 * max(hsx.GetMaximum(), hsy.GetMaximum())
21     hsx.SetMaximum(mmax)
22     hsy.SetMaximum(mmax)
23
24 #####
25
26
27 def getRandomWalkGraph(nSteps,
28                        xmin, xmax, ymin, ymax,
29                        stepxmin, stepxmax, stepymin, stepymax):
30     gr = ROOT.TGraph()
31     oldx = (xmin+xmax)/2.
32     oldy = (ymin+ymax)/2.
33     mydata = []
34     suml = 0.
35     for i in range(0,nSteps):
36         dx,dy = random.uniform(stepxmin, stepxmax), random.uniform(stepymin, stepymax)
37         newx, newy = oldx + dx, oldy + dy
38         dl2 = pow(oldx - newx, 2) + pow(oldy - newy, 2)
39         if dl2 > 0:
40             suml = suml + sqrt(dl2)
41         gr.SetPoint(i, newx, newy)
42     mydata.append([1.*newx, 1.*newy])
```

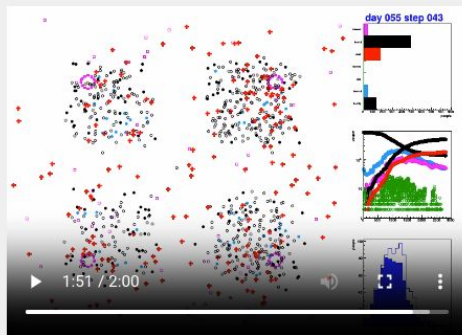


# Model

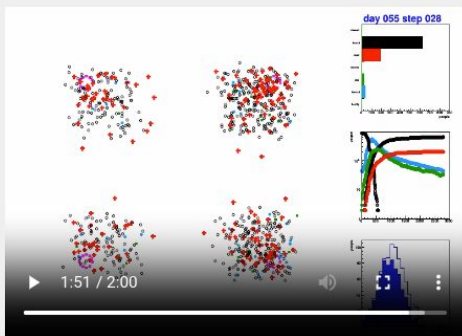
<http://jointlab.upol.cz/kvita/covidSim.html>



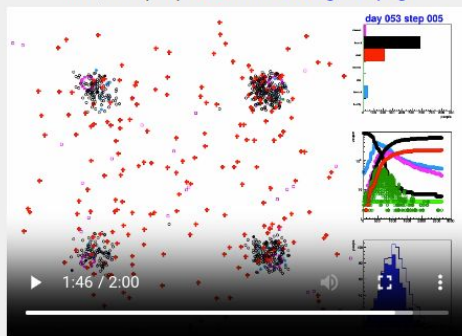
No quarantine, no super-spreaders. Quick rise of **infected** and **sick** people, more people **die**. The **last** **highes** **png**.



No super-spreaders, but **quarantene** imposed 5 times a day, 98% of sick people go and **stay home** and do not infect others. **Infection peak** arrives later, less people **die**. The **last highes** **png**.



No quarantine, **4 super-spreaders**.

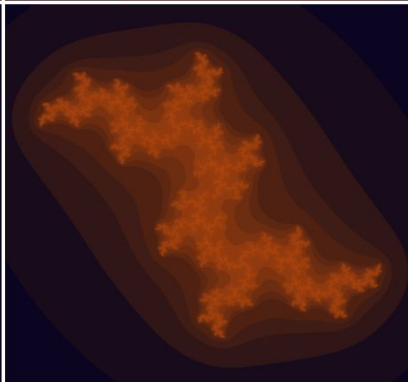
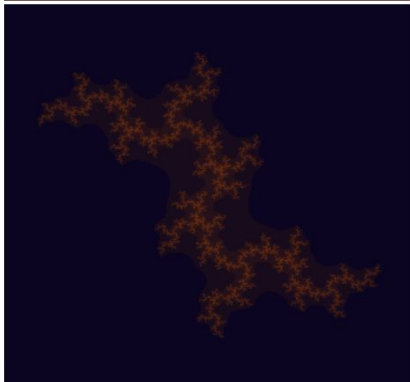
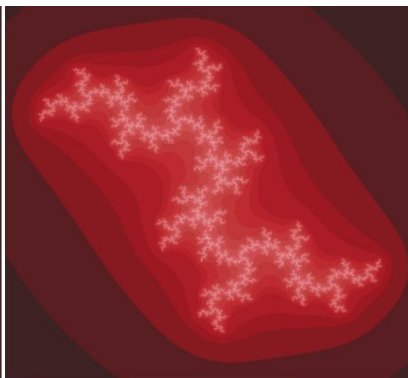
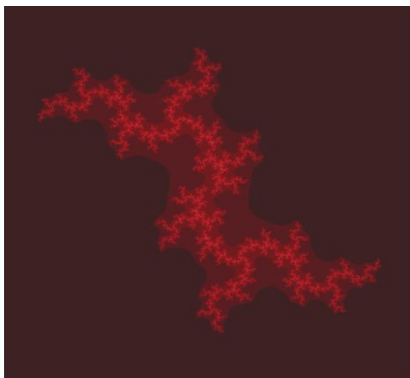
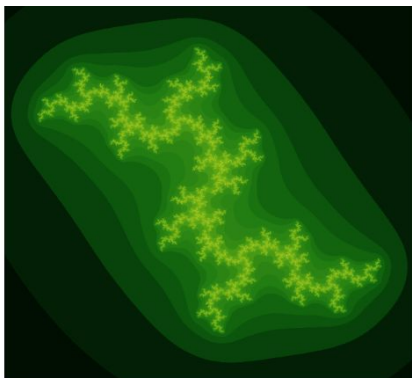


**4 super-spreaders**, but quarantine imposed daily, 90% of sick people go and stay home and do not infect others. Infection peak arrives later, although in this model everybody gets sick, too.

```
#####
51 def MakeAttractors(world):
52     attractors = []
53     # some manually placed attractors
54     # center:
55     sx = (world.GetXmax() + world.GetXmin()) / 2.
56     sy = (world.GetYmax() + world.GetYmin()) / 2.
57     # halfwidths:
58     hwx = (world.GetXmax() - world.GetXmin()) / 2.
59     hwy = (world.GetYmax() - world.GetYmin()) / 2.
60     rmin = 0.02*gkm
61     rmax = 10*gkm # exceeds the world radius, attractor works for all practical distances
62     speed = 2 # exceeding factor over random walk speed
63     sf = 0.55 # scale factor
64     attractors.append(cattractor(sx-sf*hwx, sy-sf*hwy, rmin, rmax, speed))
65     attractors.append(cattractor(sx+sf*hwx, sy+sf*hwy, rmin, rmax, speed))
66     return attractors
67
68 #####
69 def MakeFamily(world, attractors, params, x, y, nAverInFamily):
70     family = []
71     rand = world.GetRand()
72     for i in range(0, int(0.5 + np.random.poisson(nAverInFamily, 2)[0])):
73         id = world.YieldNewId()
74         age = -1
75         while age <= 0:
76             age = rand.gauss(35, 15)
77         if age > params.GetMaxAge():
78             age = params.GetMaxAge()
79         # TODO: randomize x, y within family members
80         status = gHealthy
81         if rand.uniform(0,1) < params.GetInitialSickFraction():
82             status = gInfected
83         # TODO!
84         # randomize attractors for family members!
85         randx = rand.uniform(0,1)
86         randl = 0
87
```

# Model

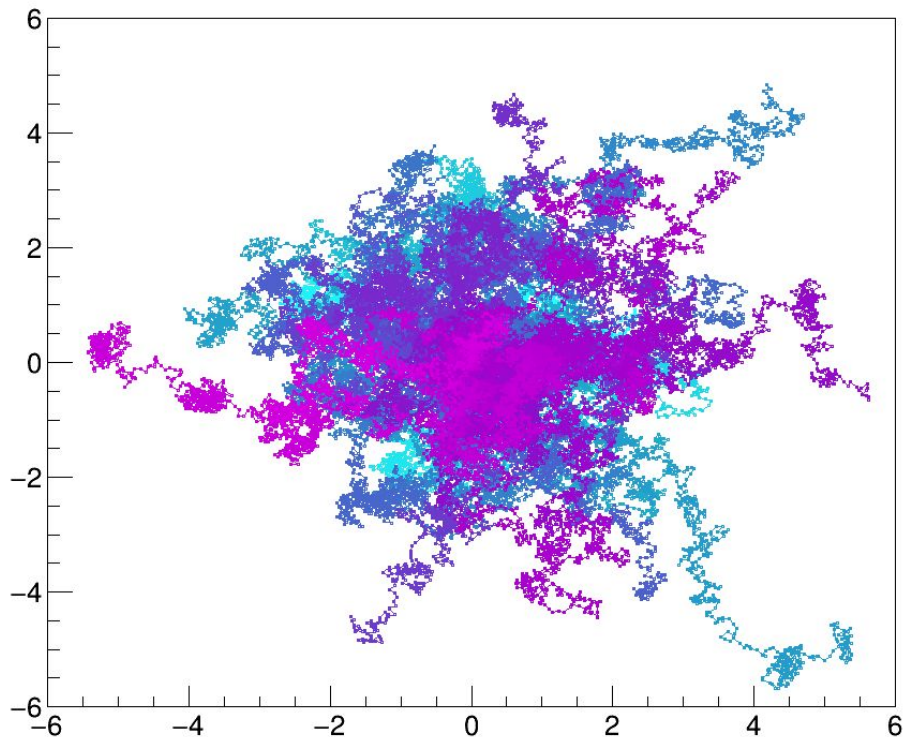
<http://jointlab.upol.cz/kvita/Mandelbrot/index.html>



```
22
23 class MyComplex:
24     def __init__(self, re, in):
25         self.re = re
26         self.in = in
27     def print(self):
28         print('re={:1.3f} in={:1.3f} mag={:1.3f}'.format(self.re, self.in, self.mag()))
29     def mag(self):
30         mag2 = pow(self.re,2) + pow(self.in,2)
31         if mag2 > 0:
32             return sqrt(mag2)
33         else:
34             return 0
35     def __add__(self, other):
36         result = MyComplex(self.re + other.re, self.in + other.in)
37         return result
38     def __truediv__(self, other):
39         result = MyComplex( (self.re*other.re + self.in*other.in)/ other.mag(), (self.
40         return result
41     def __mul__(self, other):
42         result = MyComplex(self.re*other.re - self.in*other.in, self.re*other.in + self.
43         return result
44     #def __mul__(self, real):
45     #    result = MyComplex(self.re*real, self.in*real)
46     #    return result
47
48 #####
49 #####
50 #####
51
52 def getNext(z, c, maIntag):
53     # Mandelbrot:
54     znew = c
55     if maIntag == kMandel:
56         znew = z*z + c
57     elif maIntag == kJulia:
```

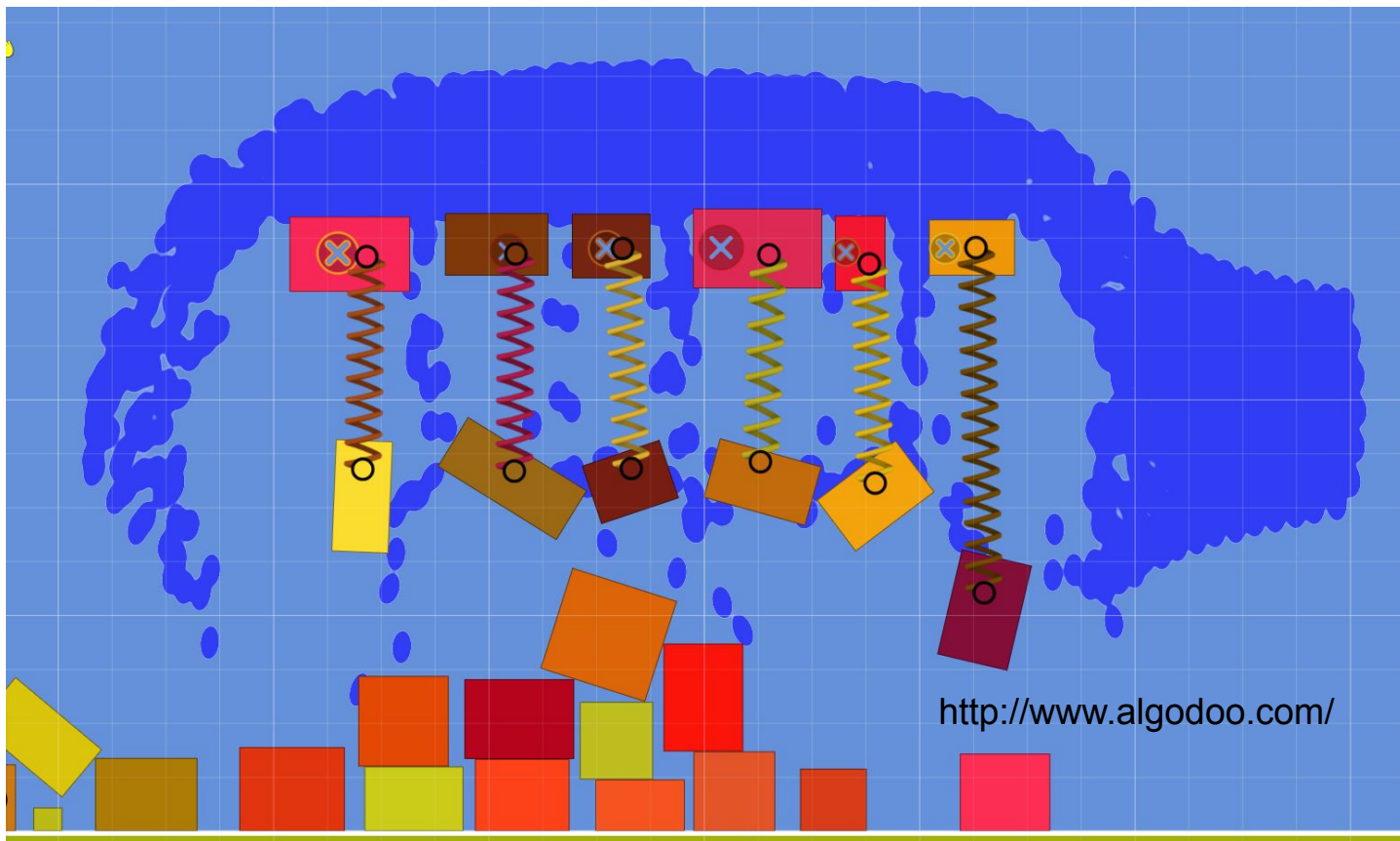
# Model

<http://jointlab.upol.cz/kvita/randomWalk.html>

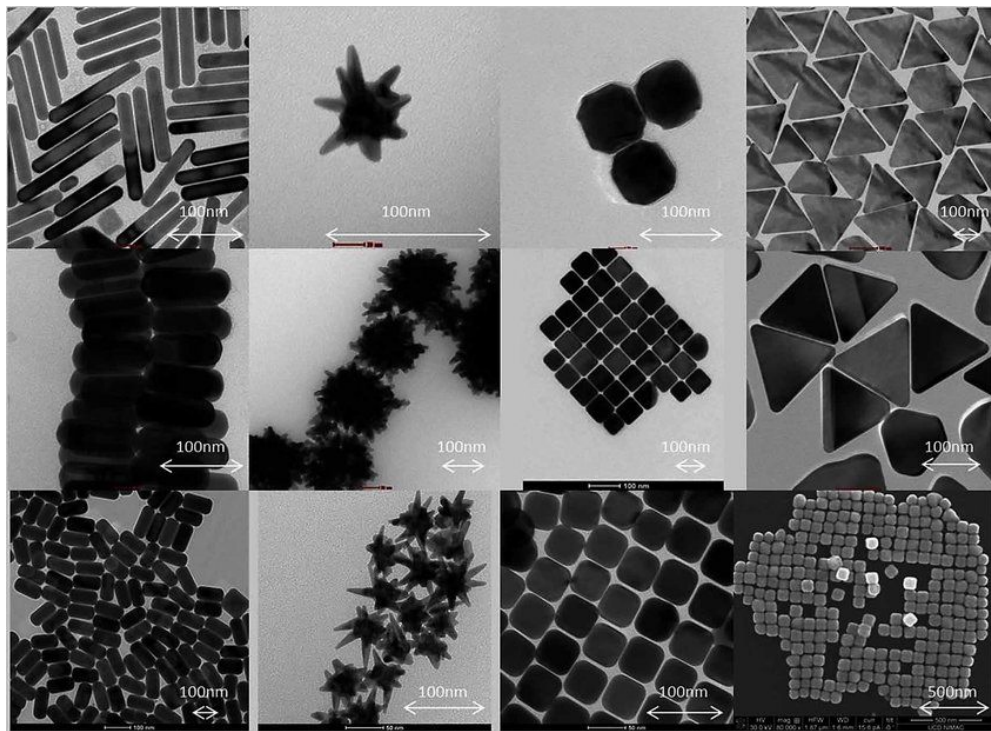


```
46 def DrawSeq(can, seq,
47             col0 = ROOT.kBlack, col1 = ROOT.kRed,
48             #useBinaryCols = False,
49             useBinaryCols = True,
50             smallerDelta = 0.,
51             #smallerDelta = 0.1,
52             #lsz = 2
53 ):
54     can.cd()
55     can.Range(0,0,1,1)
56
57     marks = []
58     root = int(sqrt(len(seq)))
59     ddx = 1. / (root)
60     ddy = ddx
61     x0 = ddx/2.
62     y0 = 1. - ddy/2.
63     nCol = root
64     R = (0.5 - smallerDelta)*(1. - 2*ddx) / root
65     print(R)
66
67
68     for j in range(0,len(seq)):
69         rnd = seq[j]
70         i = 0
71         if rnd >= 0.5:
72             i = 1
73         x = x0 + ddx*(j % nCol)
74         y = y0 - ddy*(j // nCol)
75         #print('{:1.2f},{:1.2f}'.format(x,y))
76         #if j >= nCol*nRow:
77             # break
78         mark = ROOT.TEllipse(x, y, R, R)
79         mark.SetFillStyle(1001)
80         col = col1
```

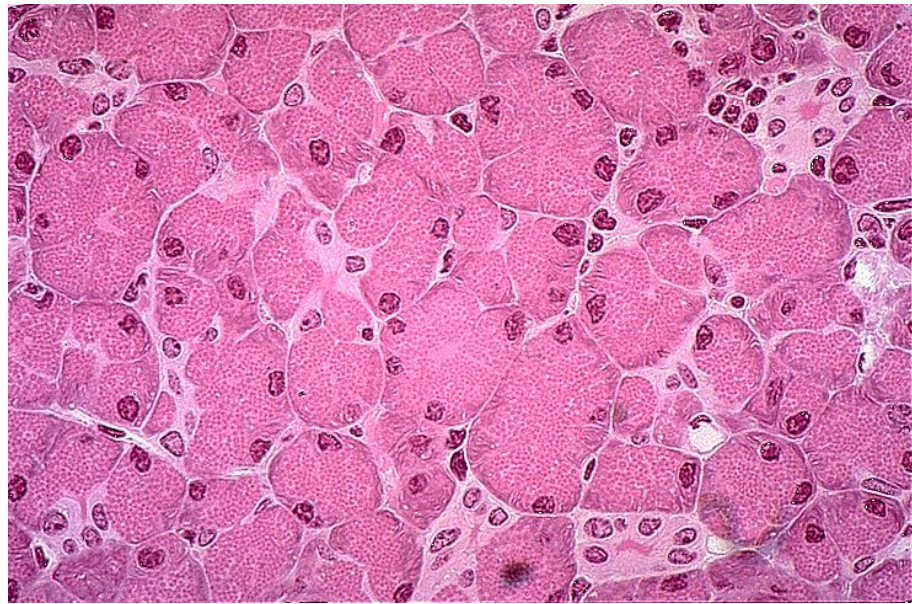
# Model



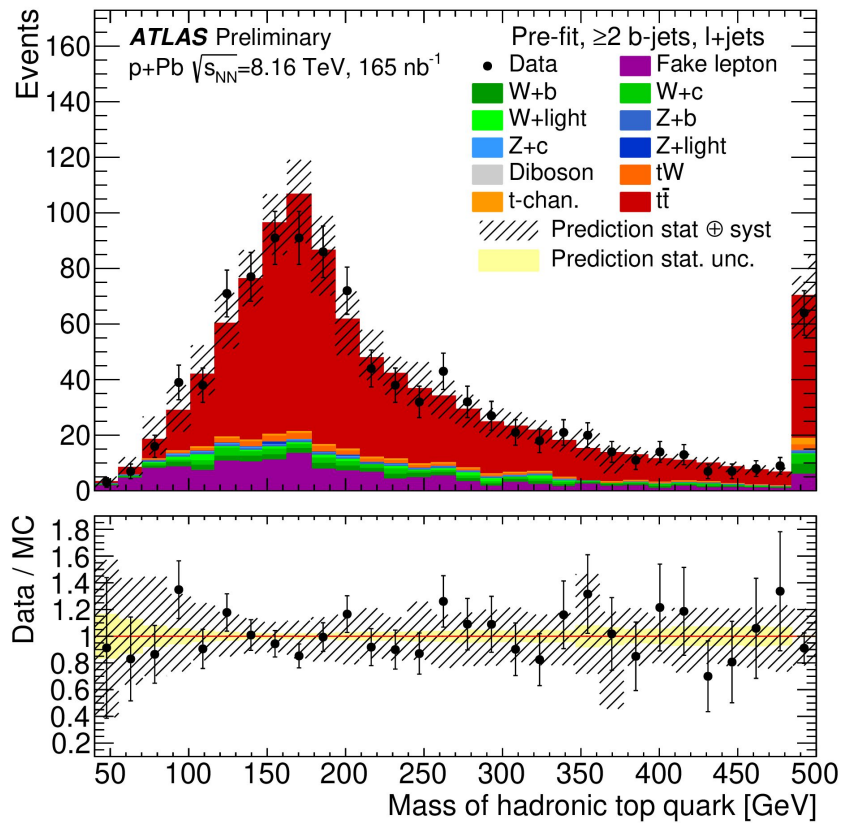
# Vzorek



# Vzorek



# Vzorek



# Vzorek

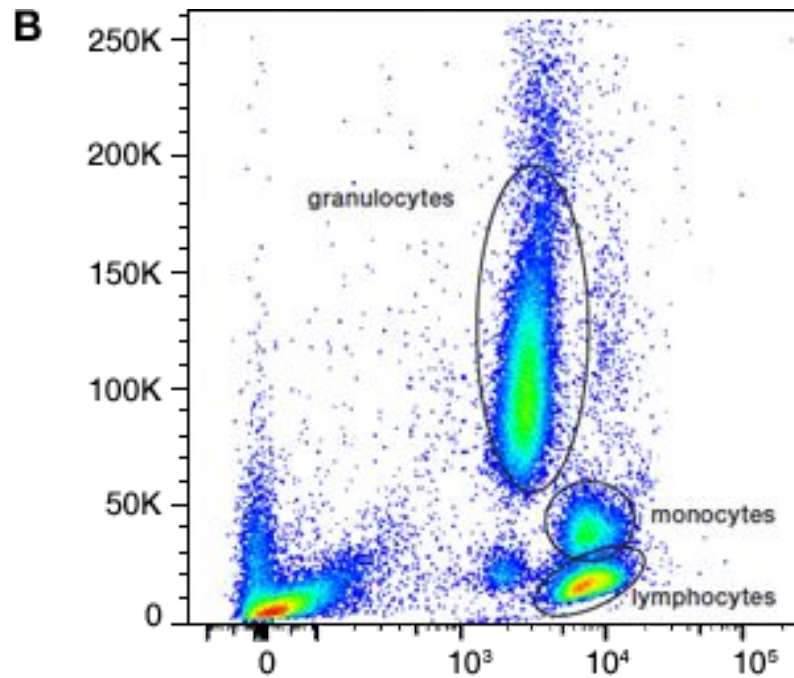




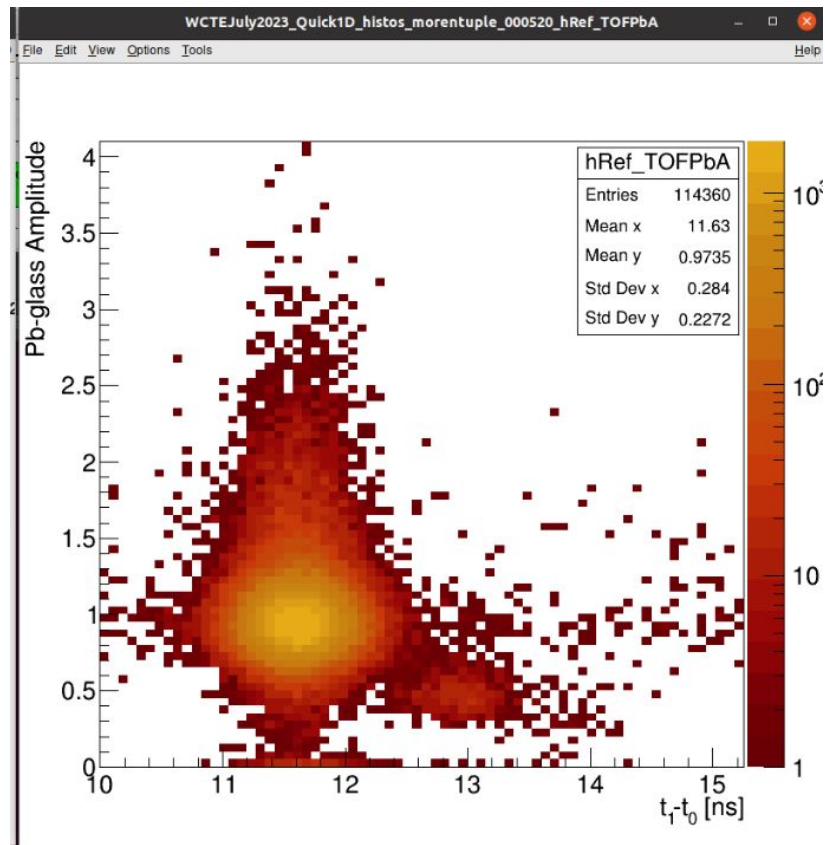
# Populace



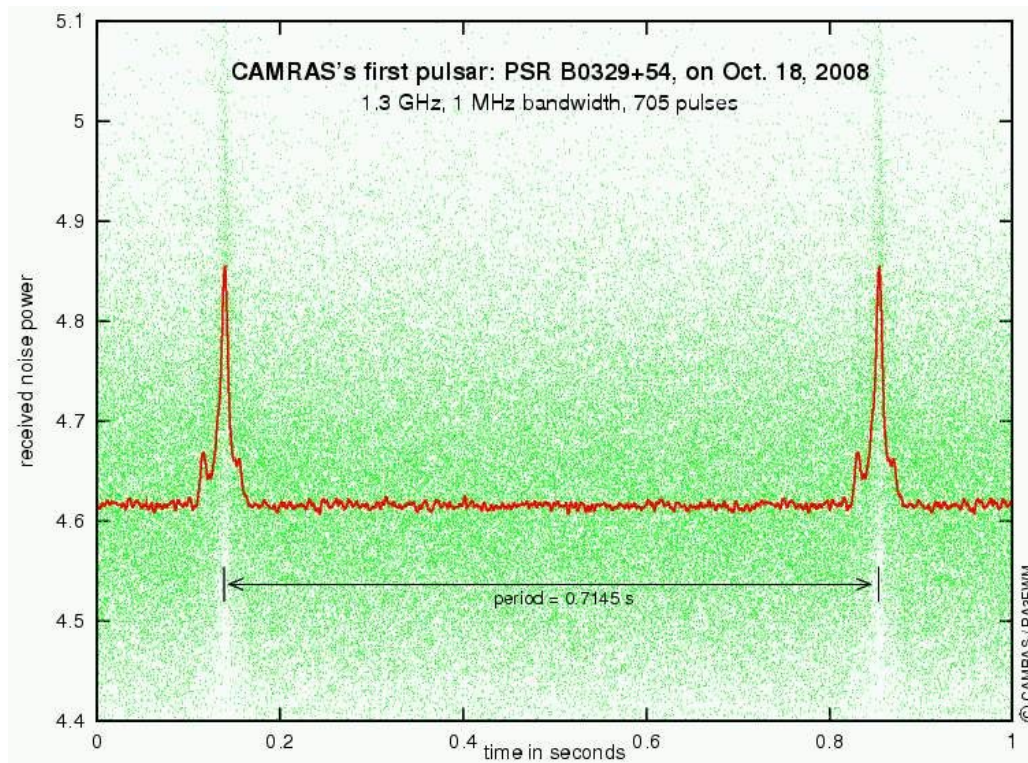
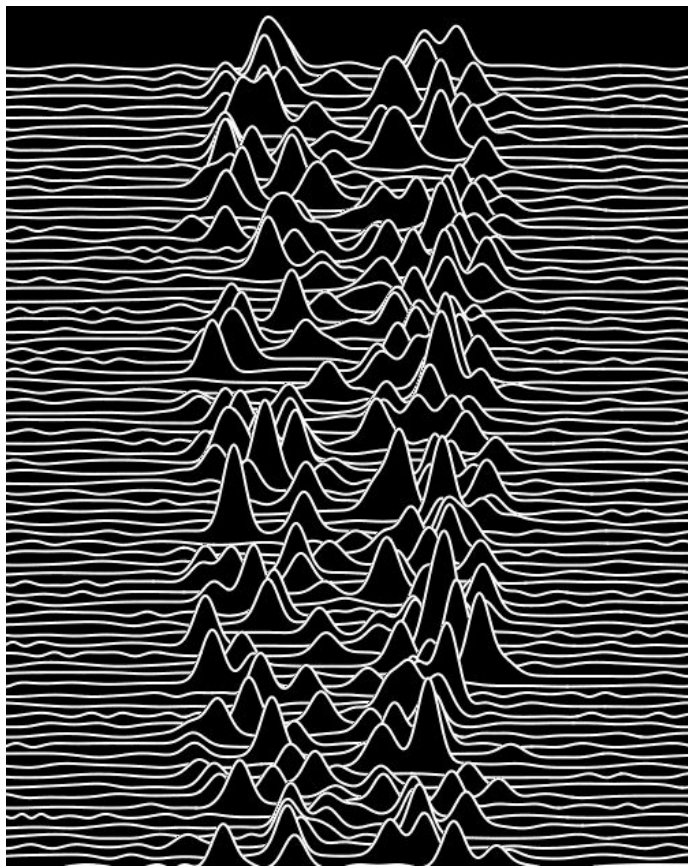
# Populace



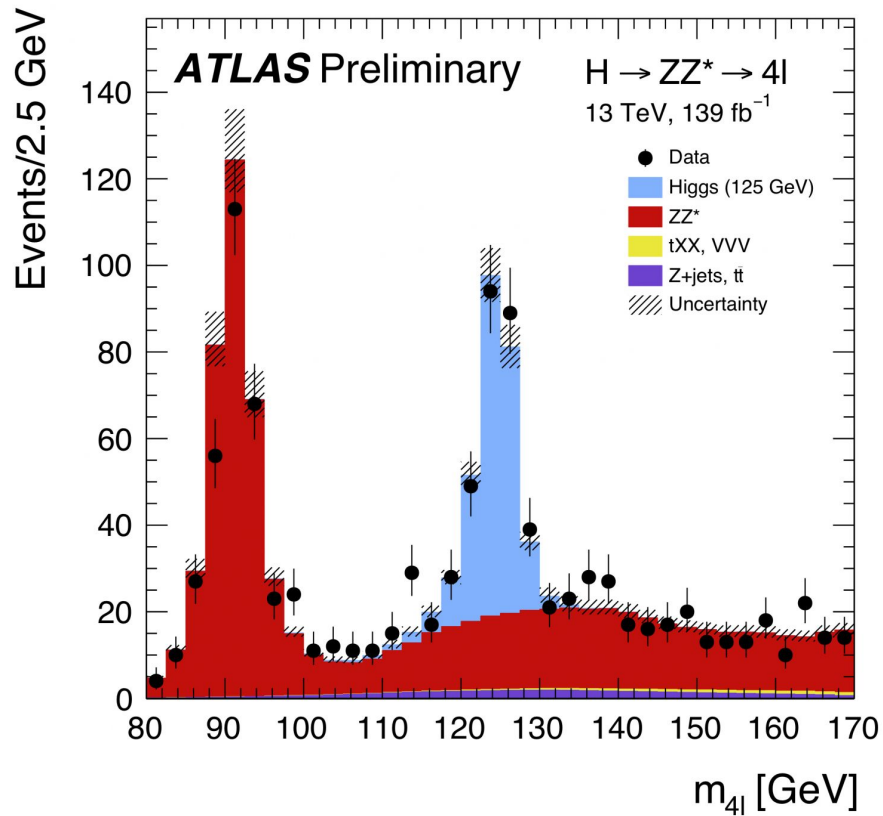
# Populace



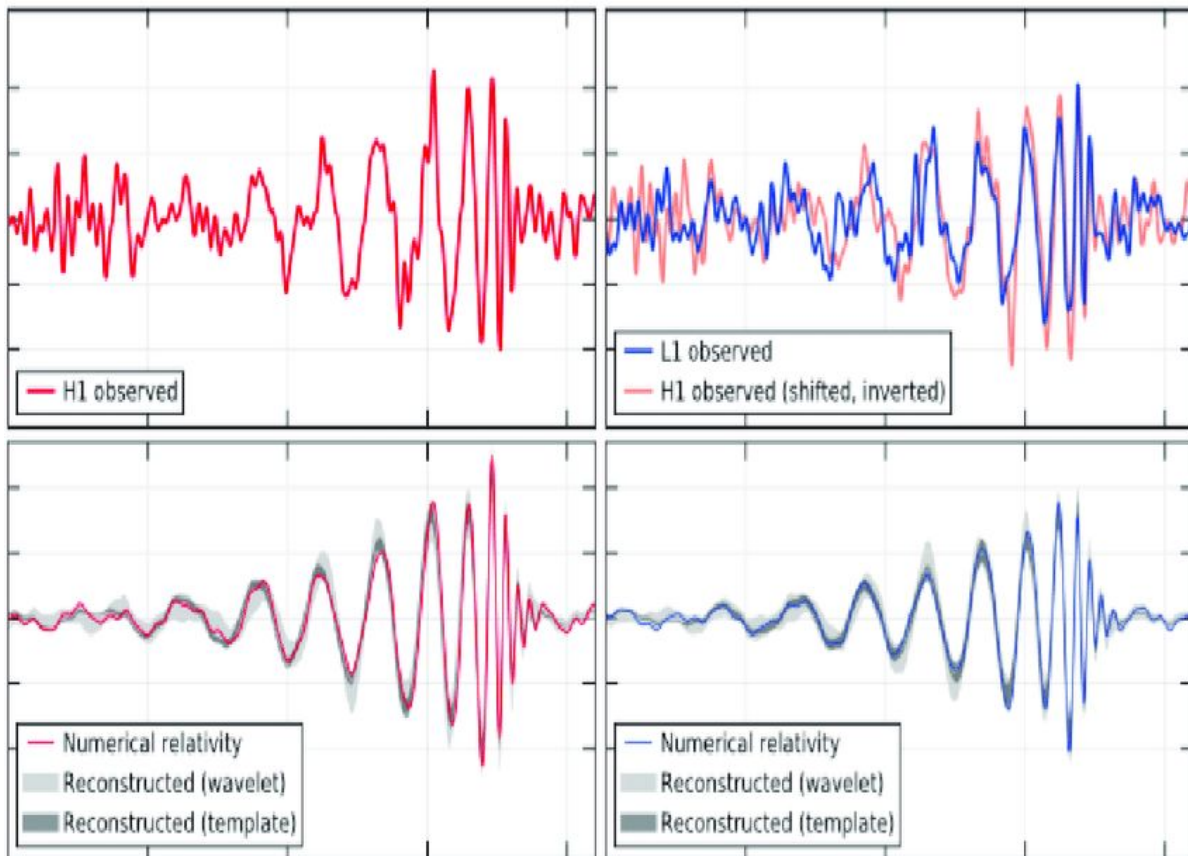
# Signal



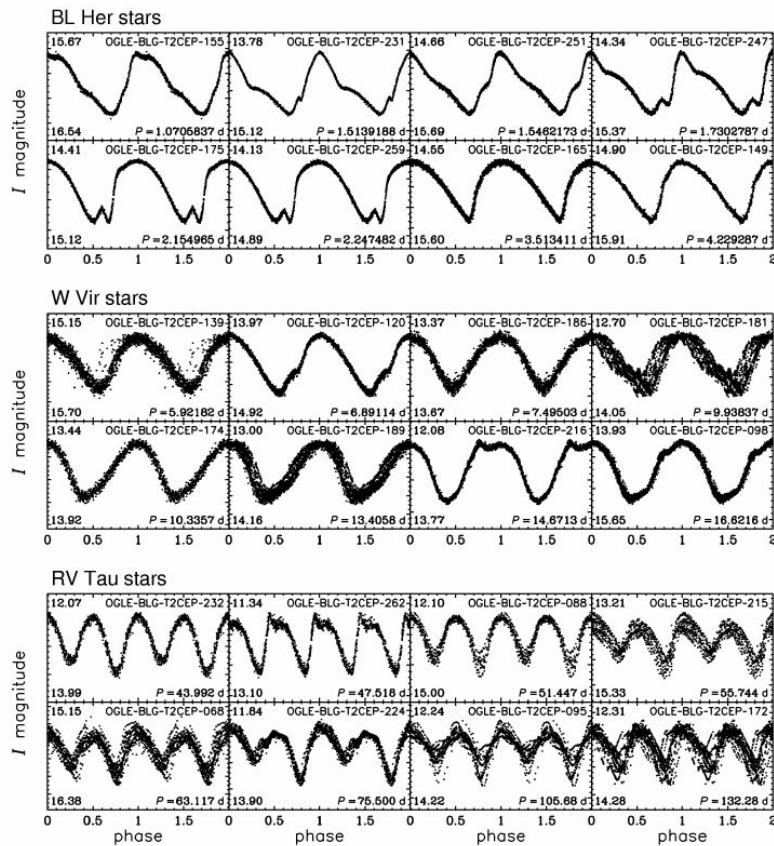
# Signal



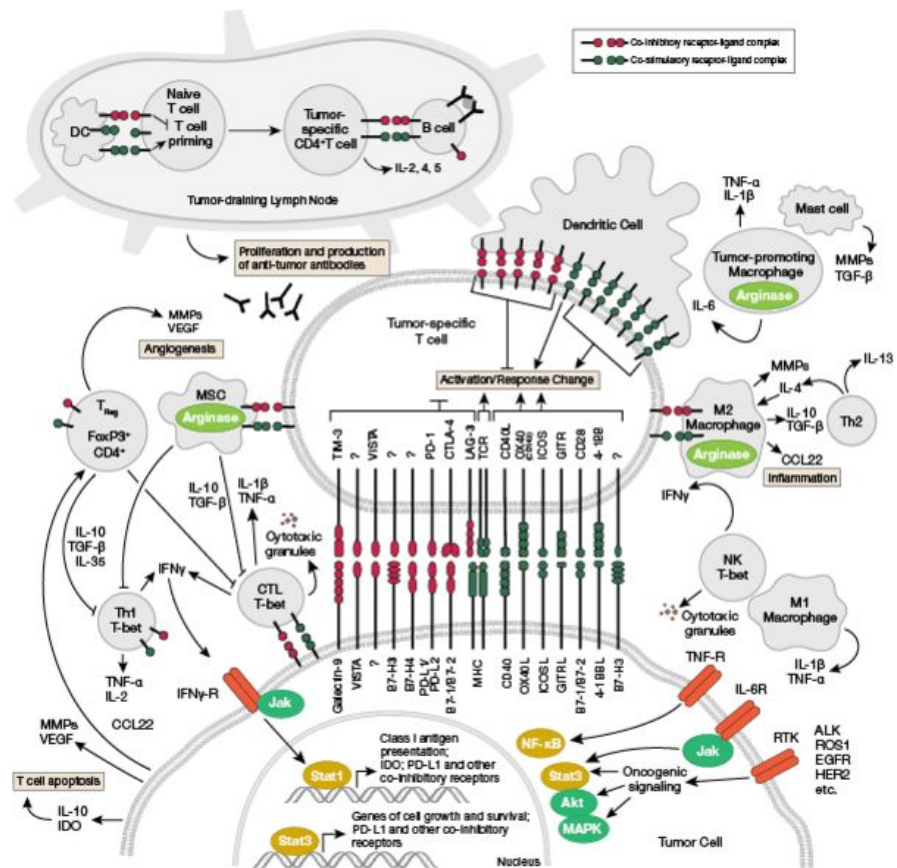
# Signal



# Signal



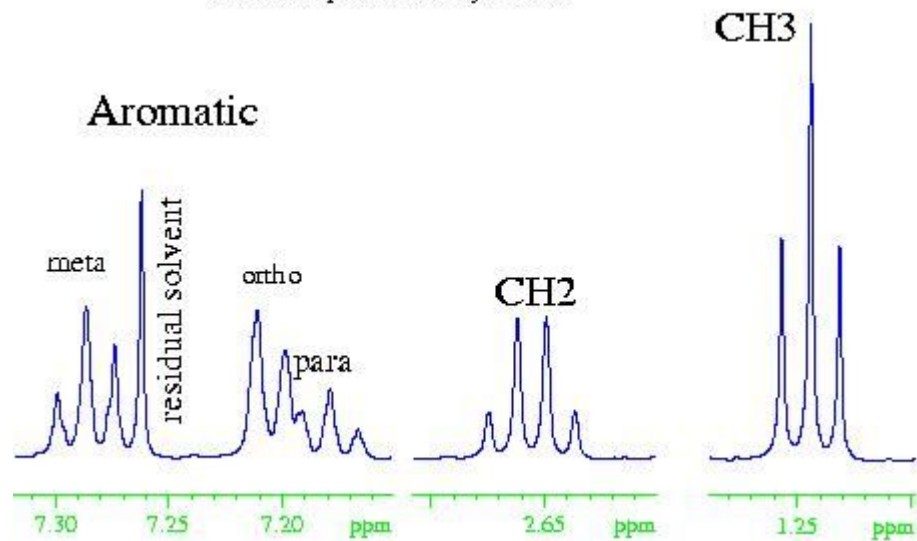
# Signal



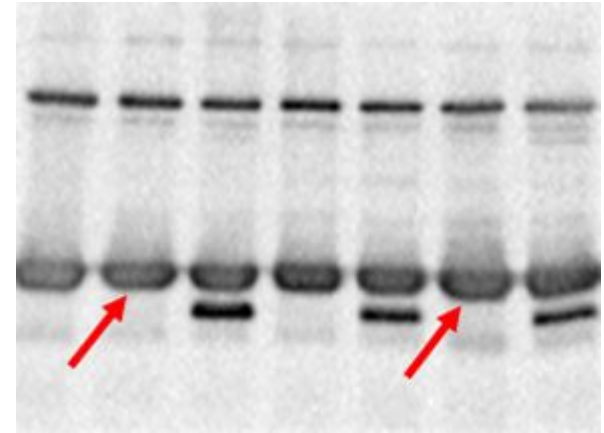
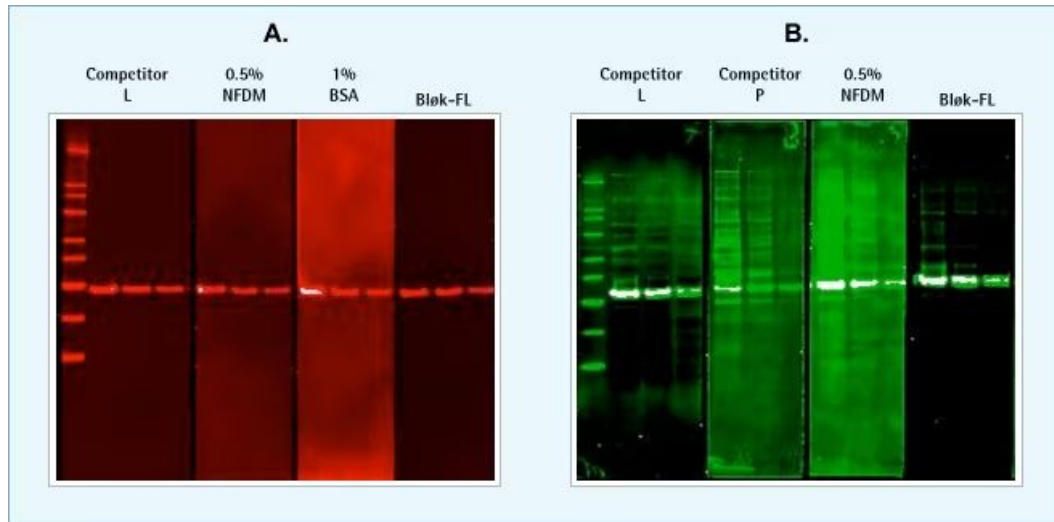


# Signal

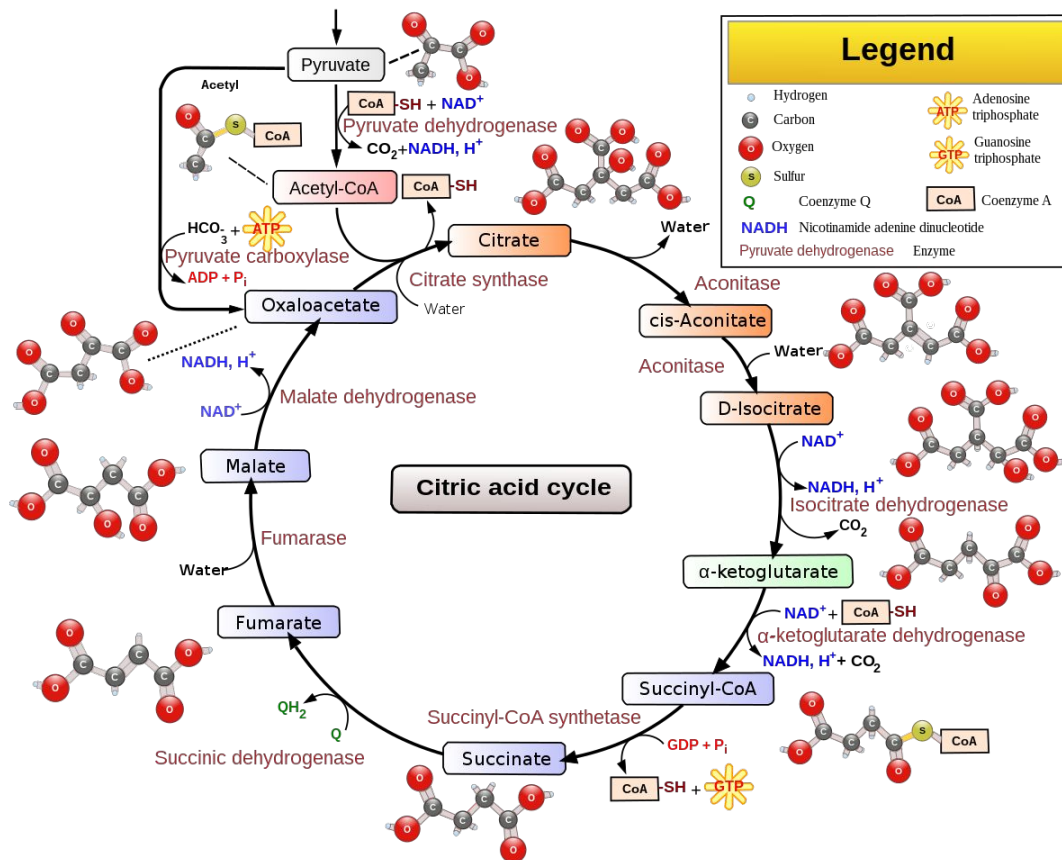
<sup>1</sup>H NMR Spectrum of Ethylbenzene



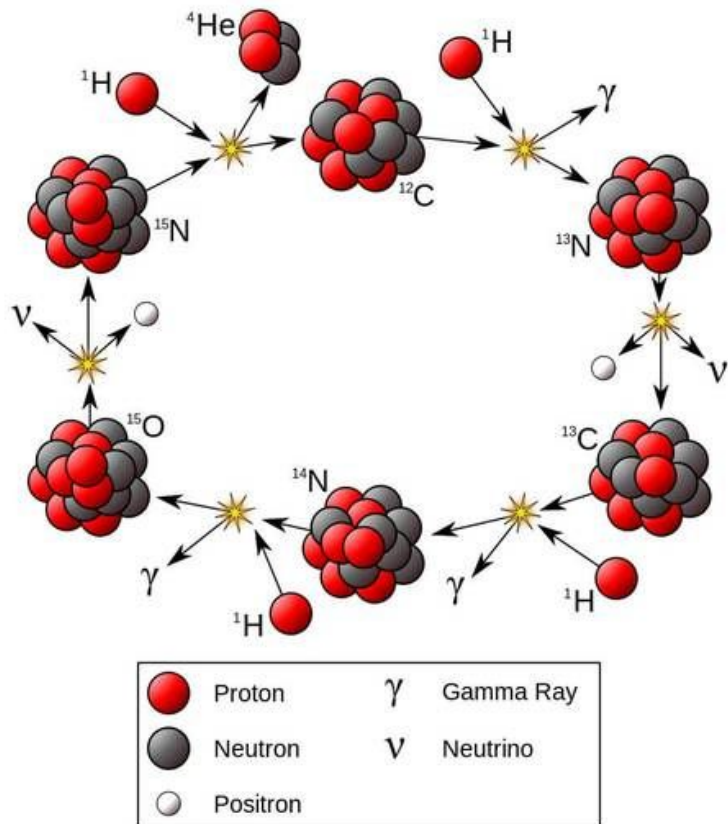
# Signal



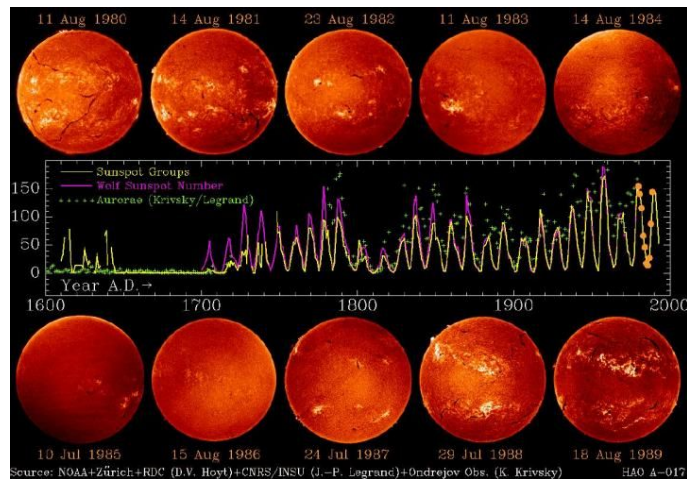
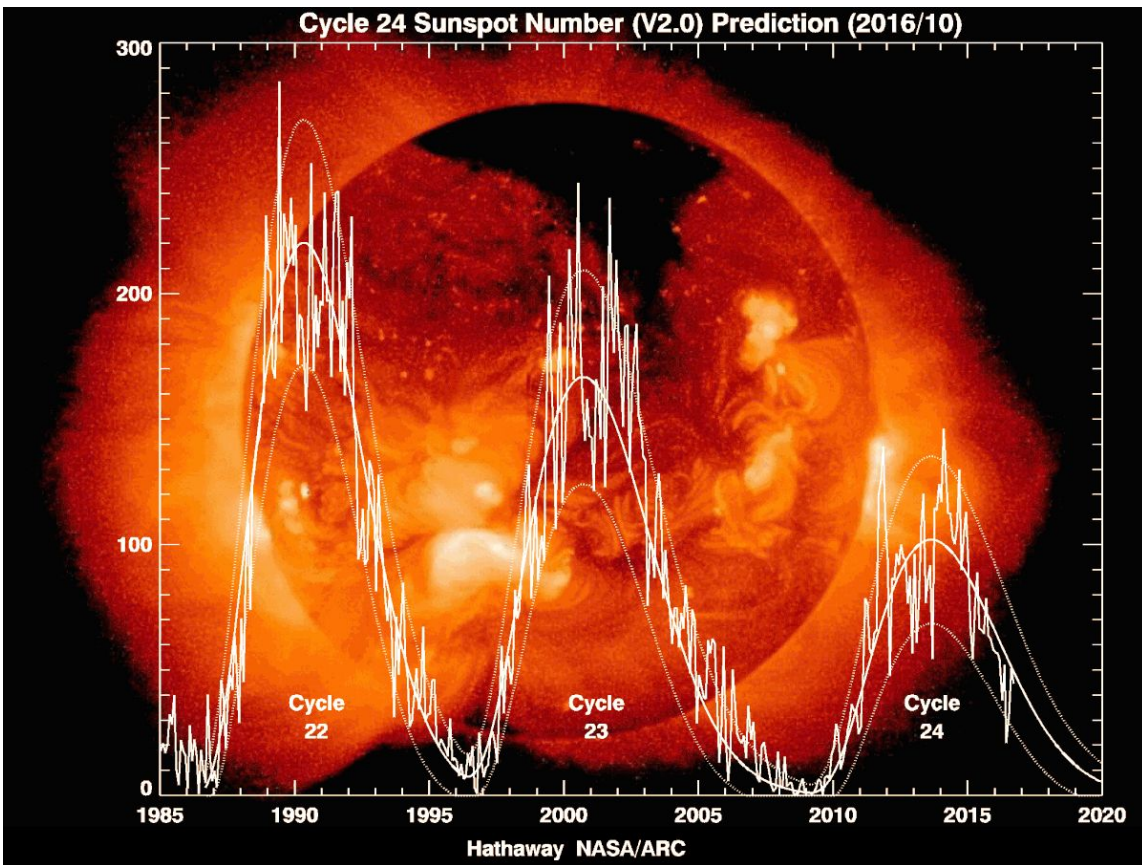
# Cyklus



# Cyklus



# Cyklus



## Muzeum Vysočiny Třebíč

zve Vás a Vaše přátele na tradiční

# JARNÍ CYKLUS PŘEDNÁŠEK v roce 2023



Muzeum Vysočiny Třebíč, Zámek 1, tel. 568 408 890  
[www.muzeumtr.cz](http://www.muzeumtr.cz)

### 7. 3. Lidové kroje v České republice

Mgr. Jan Kůča  
Muzeum Ostrov lidových krojů

### 21. 3. Okolo Ugandy za malými savci

Mgr. Alena Fornůsková, Ph.D.  
Ústav biologie obratlovců AV ČR, v. v. i.

### 4. 4. Do Kamerunu nejen za ptáky

MVDr. Oldřich Tomášek, Ph.D.  
Ústav biologie obratlovců AV ČR, v. v. i.

### 18. 4. Vláda a doba Karla VI. a Marie Terezie

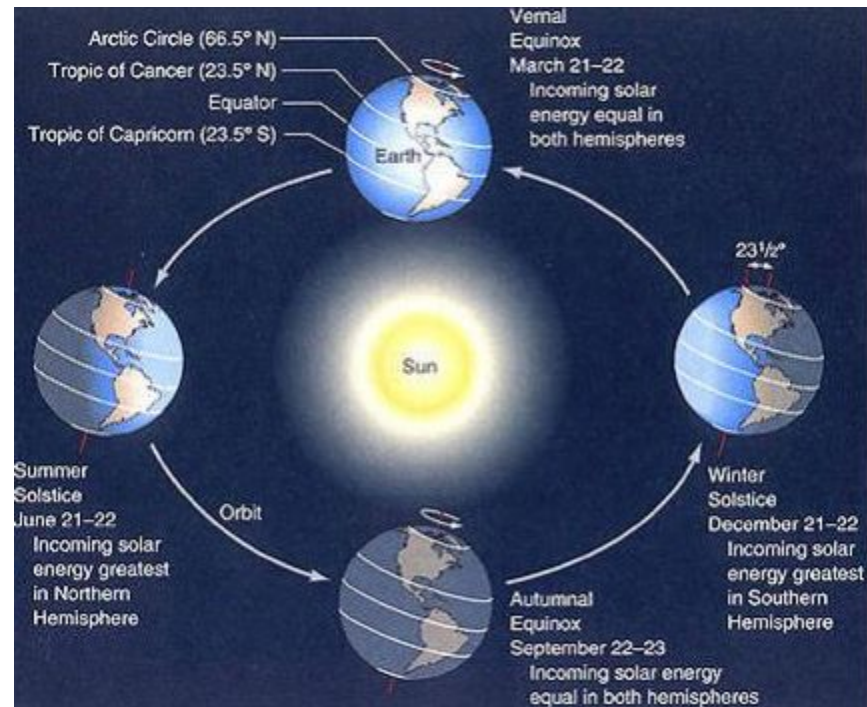
Josef Špidla  
Vyškov

**Přednášky se konají na trebičském zámku.**

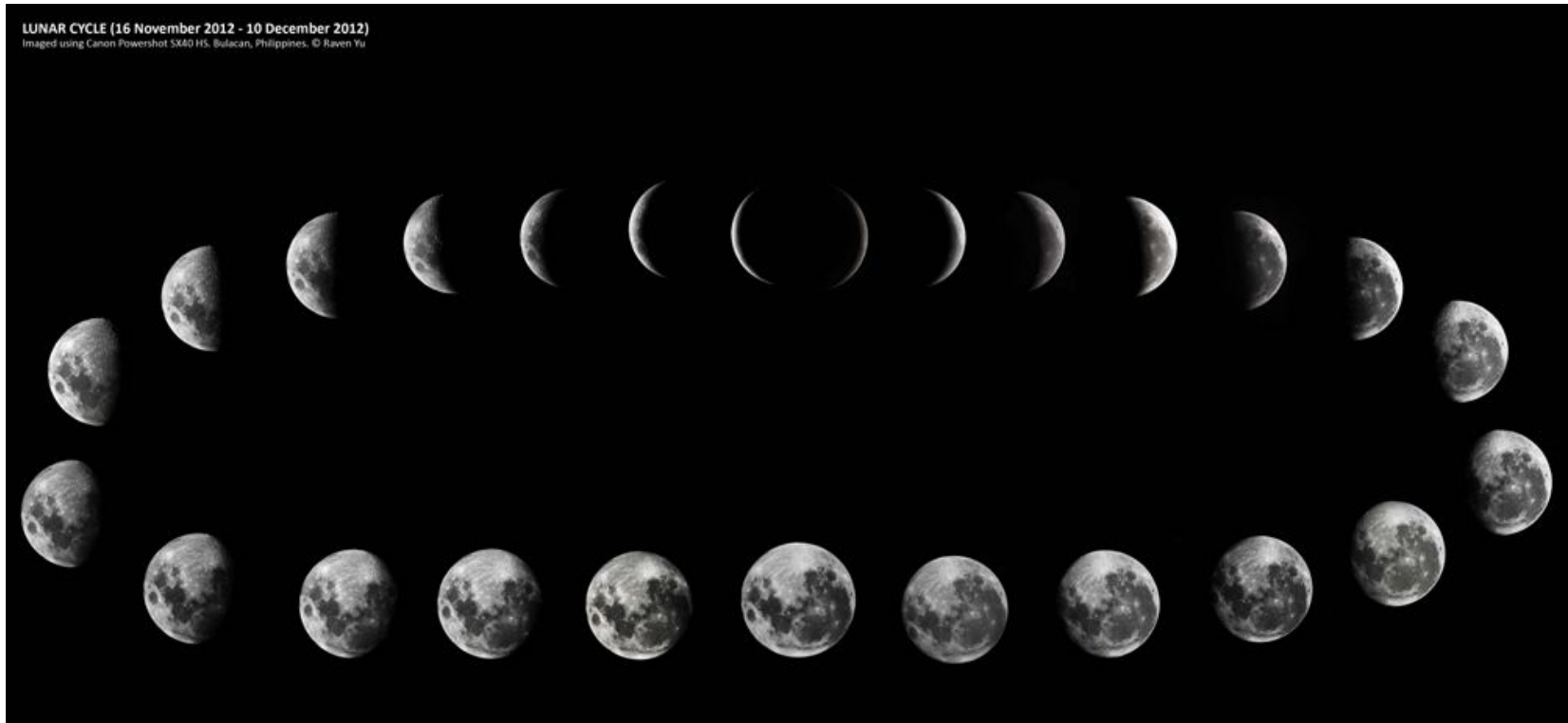
Začátky přednášek každé  
druhé úterý v 17:30 hodin.  
Vstup volný.



# Cyklus



# Cyklus





# Cyklus

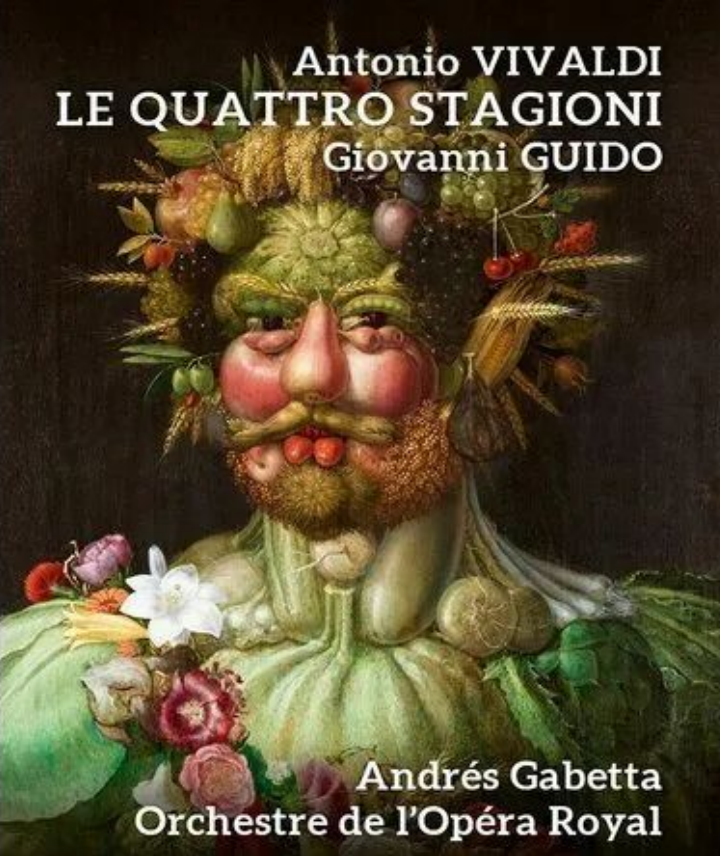
Château de  
**VERSAILLES**  
Spectacles

Cycle  
VENISE-VIVALDI  
VERSAILLES  
N°1

OPÉRA ROYAL  
**250**  
ANS  
1770-2020

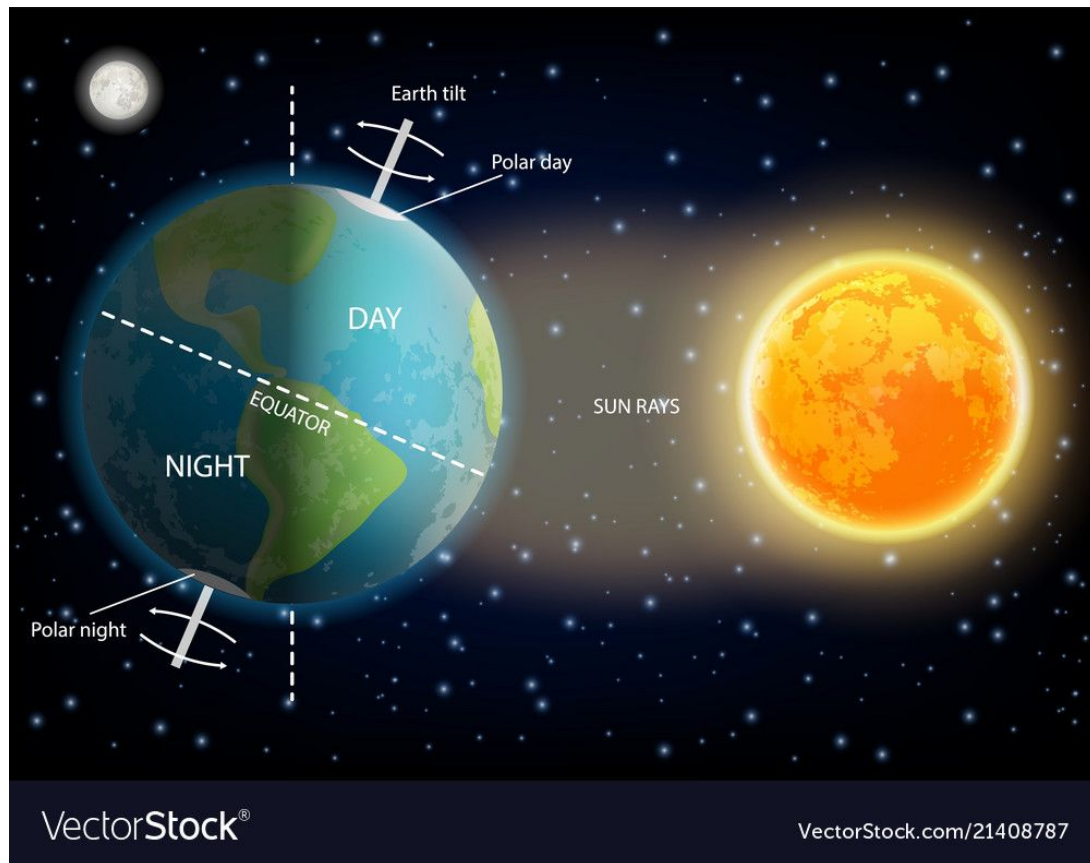
CHATEAU DE VERSAILLES

Antonio VIVALDI  
**LE QUATTRO STAGIONI**  
Giovanni GUIDO

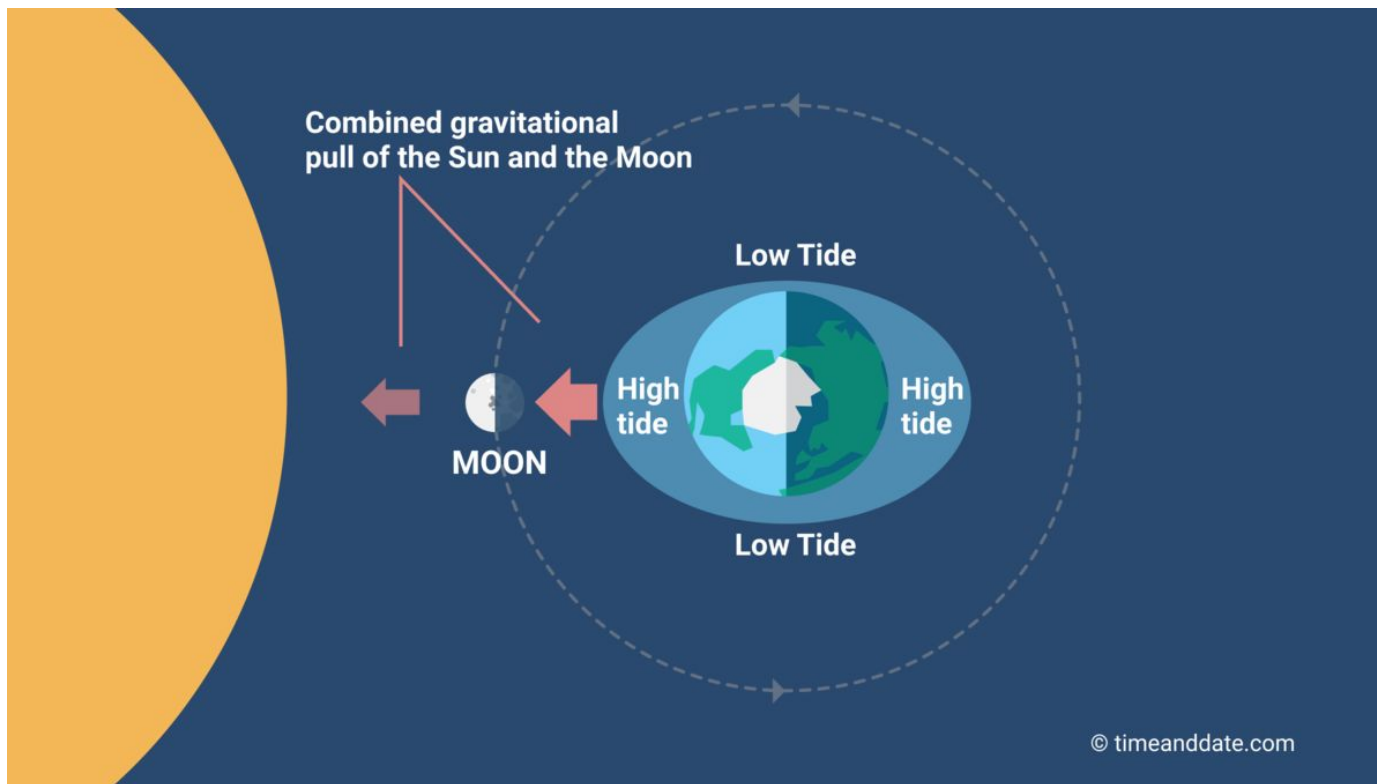


Andrés Gabetta  
Orchestre de l'Opéra Royal

# Cyklus



# Cyklus



# Rozdíly?

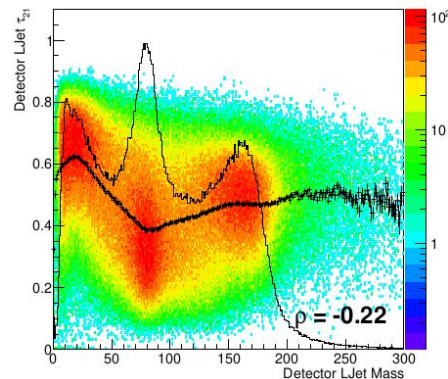
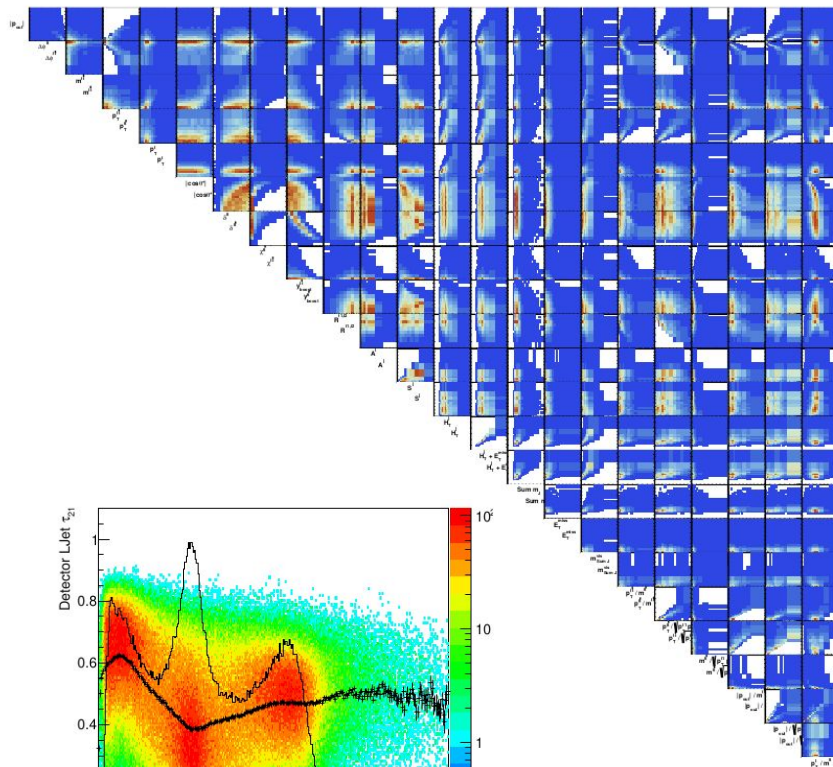
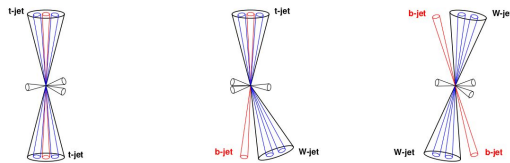
Jiné názvosloví, termíny, významy  
Ale stejné principy!

# Přírodní vědy

- Pozoruj
- Najdi zákon
- Zobecni
- Předpověz
- Ověřuj
- Opakuj

# Přírodní vědy

- Pozoruj
  - experimentuj, měř, zpracovávej c
- Najdi zákon
  - interpretuj, publikuj
- Zobecni
  - uč se, teoretizuj
- Předpověz
- Ověřuj
  - experimentuj
- Opakuj
  - publikuj



$$-\ln \mathcal{L}(\hat{\theta} \pm \sigma_{\theta}) \approx -\ln \mathcal{L}(\hat{\theta}) + \frac{1}{2\hat{\sigma}_{\theta}^2}(\hat{\theta} \pm \hat{\sigma}_{\theta} - \hat{\theta})^2 = -\ln \mathcal{L}(\hat{\theta}) + \frac{1}{2}$$