DERIVATIVE FREE OPTIMIZATION   CLASS 5   2025/2026

EXERCICE : RUNNING & UNDERSTANDING CMA-ES

In [2]:
```
pip install cma
```

In [6]:
```
from __future__ import division
import numpy as np
from matplotlib import pyplot as plt
import cma
```

fevals

In [8]:
```
res = cma.fmin2(cma.ff.elli, np.ones(10), 1e-0)
cma.plot(xsemilog=0,plot_mean=1)
```

↳ run CMA-ES on $f_{elli}(x) = \sum_{i=1}^{n} \left(10^6\right)^{\frac{i-1}{n-1}} x_i^2$

in dim $n = 10$, with initial $m_0 = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$, $\sigma_0 = 1$.

```
(5_w,10)-aCMA-ES (mu_w=3.2,w_1=45%) in dimension 10 (seed=1076219, Fri Jan  9 12:17:55 2026)
Iterat #Fevals   function value  axis ratio  sigma  min&max std  t[m:s]
    1     10 2.434067915396967e+05 1.0e+00 9.75e-01  9e-01  1e+00 0:00.0
    2     20 2.689896144368289e+05 1.2e+00 9.14e-01  8e-01  9e-01 0:00.0
    3     30 1.805662878443460e+05 1.3e+00 8.64e-01  8e-01  9e-01 0:00.0
  100   1000 1.420300222847435e+02 3.1e+01 8.43e-02  6e-03  2e-01 0:00.1
  200   2000 1.769958247849342e+00 1.8e+02 1.30e-02  2e-04  3e-02 0:00.2
  300   3000 6.427652064259963e-04 8.3e+02 2.66e-03  1e-05  1e-02 0:00.4
  400   4000 5.580420639014203e-11 1.0e+03 1.32e-06  2e-09  2e-06 0:00.5
  451   4510 3.166390616617839e-14 1.0e+03 5.84e-08  7e-11  7e-08 0:00.5
termination on {'tolfun': 1e-11} (Fri Jan  9 12:17:56 2026)
final/bestever f-value = 1.631534e-14 1.631534e-14 after 4511/4511 evaluations
incumbent solution: [-3.52353191e-08 -7.58103766e-09  2.59719521e-09  6.21442969e-09
  2.20001196e-10  1.09270517e-10 -1.67589778e-10 -1.26363661e-10 ...]
std deviations: [6.67937818e-08 3.37478250e-08 1.68194076e-08 7.18462019e-09
 3.19613494e-09 1.45321955e-09 6.37733349e-10 3.46904695e-10 ...]
```

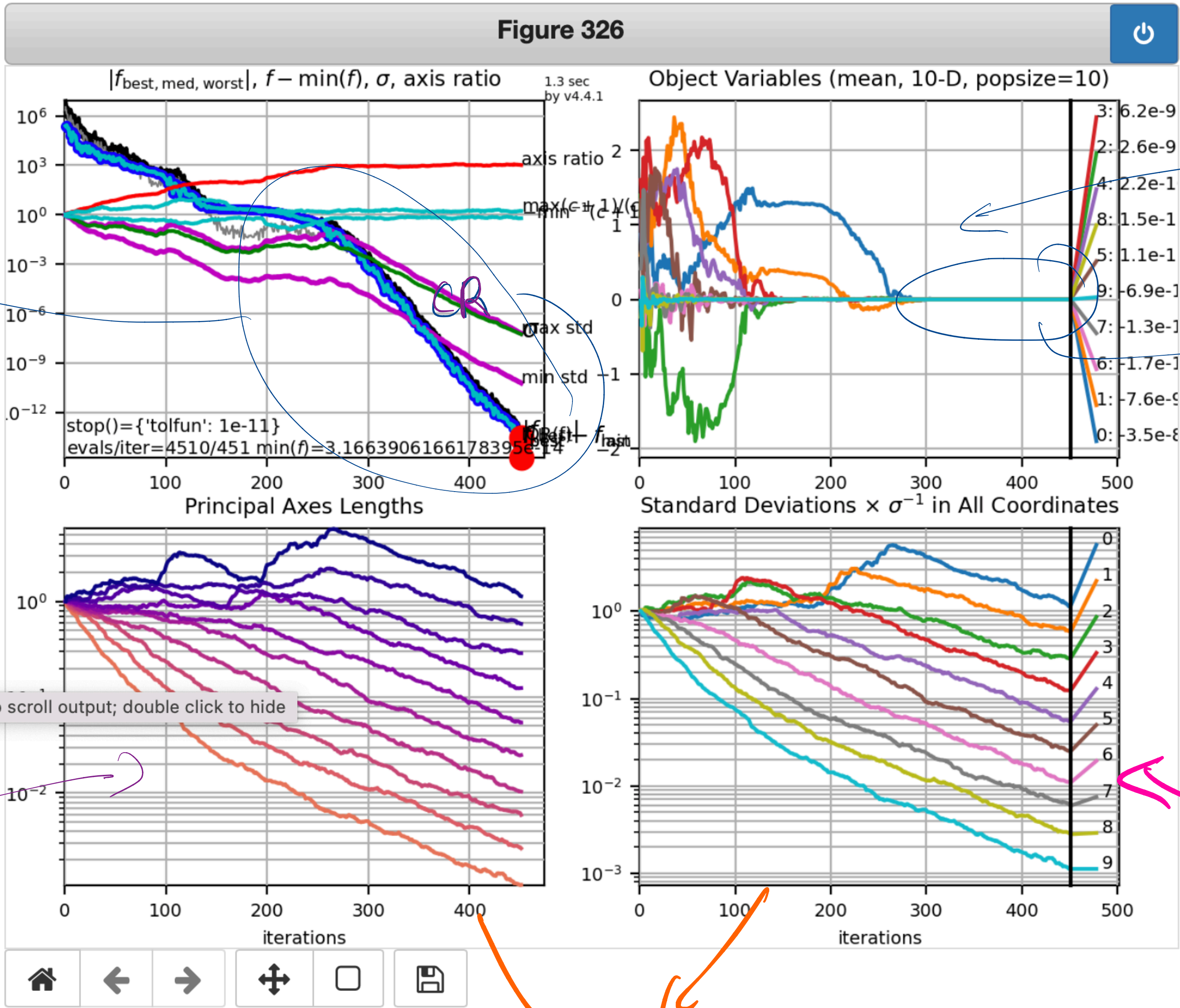
Plot associated to the run on felli

We observe linear convergence

$\frac{1}{t} \ln \|m_t - m^*\|$

$\downarrow$

$-CR$



Display the trajectory followed by the mean vector.

Display 10 curves,

close to zero

Better display on next slide

$C_t =$

$\sqrt{\ }$

Display square root of diagonal values

Display $\sqrt{\text{eigenvalues}}$

Out [8]: `<cma.logger.CMADataLogger at 0x7fde38b6e4d0>`

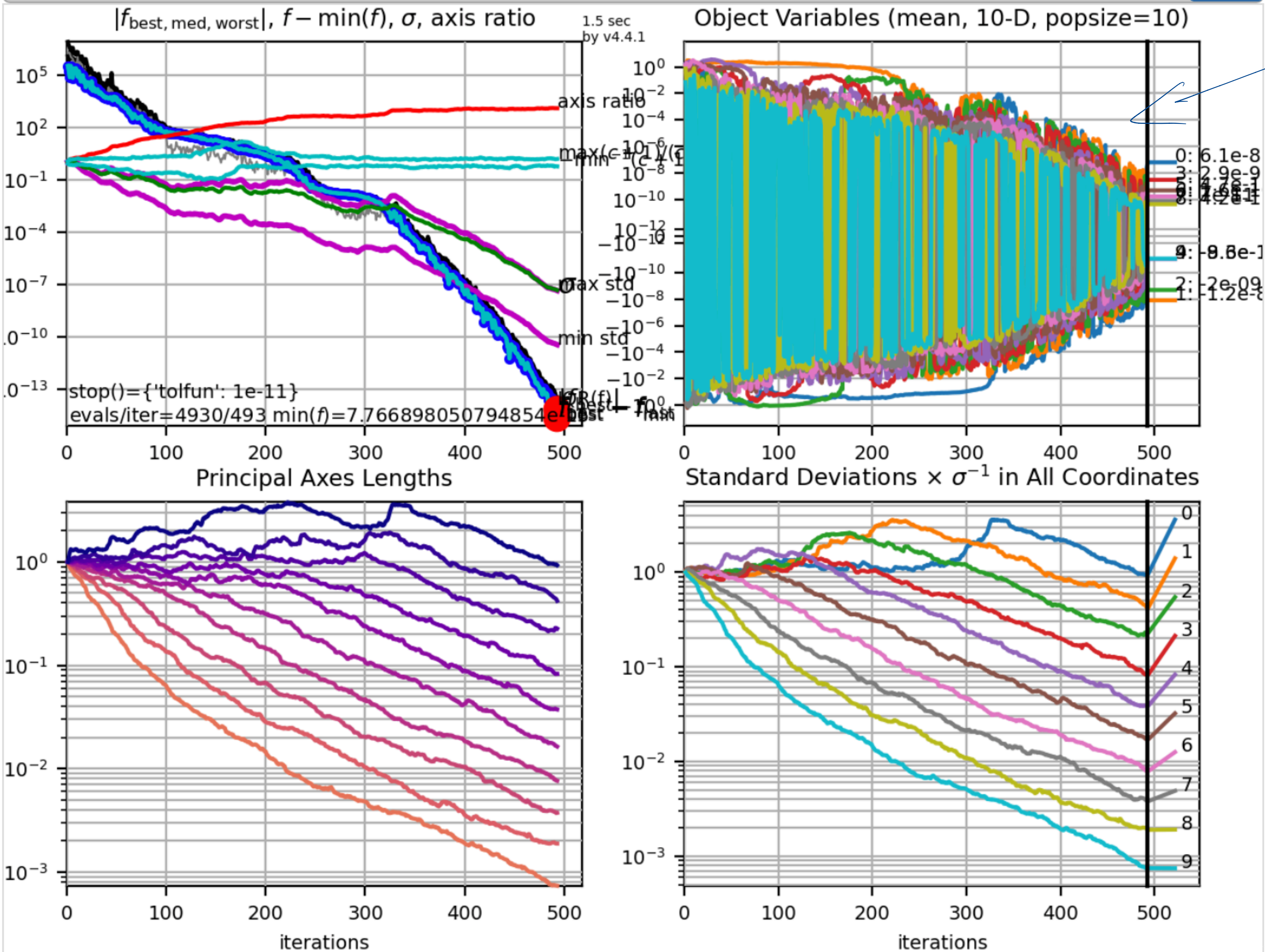
Ct is close to diagonal

In [9]:

```
res = cma.fmin2(cma.ff.elli, np.ones(10), 1e-0)
cma.plot(xsemilog=1,plot_mean=1)
```

1.80880847e-09 7.99007576e-10 3.70950878e-10 1.79498800e-10 ...]

Figure 327



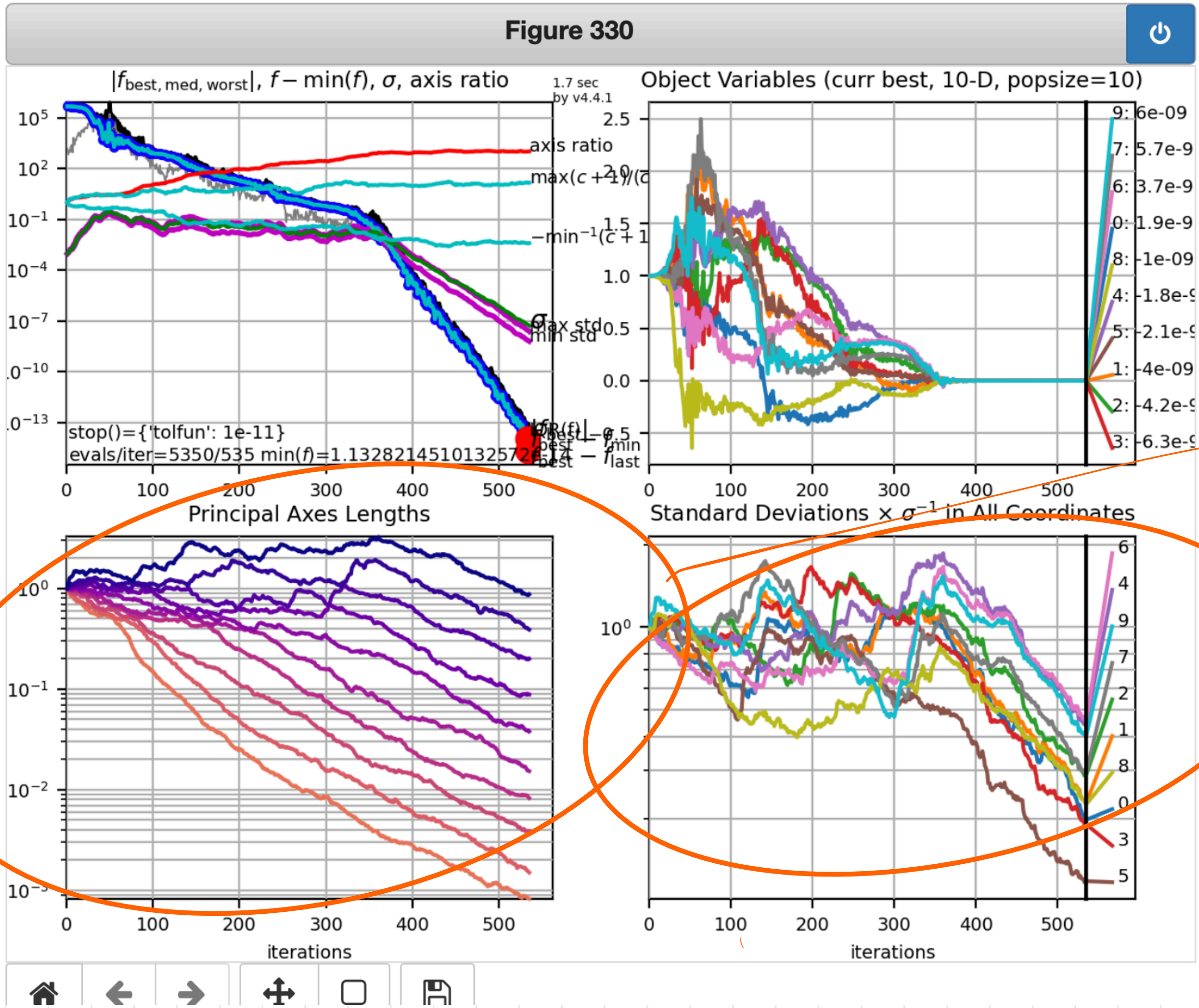
Better visualization

We observe that the mean oscillate around the optimum

Fix a rotation matrix R.

$$f_{ellirot}(x) = f_{elli}(Rx)$$

In [12]:
```
res = cma.fmin(cma.ff.ellirot, np.ones(10), 1e-3)
cma.plot()
```



Look different the covariance matrix is not close to diagonal.

We have observed on the plots that on felli the covariance matrix is close to a diagonal matrix,

on fellirot it is "far" from a diagonal matrix.

On $f_{elli}(x) = \sum_{i=1}^{n} (10^6)^{\frac{i-1}{n-1}} x_i^2$

$$= x^T \begin{pmatrix} 1 & & (0) \\ & (10^6)^{\frac{i-1}{n-1}} & \\ (0) & & 10^6 \end{pmatrix} x = \frac{1}{2} x^T H_{elli} \, x$$

$$H_{elli} = 2 \begin{pmatrix} 1 & & (0) \\ & \ddots & \\ (0) & & 10^6 \end{pmatrix}$$

$\longrightarrow$ Hessian matrix is diagonal

↳ Here if $C \propto H^{-1}$, the covariance should be diagonal

$$f_{\text{ellirot}}(x) = f_{\text{elli}}(Rx)$$

$$= \frac{1}{2}(Rx)^T H_{\text{elli}} Rx$$

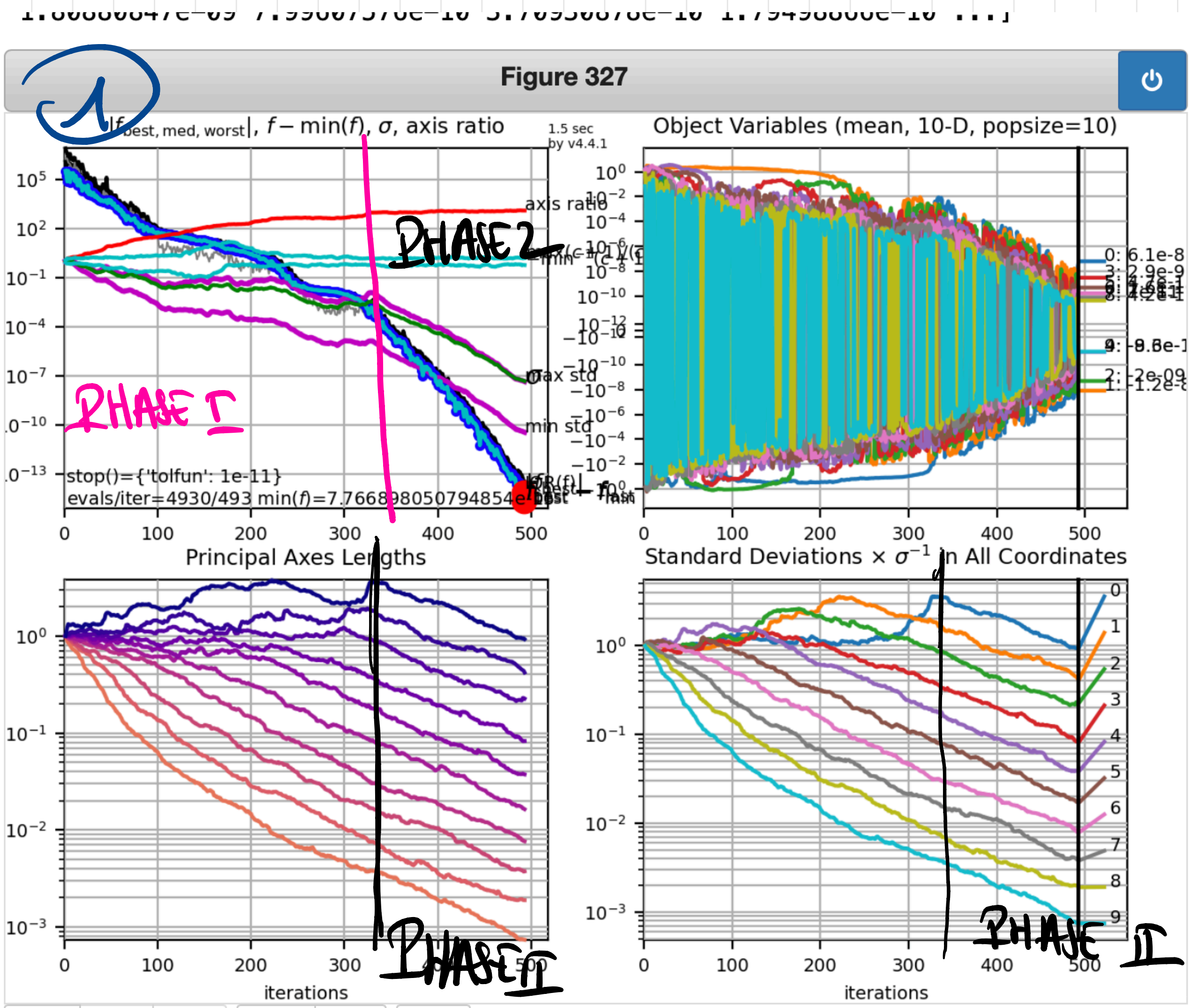
$$= \frac{1}{2} x^T R^T H_{\text{elli}} R x$$

$$H_{f_{\text{ellirot}}} = R^T \left(2 \begin{pmatrix} 1 & & (0) \\ & \ddots & \\ (0) & & 10^6 \end{pmatrix} \right) R$$

↳ Hessian matrix is not diagonal

↳ Here if $C \propto H^{-1}$, the covariance matrix should not be diagonal.

We observe two phases on the convergence plot 1



In Phase II the algorithm has learned H^{-1}, and then converges faster, at the convergence rate observed on the sphere function

In Phase I the algorithm is learning the inverse Hessian.

In Phase II, $C_t \simeq \alpha_t H^{-1}$, the matrix

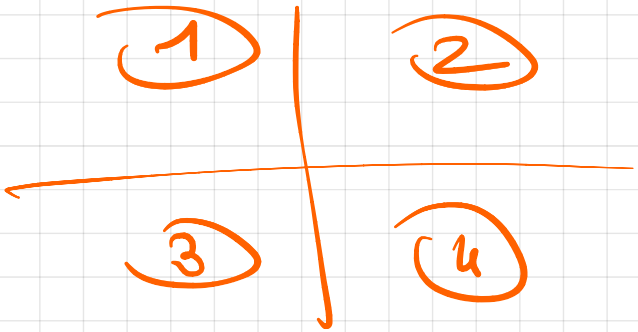
t iteration

$\alpha_t \xrightarrow[t\to+\infty]{} 0$

"goes" to zero, in a way proportional to H^{-1}.

For the exam, you should be able from the plot to find out about the function. Let's say I give you different convex-quadratic function, you should be able to associate the display, to the function.

The display is cut in 4 parts



a) If you compare 3 and 4, it tells you (assume a convex-quadratic function) if C_t is diagonal or not (in the final phase) and since $C_t \propto H^{-1}$, it tells us if the problem has a diagonal Hessian (thus if the problem is separable).

b) by looking at 3, you can deduce the eigenvalues of the Hessian matrix.