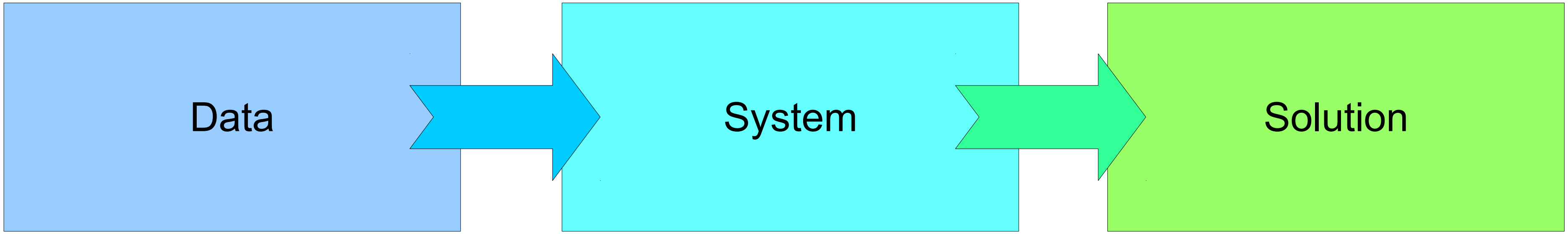
Inversion of ill-conditionned systems

DDM@LMT
2017-2018

What is an ill-conditionned system ?

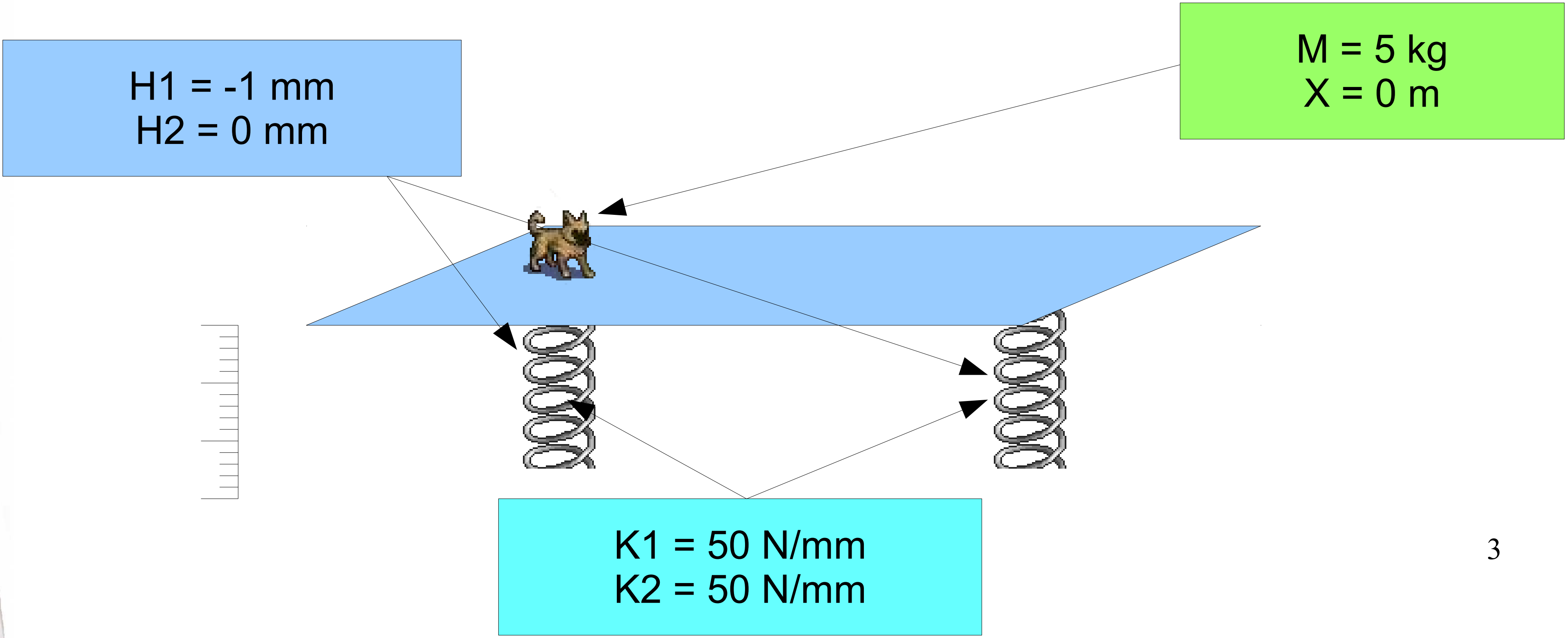
What is a well-conditionned system ?



Data

System

Solution

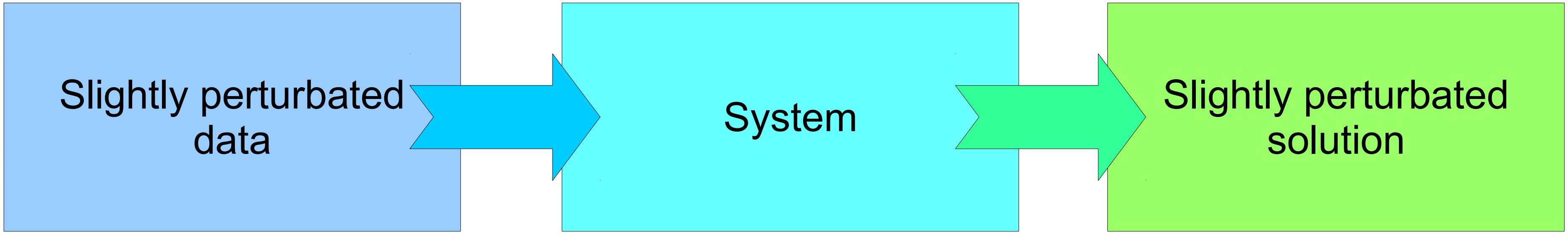


H1 = -1 mm
H2 = 0 mm

M = 5 kg
X = 0 m

K1 = 50 N/mm
K2 = 50 N/mm

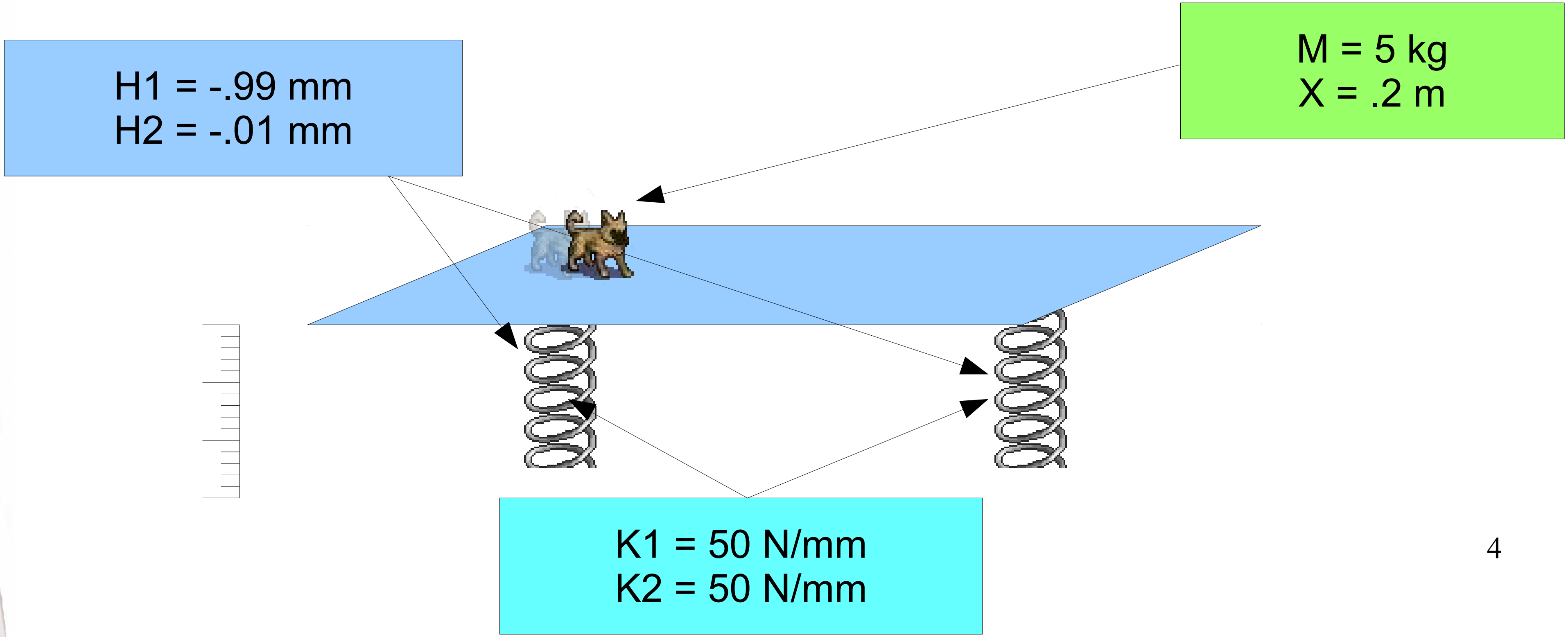
What is a well-conditionned system ?



Slightly perturbated data

System

Slightly perturbated solution

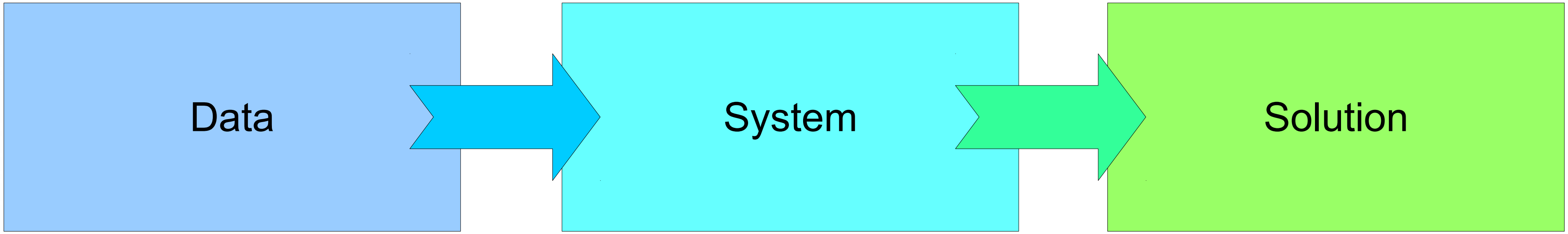


H1 = -.99 mm
H2 = -.01 mm

M = 5 kg
X = .2 m

K1 = 50 N/mm
K2 = 50 N/mm

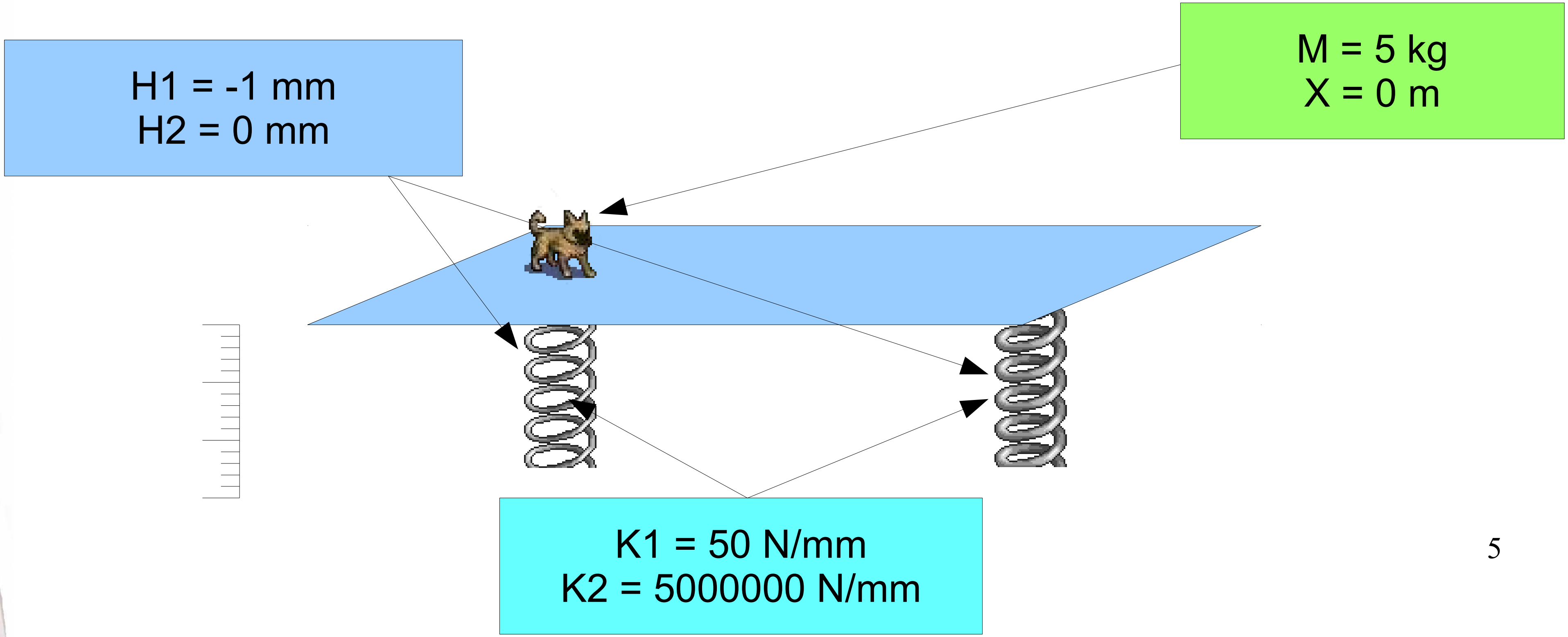
What is an ill-conditionned system ?



Data

System

Solution

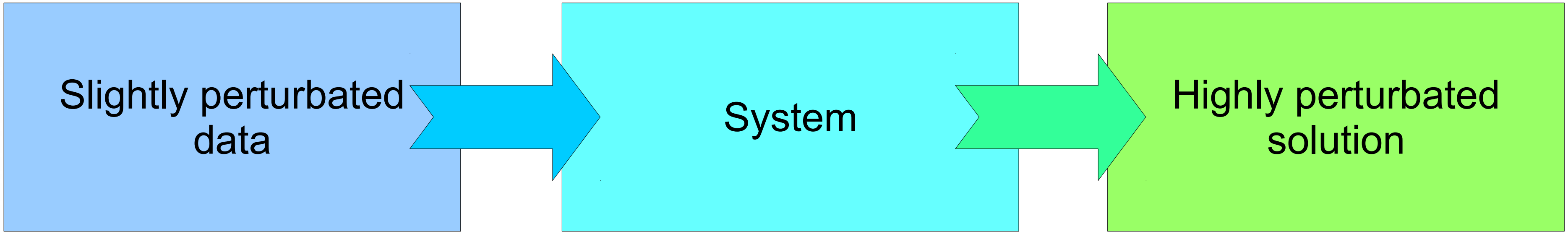


H1 = -1 mm
H2 = 0 mm

M = 5 kg
X = 0 m

K1 = 50 N/mm
K2 = 5000000 N/mm

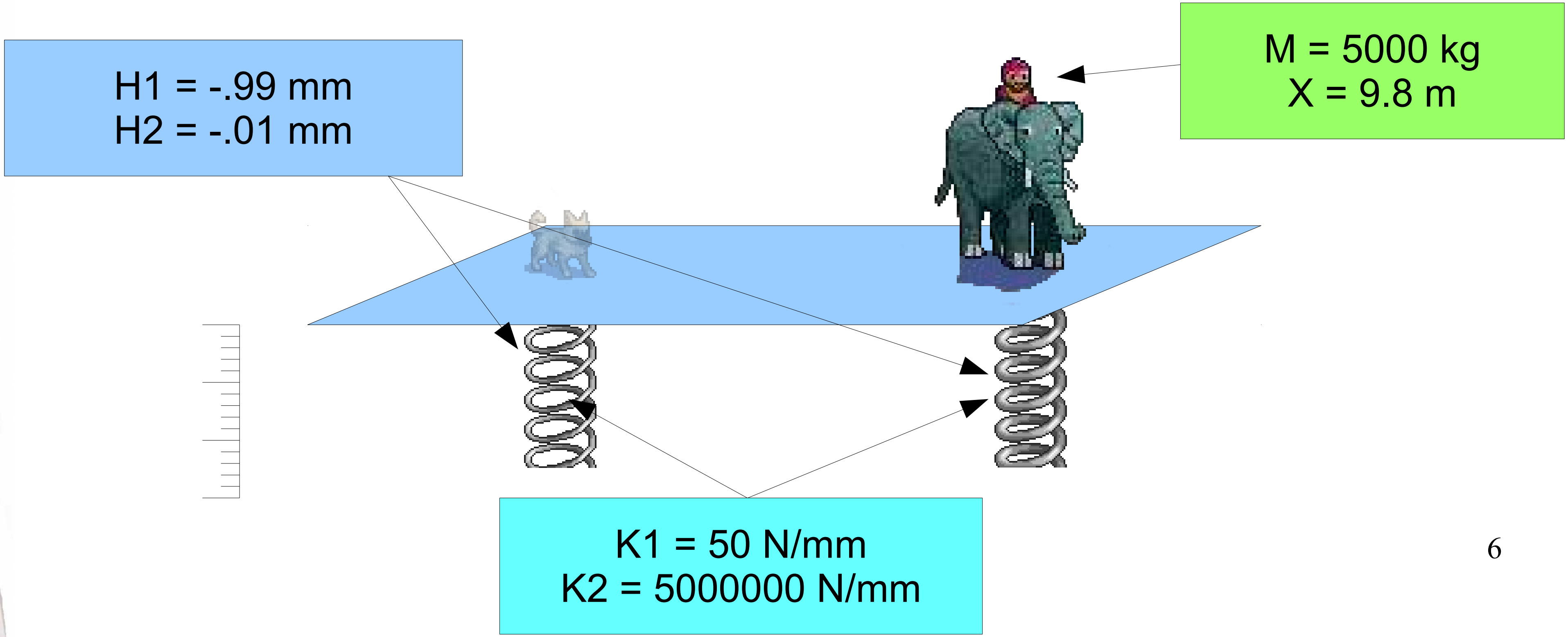
What is an ill-conditionned system ?



Slightly perturbated data

System

Highly perturbated solution



H1 = -.99 mm
H2 = -.01 mm

M = 5000 kg
X = 9.8 m

K1 = 50 N/mm
K2 = 5000000 N/mm

What is an ill-conditionned matrix ?

A is a SDP matrix $$Ax = b$$

Rem : if A is not SDP, work on the normal system

$$A^T A x = A^T b$$

Perturbated system : $$A(x + \delta x) = b + \delta b$$

We have then : $$A\delta x = \delta b$$

Goal : find a stablilty inequality $$\frac{||\delta x||}{||x||} \leq ??? \frac{||\delta b||}{||b||}$$

Relative error on the solution

Relative error on the input

A constant to find

What is an ill-conditionned matrix ?

$$\delta x = A^{-1}\delta b \qquad\qquad Ax = b$$

By definition of the induced 2-norm :

$$||A^{-1}|| = \max \operatorname{eig}(A^{-1}) = \frac{1}{\min \operatorname{eig}(A)} = \max_{\delta b} \frac{||A^{-1}\delta b||}{||\delta b||}$$

$$= \max_{A^{-1}\delta b = \delta x} \frac{||\delta x||}{||\delta b||}$$

$$||A|| = \max \operatorname{eig}(A) = \max_{x} \frac{||Ax||}{||x||} = \max_{Ax=b} \frac{||b||}{||x||}$$

$$\boxed{||b|| \leq \max \operatorname{eig}(A)||x||} \qquad \boxed{||\delta x|| \leq \frac{1}{\min \operatorname{eig}(A)}||\delta b||}$$

What is an ill-conditionned matrix ?

$$||b|| \leq \max \mathrm{eig}(A)||x|| \qquad ||\delta x|| \leq \frac{1}{\min \mathrm{eig}(A)}||\delta b||$$

$$\frac{||\delta x||}{||x||} \leq \boxed{\frac{\max \mathrm{eig}(A)}{\min \mathrm{eig}(A)}} \frac{||\delta b||}{||b||}$$

Condition number of the Matrix

What is the limit ?

100 ?

10^10 ?

10^3567 ?

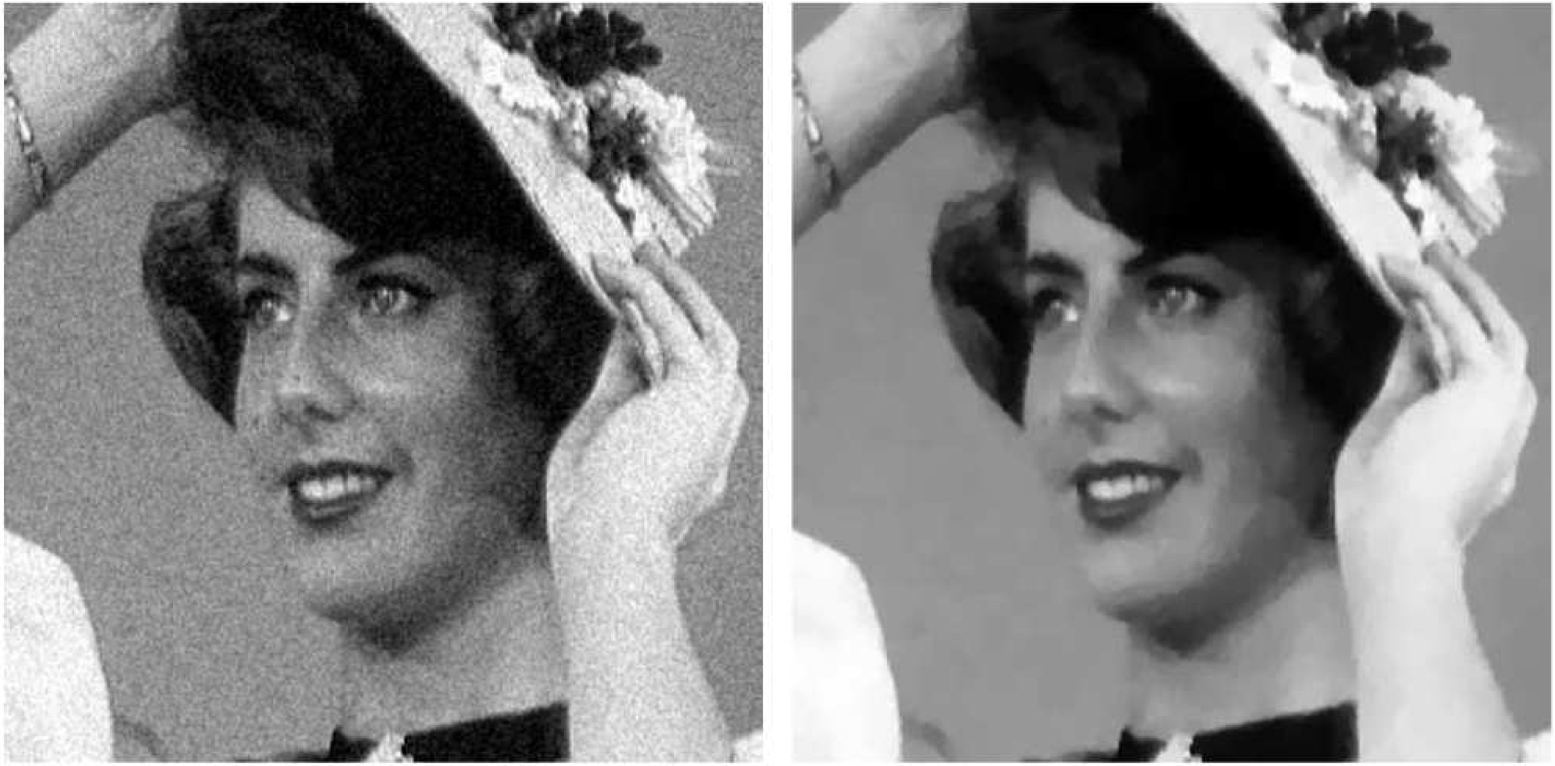
1 ?

It depends

What is an ill-conditionned matrix ?

The acceptable condition number depends on your right hand side.

Extreme case : image denoising [Chambolle 04]



$$Ix = b$$

Identity

De-noised image

Noised image

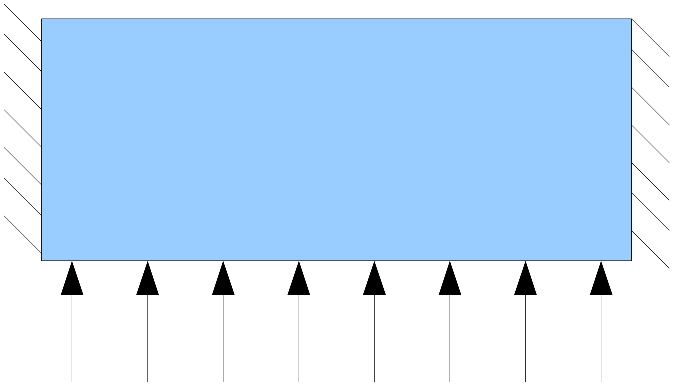
What is an ill-conditionned matrix ?

Other extreme case : control problem

Cf pti déj
0-1-2-3

In that case, there is no noise on the right hand side

Prescripted displacement & 0 Neumann condition

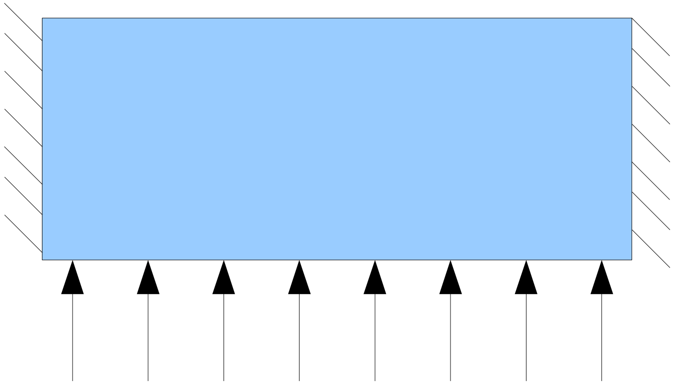


Load to identify

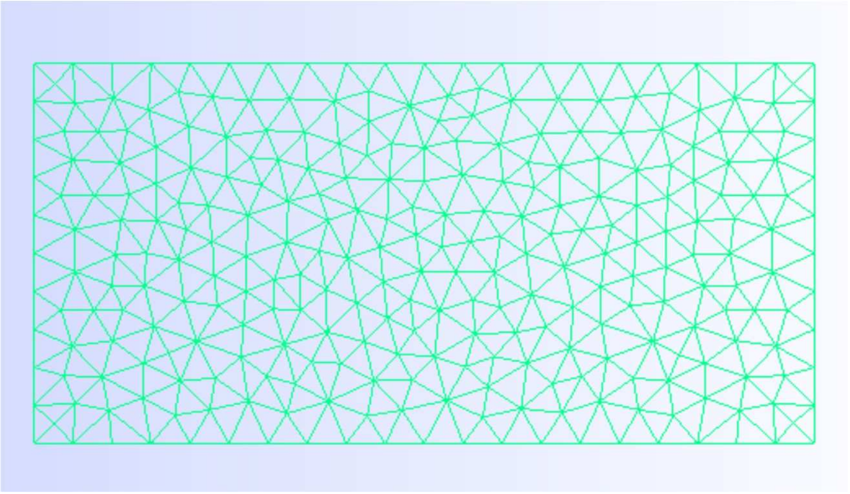
What is an ill-conditionned matrix ?

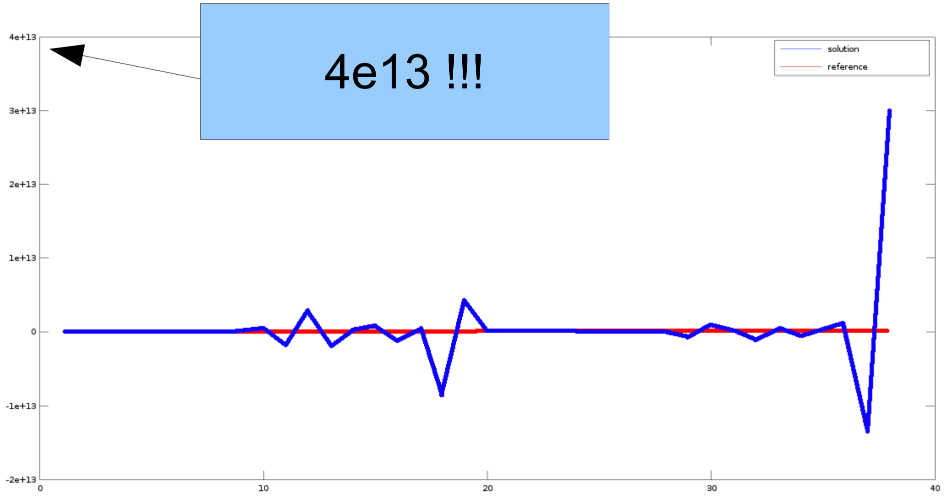
Numerical inversion of the system

Prescripted displacement



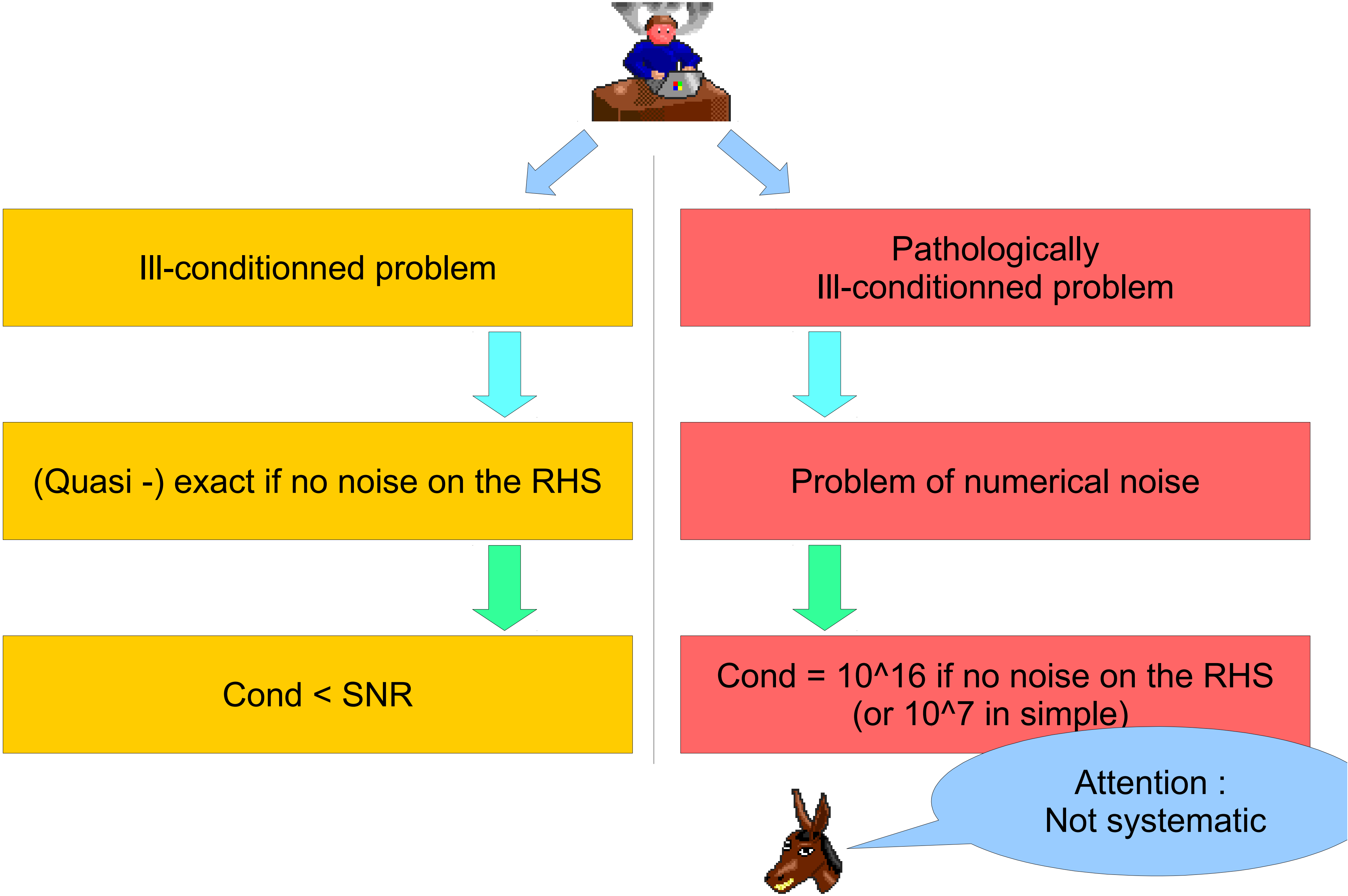
Load to identify





4e13 !!!

What is an ill-conditionned matrix ?



Ill-conditionned problem

(Quasi -) exact if no noise on the RHS

Cond < SNR

Pathologically
Ill-conditionned problem

Problem of numerical noise

Cond = 10^16 if no noise on the RHS
(or 10^7 in simple)

Attention :
Not systematic

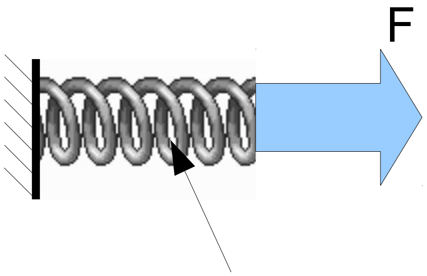
When do you get ill-conditionned matrices ?

When you mix data that don’t have the same magnitude
Exemple : imposing Dirichlet BCs by Lagrange multipliers

When you are modelling an instable physical phenomenon

When you try to do something morally reprehensible
Exemple : inverse problems in general

Example of benign ill-conditionned system



K = 1e30 nN/ly

Equivalent system :

$$\begin{pmatrix} 10^{30} & -10^{30} & 1 \\ -10^{30} & 10^{30} & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ f \\ 0 \end{pmatrix}$$

Equilibrated system :

$$\begin{pmatrix} 10^{30} & -10^{30} & 10^{30} \\ -10^{30} & 10^{30} & 0 \\ 10^{30} & 0 & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ 10^{-30}\lambda \end{pmatrix} = \begin{pmatrix} 0 \\ f \\ 0 \end{pmatrix}$$

How to invert ill-conditionned systems ?

How to invert ill-conditionned systems ?

First idea : try to see if the case is benign

Try scaling the lines and columns

The Frobenius norm
Is very good for that

Try permutating lines or columns

Linear solvers won’t necessary do it for you

How to invert ill-conditionned systems ?

General idea for inverse problems

$$Ax = b$$

Full-rank system

Rank deficient system

Only 1 solution : the abherrant one

An infinity of solutions,
all of them are likely to be aberrant

And none of them is the solution of the underlying physical problem

We don’t trust the system : we need extra information

Careful balance between extra information and the system

How to invert ill-conditionned systems ?

Choice of the balance parameter

Methods that use the noise level

Effecient if the condition number is < 10^16

Fairly stable in that case

Methods that don’t use the noise level

Bakushinsky veto [Bakushinsky 84]
The method will not converge in the worst case

Quite used and reliable in practice

Tikhonov Regularization

Tikhonov regularization

$$Ax = b$$

Normal equations :

$$x = \arg\min_{x^*} \|Ax^* - b\|^2 \iff A^T A x = A^T b$$

We'd like to control the distance to a reference x0 wrt. a certain norm.

$$\|x - x_0\|_n$$

Final minimization problem [Tikhonov 63]

$$x = \min_{x^*} \overset{\text{Residual}}{\boxed{\|Ax^* - b\|^2}} + \overset{\text{Balance parameter}}{\boxed{\mu}} \overset{\text{Regularization term}}{\boxed{\|x^* - x_0\|_n^2}}$$

Tikhonov regularization

Choice of x0 and n

$$x = \min_{x^*} ||Ax^* - b||^2 + \mu||x^* - x_0||_n^2$$

Ex : Young modulus of a variant of steel
x0 = 210 000 MPa

In some cases, we don’t know what x0 to use

In those cases, x0 = 0

n can be the L2 norm

Or, for mechanical fields, a norm penalizing the gradients
Or the gap to equilibrium [Claire 04]

Tikhonov regularization

Quadratic Tikhonov regularization

$$x = \min_{x^*} \|Ax^* - b\|^2 + \mu\|Lx^* - Lx_0\|^2$$

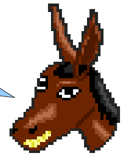
The functionnal is convex (if μ > 0)
Minimum atteigned for the zero of the gradient :

$$\boxed{(A^T A + \mu L^T L)x = A^T b + \mu L^T L x_0}$$

Rem : is case A is SDP, the following problem is also relevant :

$$(A + \mu L^T L)x = b + \mu L^T L x_0$$

And even if A is only squared, this may work



But it’s dirty !

Tikhonov regularization

Remark on the normal equation for squared matrices :

$$Ax = b \qquad \text{vs} \qquad A^T A x = A^T b$$

cond(A’A) = cond(A)^2

If A is reasonably well-conditionned,
Working on one or the other system is equivalent

Because the error comes from the noise on b
That is multiplied by A’

If A is too badly conditionned
(ie float point errors enter into consideration),
The simple system is preferable

Because the error comes from every operation

Tikhonov regularization

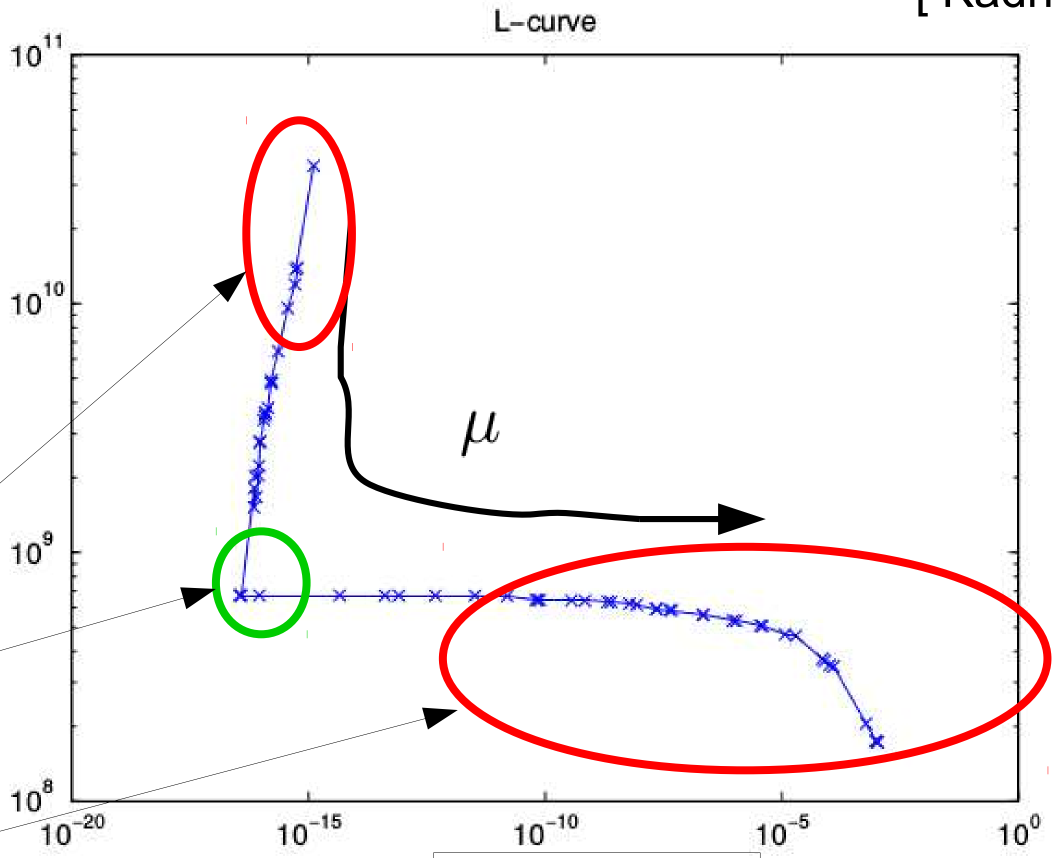
Determination of the regularization parameter μ

$$x = \min_{x^*} \|Ax^* - b\|^2 + \mu \|x^* - x_0\|_n^2$$

[Kadri 11]

L-curve [Hansen 92]

$\|x - x_0\|_n$



Not enough regularization

Good balance

Too much regularization

$\|Ax^* - b\|$

Tikhonov regularization

Determination of the regularization parameter μ

$$x = \min_{x^*} \|Ax^* - b\|^2 + \mu \|x^* - x_0\|_n^2$$

Morozov : $\|Ax - b\|^2 = \|\delta b\|^2$

Arcangeli : $\|Ax - b\|^2 = \dfrac{\|\delta b\|^2}{\mu}$

Arcangeli's criterion
Works only if L = 1

And was proven to have a
good convergence wrt
the noise

Rem : $\|\delta b\|^2$ Is the residual of the solution of the noise free problem

Truncated Generalized Singular Values Decomposition

Truncated Generalized Singular Values Decomposition

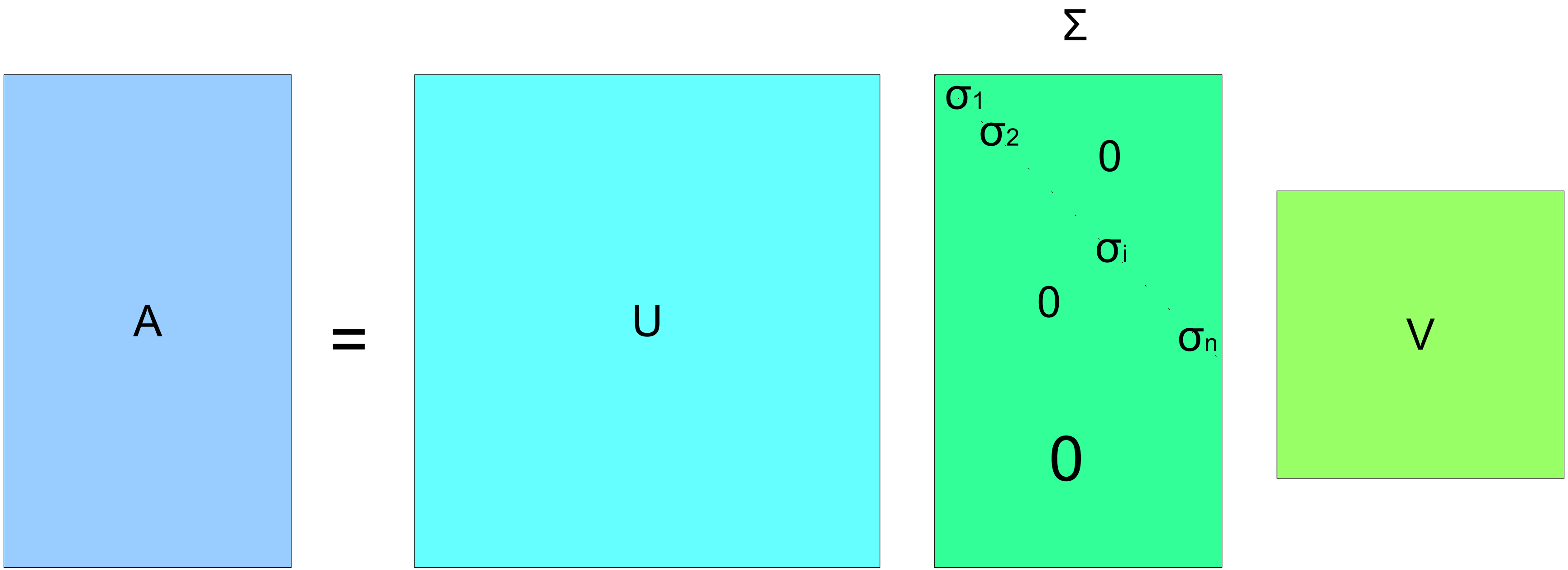
$$Ax = b$$

$$A = U\Sigma V^T$$

With :

$$U^T U = 1 \qquad V^T M_v V = 1$$

$$M_v^T = M_v$$



Σ

A

=

U

σ_1

σ_2

0

σ_i

0

σ_n

0

V

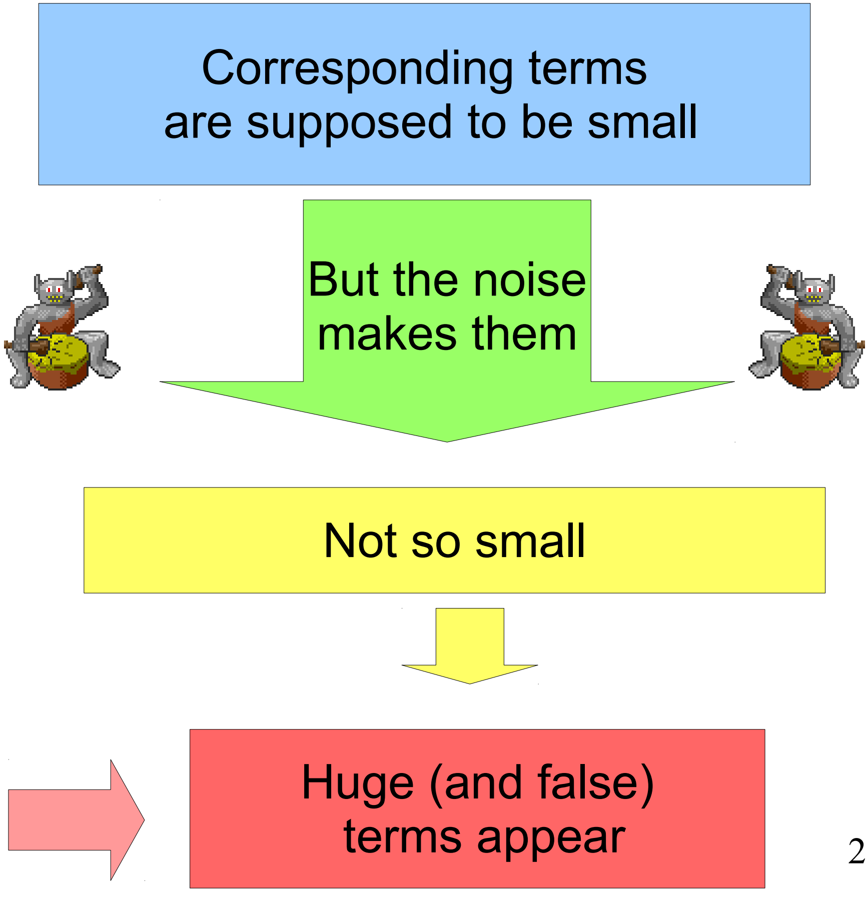
Truncated Generalized Singular Values Decomposition [Hansen 87]

$$Ax = b \qquad A = U\Sigma V^T$$

$$U\Sigma V^T x = b$$

$$x = M_v V \Sigma^{-1} U^T b$$

$$\Sigma^{-1} = \begin{pmatrix} \frac{1}{\sigma_1} & & & \\ & \frac{1}{\sigma_2} & & \\ & & \ddots & \\ & & & \frac{1}{\sigma_n} \end{pmatrix}$$



Corresponding terms are supposed to be small

But the noise makes them

Not so small

Huge (and false) terms appear

Inverse of small values => Big

Truncated Generalized Singular Values Decomposition [Hansen 87]

$$x = M_v V \Sigma^{-1} U^T b$$

$$\tilde{\Sigma}_k^{-1} = \begin{pmatrix} \frac{1}{\sigma_1} & & & & & \\ & \ddots & & & & \\ & & \frac{1}{\sigma_k} & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix}$$

$$\boxed{x_k = M_v V \tilde{\Sigma}_k^{-1} U^T b}$$

Truncated Generalized Singular Values Decomposition

$$Ax = b$$

$$x_k = M_v V \tilde{\Sigma}_{\boxed{k}}^{-1} U^T b$$

Parameter of the method

Rem : it is possible to add an a-priori known term in $\mathfrak{Span}(\Sigma - \tilde{\Sigma}_k)$

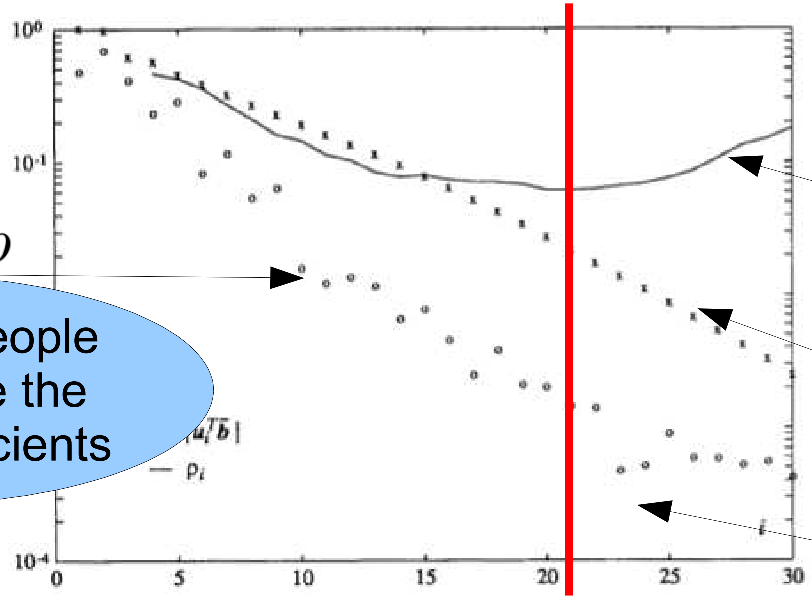
PB : generalization for non-linear problems

Truncated Generalized Singular Values Decomposition

Choice of the nb of singular values (k)

L-curve

Picard Plot [Hansen 90]



Right hand side $U^T b$

Solution $\tilde{\Sigma}_k^{-1} U^T b$

Singular values

Stagnation of the Rhs

In practice, people Prefer to use the Fourier coefficients

The Morozov criterion is still avaliable

Truncated Generalized Singular Values Decomposition

Remank : The damped GSVD [Ekstrom 74]

$$\tilde{\Sigma}_{\mu}^{-1} = \begin{pmatrix} \dfrac{1}{\sigma_1 + \mu} & & & \\ & \dfrac{1}{\sigma_2 + \mu} & & \\ & & \ddots & \\ & & & \dfrac{1}{\sigma_n + \mu} \end{pmatrix}$$

This is equivalent to the Tikhonov regularization

This gives a new method to compute the optimal parameter for Tikhonov (adaptation of Picard)

And generalization to NL problems is then trivial

Early-stopped iterative algorithm

Early-stopped iterative algorithm

$$\{x_i\}_{i=0..n}, \lim_{i \to +\infty} x_i \text{ Solution of } Ax = b$$

Usual criterion on the residual (<1e-12)

In case we don’t let the algorithm finish :

Exemple : Richardson algorithm : eig(A) < 1

$$x_{i+1} = (\mathcal{I} - A)x_i + b$$

$$x_{i+1} = V(\mathcal{I} - \Theta)V^T x_i + b$$

$$\tilde{x}_{i+1} = (\mathcal{I} - \Theta)\tilde{x}_i + \tilde{b}$$

The small eigenvalues converge slowly

Early-stopping doesn’t take them into account

Early-stopped iterative algorithm

Usually, in iterative algorithms, last eigenvalues converge more slowly

Link with the truncated eigenvalues method

Mainly for Krylov solvers

(Block-) Conjugate Gradient

GMRES / Orthodir [Calvetti 02]

Conjugate Residuals

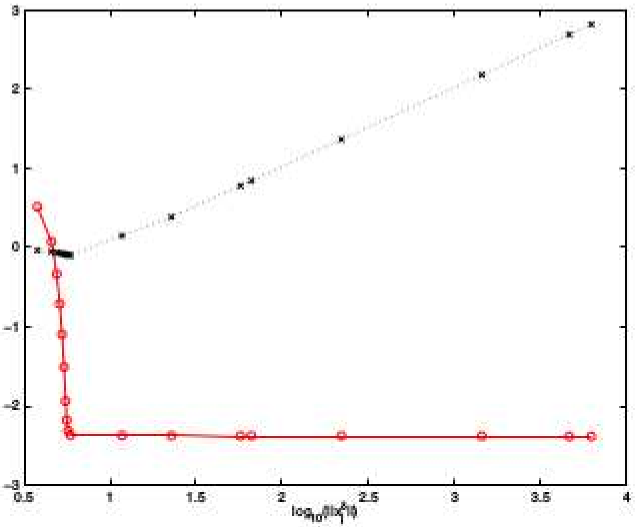
But also for Steepest descent (that is not a Krylov)

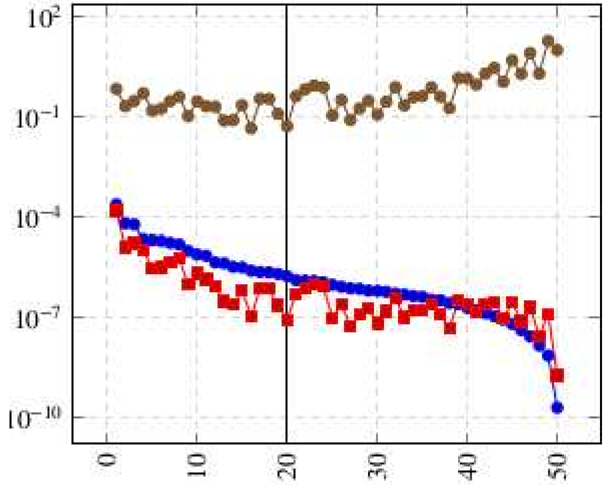
Early-stopped iterative algorithm

Choice of the stopping criterion

L-curve

Picard plot
(in some cases)

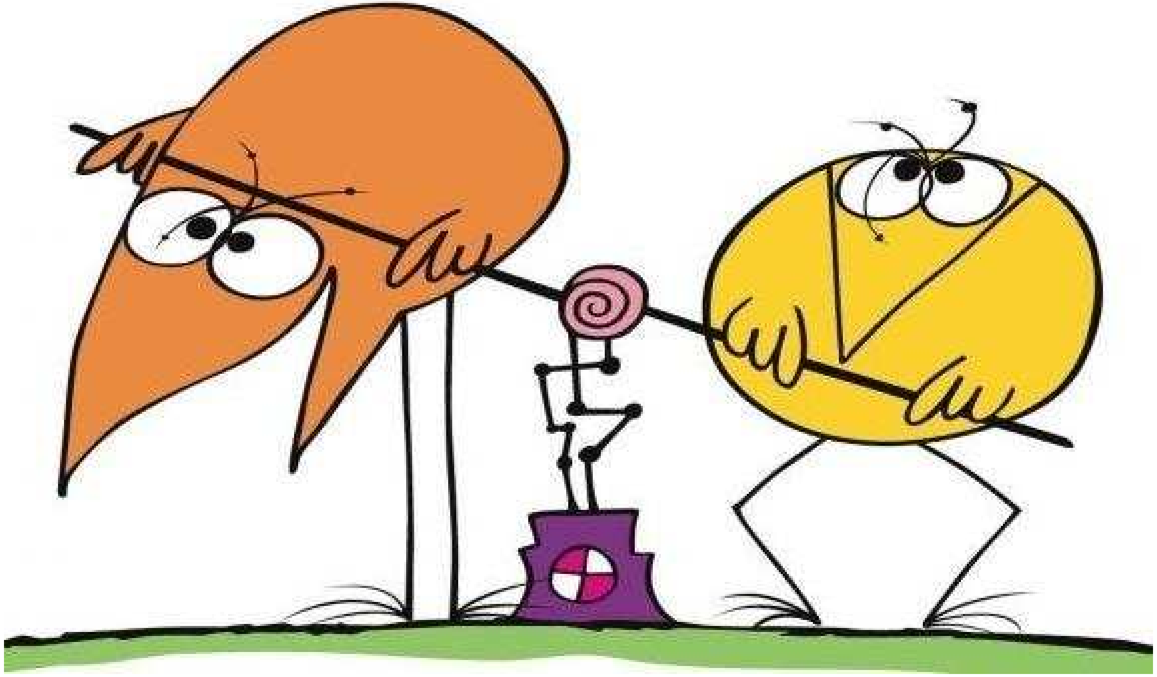




[Calvetti 02]

[🐴 18]

Bayesian inversion



Bayesian inversion

$$Ax = b$$

we are set in the linear setting but the method is more general

In the probabilistic setting

$$\tilde{x} \qquad \tilde{b}$$

A-priori probability law of the unknown (ex : isoprobability)

$$\mathfrak{p}(\tilde{x} = x)$$

Probability law of the right hand side (from the measurement uncertainty)

$$\mathfrak{p}(\tilde{b} = b | \tilde{x} = x)$$

Bayes theorem gives the a-posteriori law of the unknown

$$\mathfrak{p}(\tilde{x} = x | \tilde{b} = b)$$

Bayesian inversion

$$Ax = b$$

Bayes theorem :

$$\overbrace{\mathfrak{p}(\tilde{x} = x|\tilde{b} = b)}^{\text{a-posteriori}} = \frac{\overbrace{\mathfrak{p}(\tilde{b} = b|\tilde{x} = x)}^{\text{measure}}\overbrace{\mathfrak{p}(\tilde{x} = x)}^{\text{a-priori}}}{\underbrace{\mathfrak{p}(\tilde{b} = b)}_{\text{Normalization constant}}}$$

Bayesian inversion

Remark on the probability law of the right hand side

$$\mathfrak{p}(\tilde{b} = b | \tilde{x} = x)$$

If there is no measurement noise, b = Ax

The probability is function of the distance between b and Ax
And of the measurement noise level

Estimated correlation of the measurement noise

$$\mathfrak{p}(\tilde{b} = b | \tilde{x} = x) = \frac{1}{(2\pi)^{n_b/2}\sqrt{\det(C_b)}} e^{-\frac{1}{2}(b-Ax)^T C_b^{-1} (b-Ax)}$$

Normalization stuff

Residual between x and b

For each value of x
An evaluation of Ax is required

Bayesian inversion

$$Ax = b$$

We know thanks to Bayes theorem : $\mathfrak{p}(\tilde{x} = x|\tilde{b} = b)$

But a numerical result is needed. For that we can compute :

The mean

The arg max of the probability law

The median

The covariance matrix

Upper order statistical moments

Bayesian inversion

Computation of the mean (similar for the other quantities)

$$\mathfrak{E}(\tilde{x}|\tilde{b}=b)=\int_{\mathcal{R}^n} x\mathfrak{p}(\tilde{x}=x|\tilde{b}=b)dx$$

$$\mathfrak{E}(\tilde{x}|\tilde{b}=b)=\int_{\mathcal{R}^n} x\frac{\mathfrak{p}(\tilde{b}=b|\tilde{x}=x)\mathfrak{p}(\tilde{x}=x)}{\mathfrak{p}(\tilde{b}=b)}dx$$

$$\int_{\mathcal{R}^n} \mathfrak{F}(x)\mathfrak{p}(\tilde{x}=x)dx$$

➡ Monte-Carlo integration

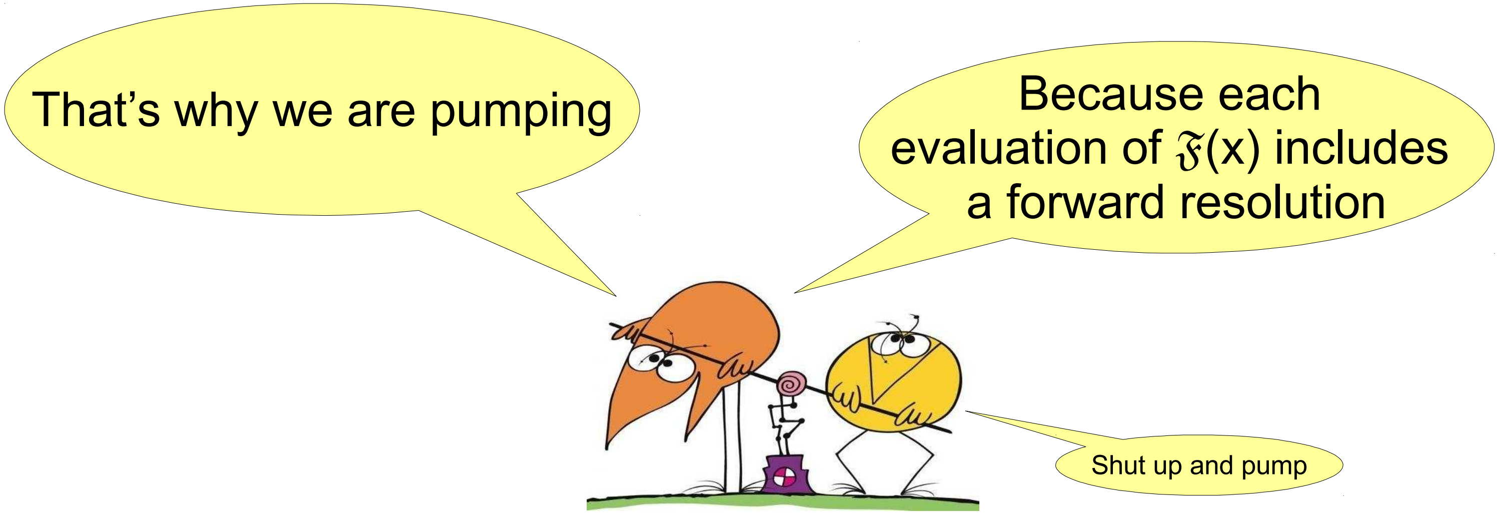
Bayesian inversion

Monte-Carlo integration

$$\int_{\mathcal{R}^n} \mathfrak{F}(x)\mathfrak{p}(\tilde{x} = x)dx$$

Sample the a-priori probability density

Compute the associated $\mathfrak{F}$(x)
including the probability of an output knowing the input


That's why we are pumping
Because each evaluation of $\mathfrak{F}$(x) includes a forward resolution
Shut up and pump

Sum the $\mathfrak{F}$(x) and divide by the nb of samples

Bayesian inversion

Monte-Carlo integration convergence properties

$$\int_{\mathcal{R}^n} \mathfrak{F}(x)\mathfrak{p}(\tilde{x}=x)dx$$

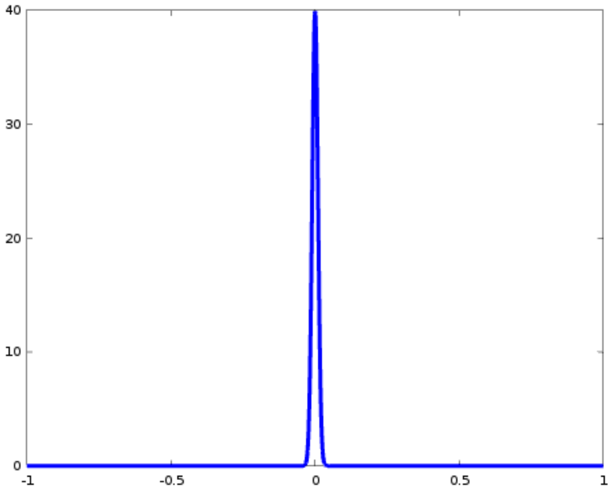
Converges $\mathfrak{p}$ - almost surely with rate K n^1/2
(n is the number of samples)

And K may be very small if $\mathfrak{p}$ is « large »
And $\mathfrak{F}$ is « small »

Exemple :

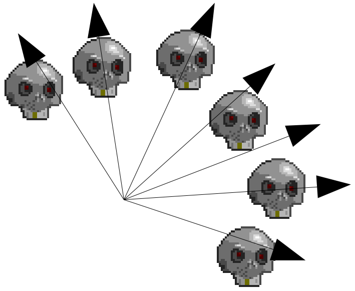
$\mathfrak{p}$(x) is the equiprobable law on [-1;1]

$\mathfrak{F}$ is gaussian of variance 1/100



+

Curse Of Dimensionality



Bayesian inversion

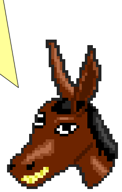
How to cure those bad properties ?

Multilevel Monte-Carlo [Giles 08]

Markov Chain [Gilks 95]

Kalman Filter [Kalman 60]

Though Kalman claimed that his method had nothing to do with Bayesian inversion



Transport Maps method / Variational Bayesian

Reduced forward model [Rubio 18]

Bayesian inversion

Remark in the case of Gaussian probabilities (1/2

Because of the Central Limit Theorem Many people use gaussian probabilities

$$\mathfrak{p}(\tilde{x} = x) = \frac{1}{(2\pi)^{n_x/2}\sqrt{\det(C_x)}} e^{-\frac{1}{2}(x-x_0)^T C_x^{-1}(x-x_0)}$$

$$\mathfrak{p}(\tilde{b} = b|\tilde{x} = x) = \frac{1}{(2\pi)^{n_b/2}\sqrt{\det(C_b)}} e^{-\frac{1}{2}(b-Ax)^T C_b^{-1}(b-Ax)}$$

$$\mathfrak{p}(\tilde{x} = x|\tilde{b} = b) = Cte \cdot e^{-\frac{1}{2}\left(x^T(A^T C_b^{-1} A + C_x)x - 2(x_0^T C_x^{-1} + b^T C_b^{-1} A)x + x_0^T C_x^{-1} x_0 + b^T C_b^{-1} b\right)}$$

Bayesian inversion

Remark in the case of Gaussian probabilities (2/2)

The mean of x respects :

$$\min_x x^T(A^T C_b^{-1} A + C_x)x - 2(x_0^T C_x^{-1} + b^T C_b^{-1} A)x + x_0^T C_x^{-1} x_0 + b^T C_b^{-1} b$$

$$(A^T C_b^{-1} A + C_x^{-1})x = (C_x^{-1} x_0 + A^T C_b^{-1} b)$$

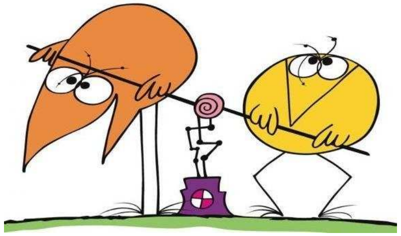
We study the particuliar case where :

$$C_b^{-1} = \nu \mathcal{I} \qquad C_x^{-1} = \eta \mathcal{I} \qquad \mu = \frac{\nu}{\eta}$$

$$(A^T A + \mu \mathcal{I})x = (\mu x_0 + A^T b)$$

In the gaussian case, Bayesian inversion can be seen as a way to determine the regularization parameter

But how do you find
the variance of the prior law ?



Relaxtion of the constraint

Relaxation of the constraint
[Ladevèze 93]

$$\min_{Cx=\enclose{circle}{d}} \|Ax-b\|^2$$

Noisy

$$\min_{x} \|Ax-b\|^2 + \mu\|Cx-d\|^2$$

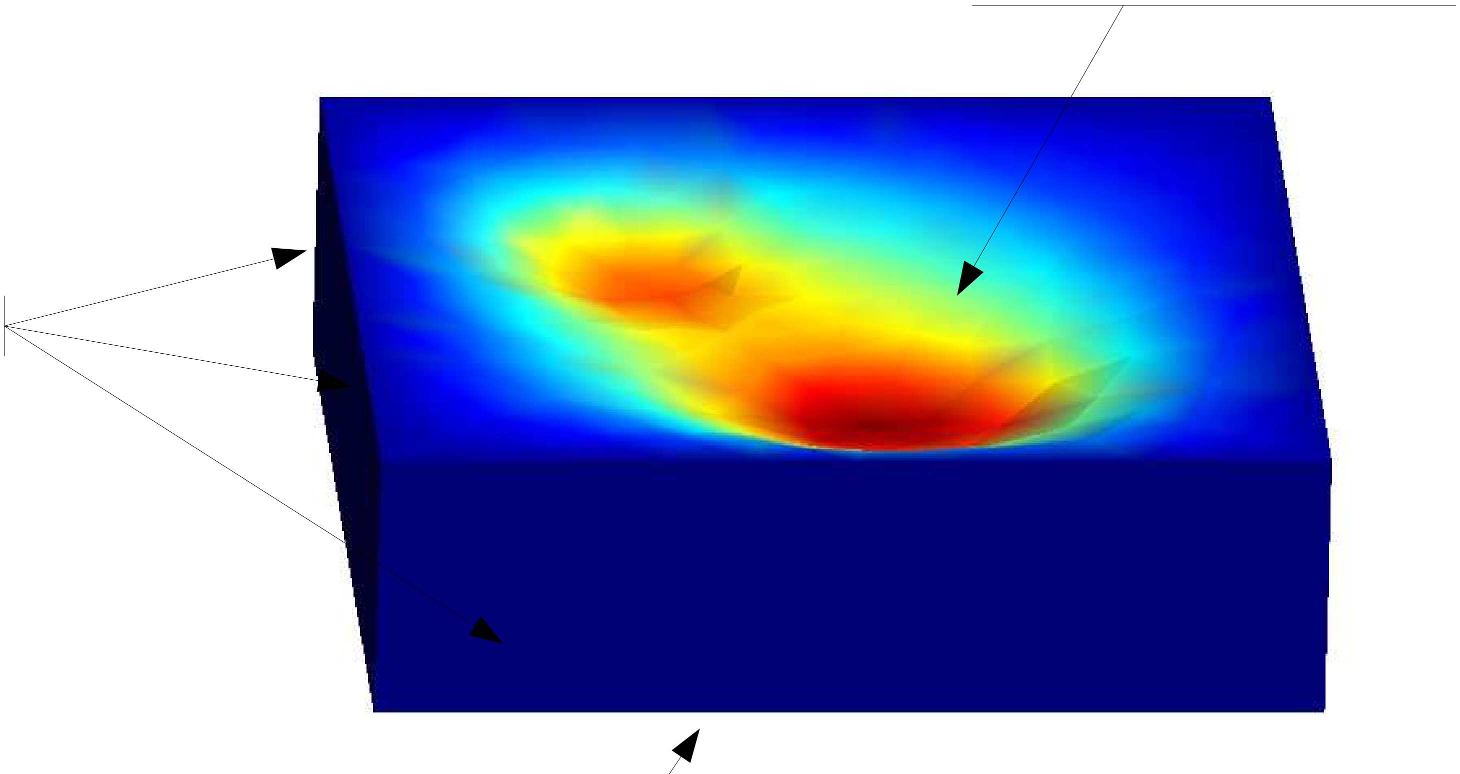
Useful only in the case where
(A ; C) is overdetermined

Illustrative examples

Illustrative examples

Identification of BCs (Cauchy problem)

Cf pti déj 0-1-2-3



Identification of U and F

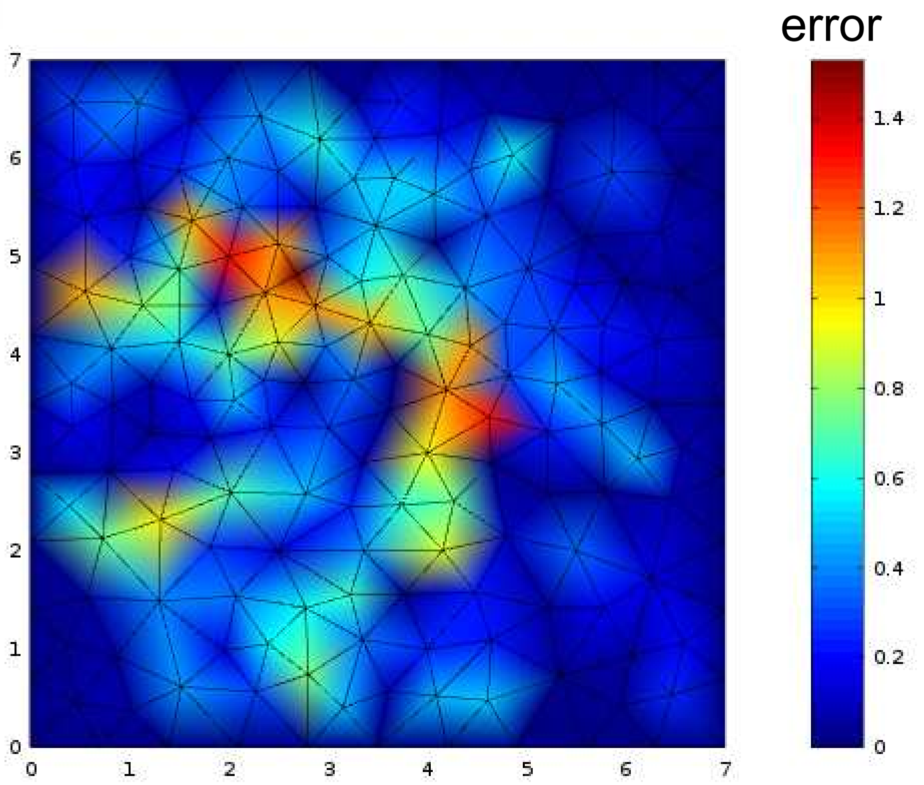
U = 0

F = 0 & Measurement of U

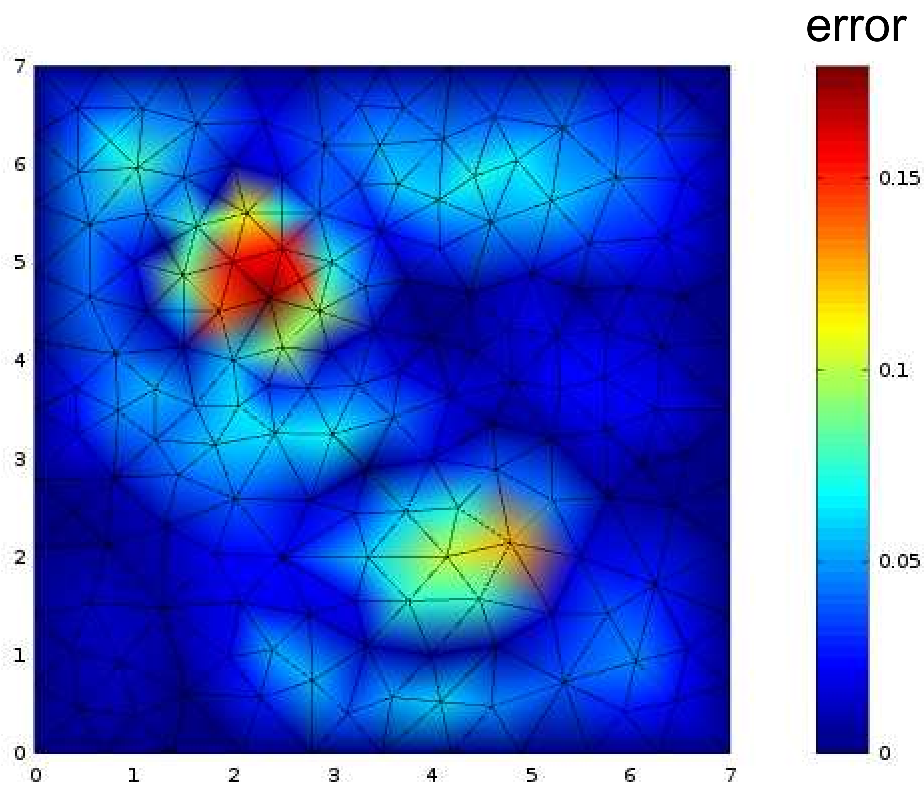
Illustrative examples

Identification of BCs (Cauchy problem)

Cf pti déj 0-1-2-3



10 % noise, no regularization

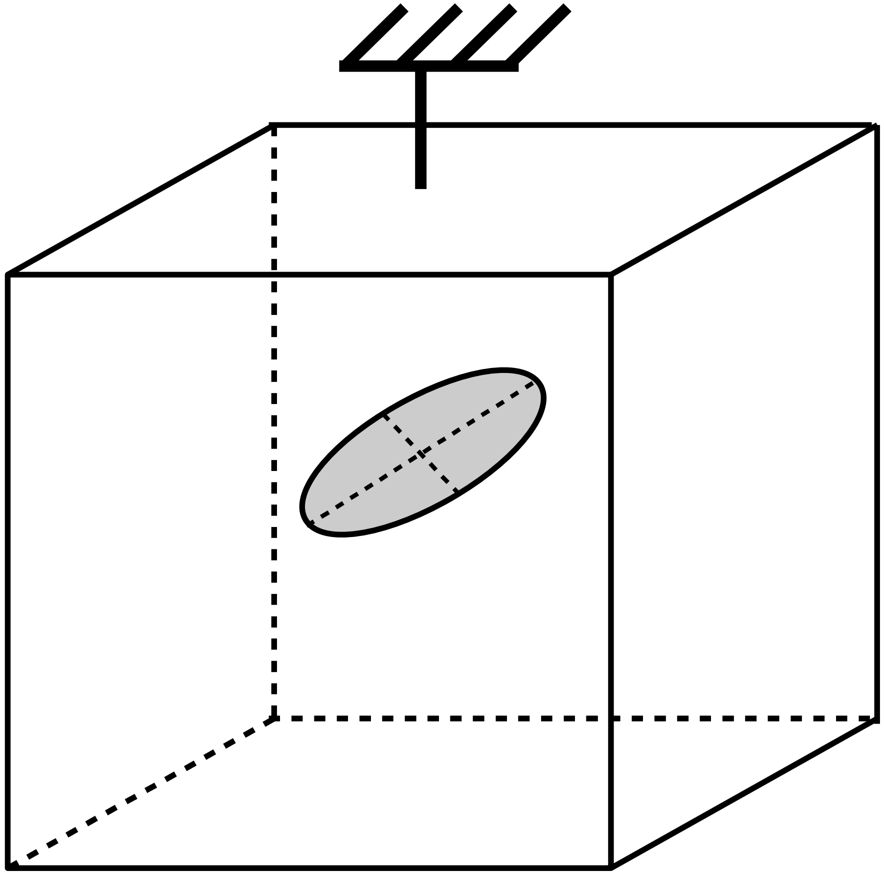


10 % noise,
Regularization by early-stopped conjugate gradient and Ritz post-analysis

Illustrative examples

Identification of a displacement gap (crack)

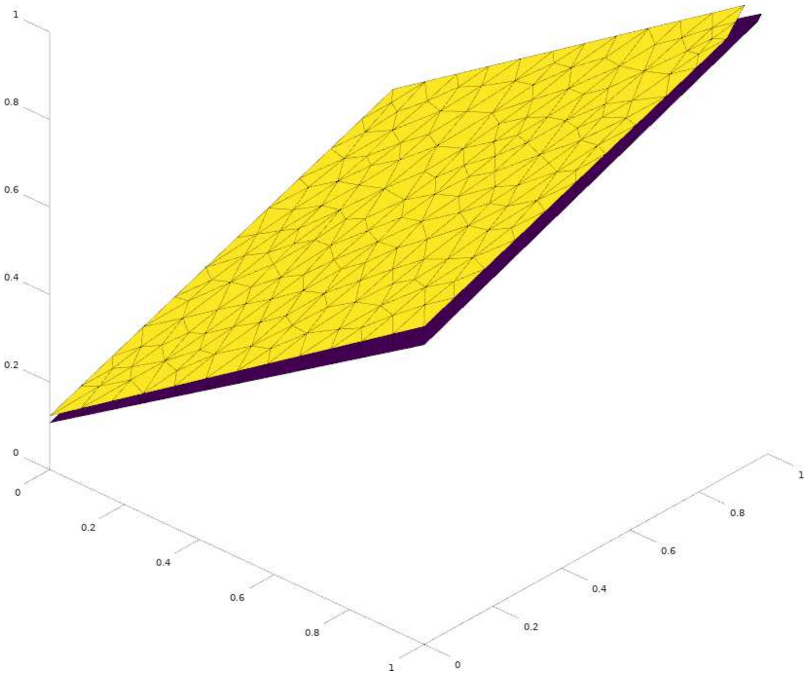
Cf pti déj
4



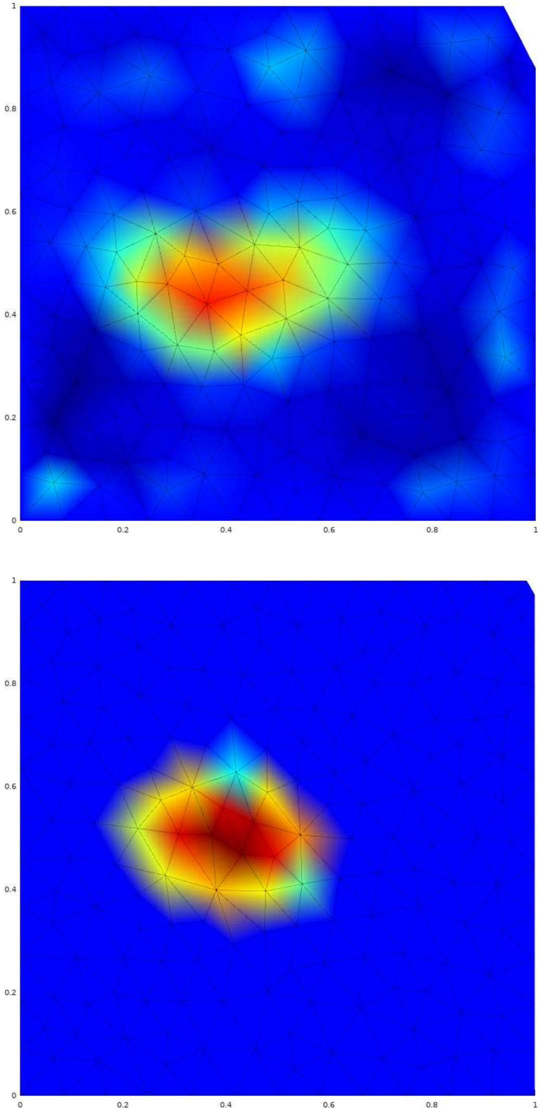
Illustrative examples

Identification of a displacement gap (crack)

Cf pti déj
4



Identification of the plane
(Tikhonov regularization)



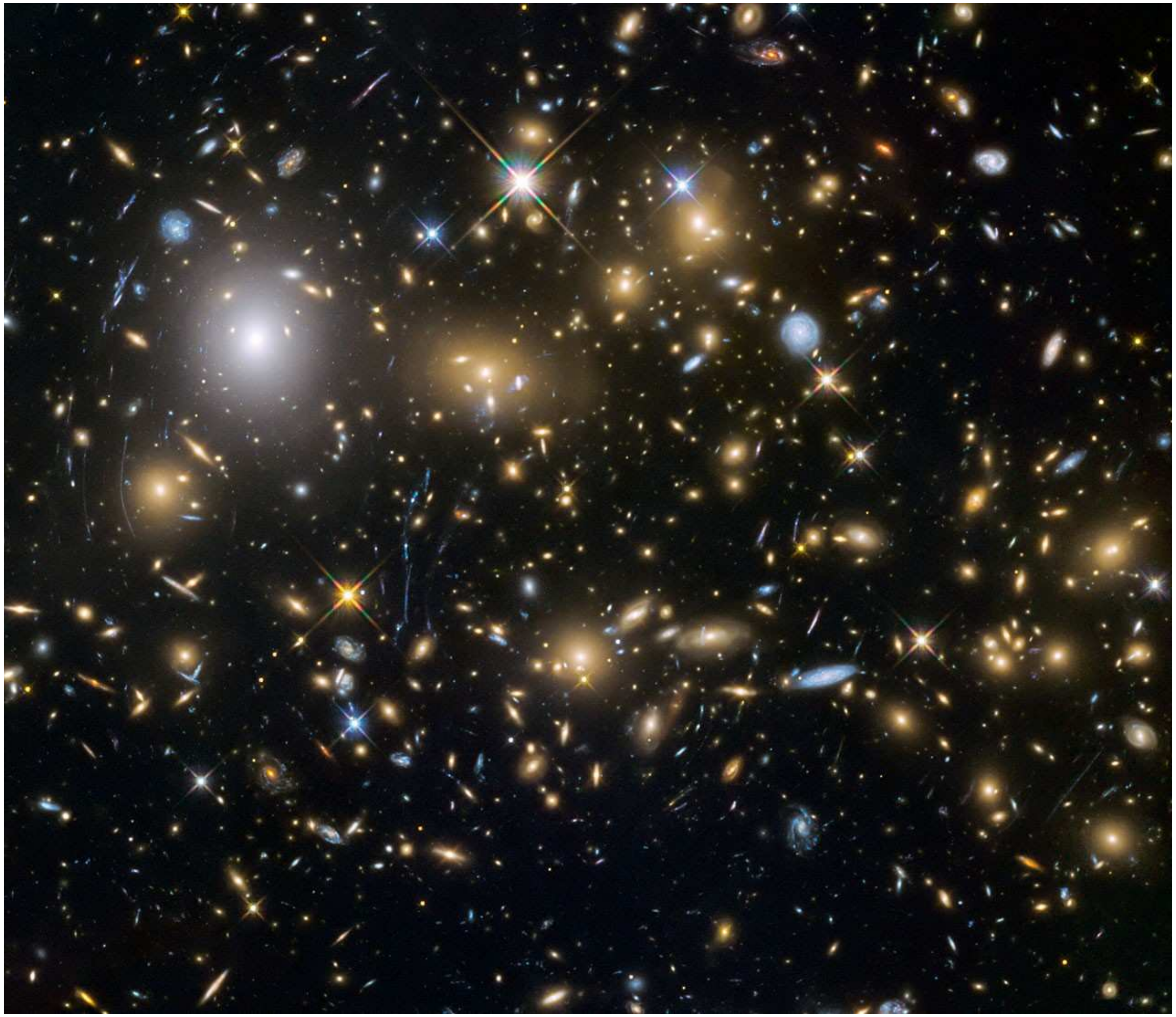
Identification of the gap
(Identification and reference)

Remark : library for regularization
By Christian Hansen

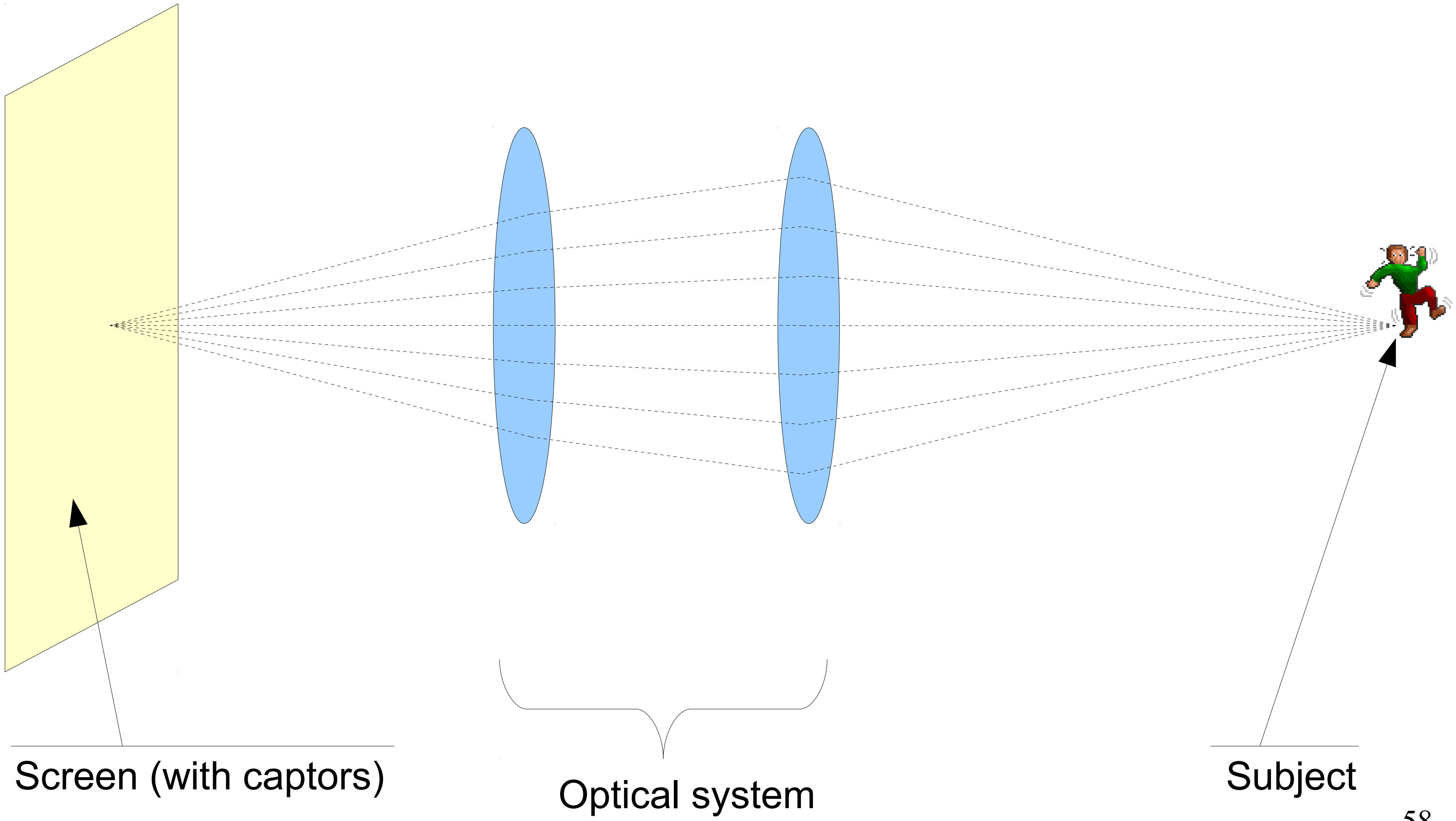
Featuring tools to implement in practice the presented methods
And many other nice features as well !

http://www.imm.dtu.dk/~pcha/Regutools/

Applicative example : correction of blurred images



Applicative example : Correction of blurred images

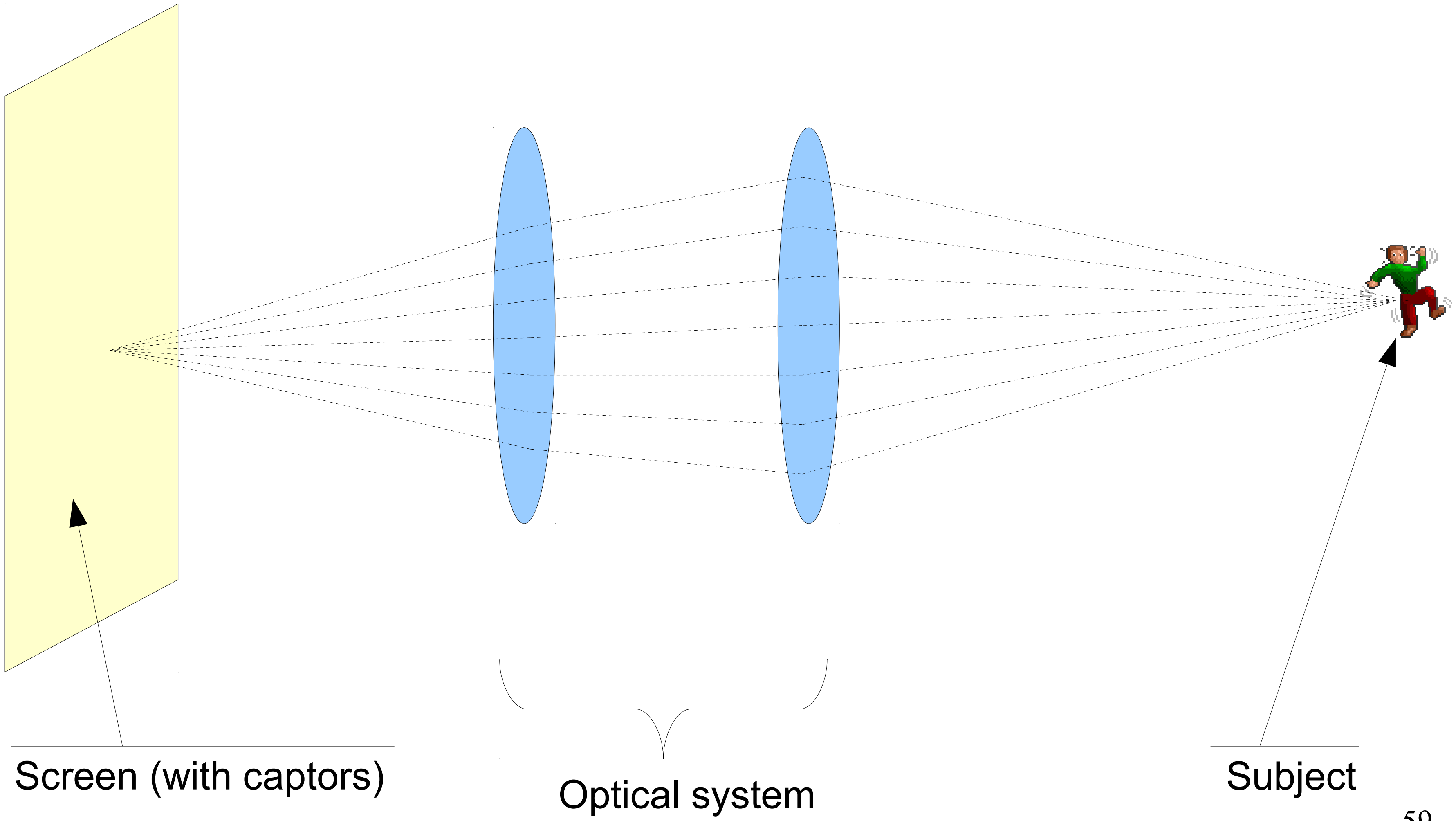


Screen (with captors)

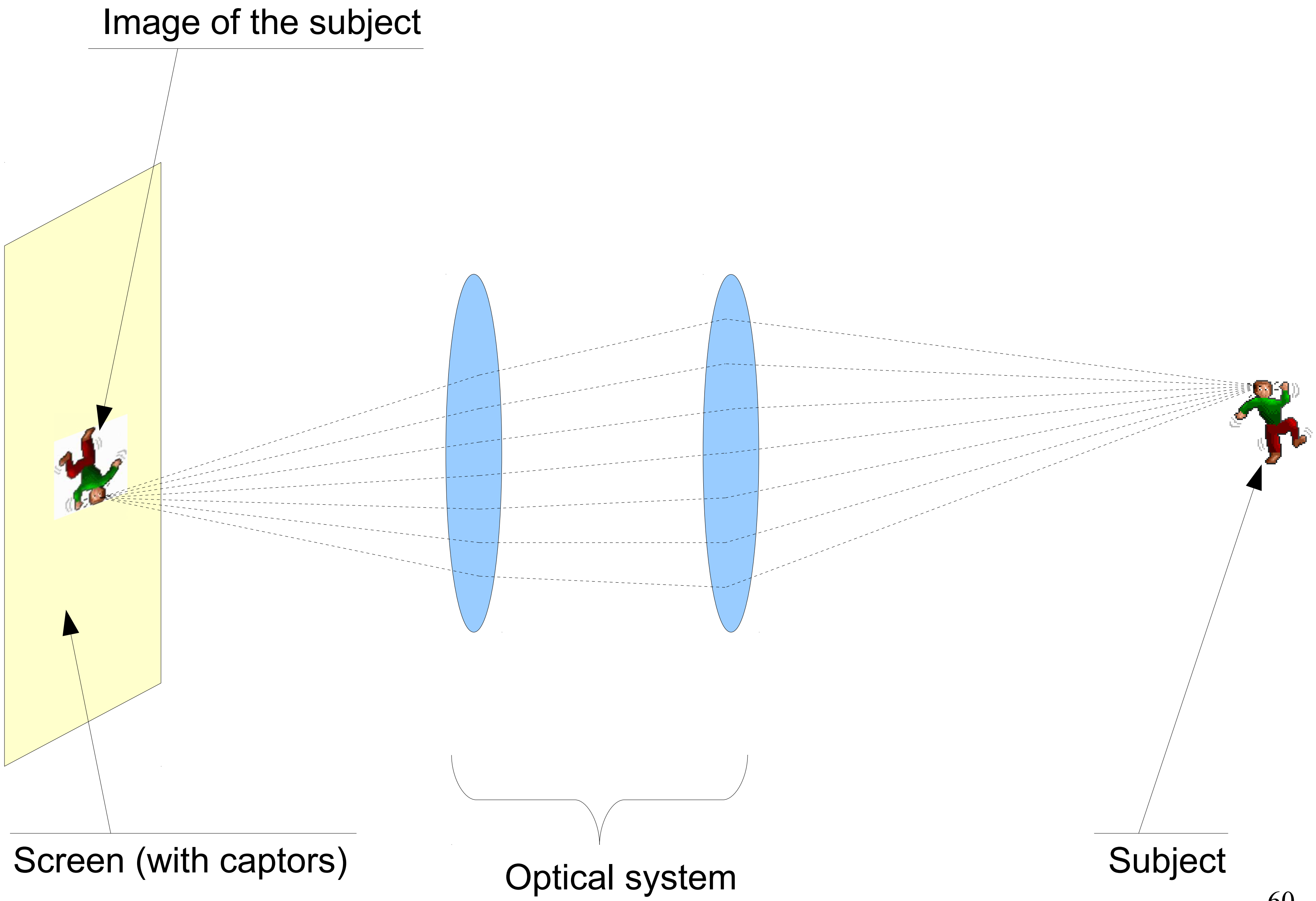
Optical system

Subject

Applicative example : Correction of blurred images

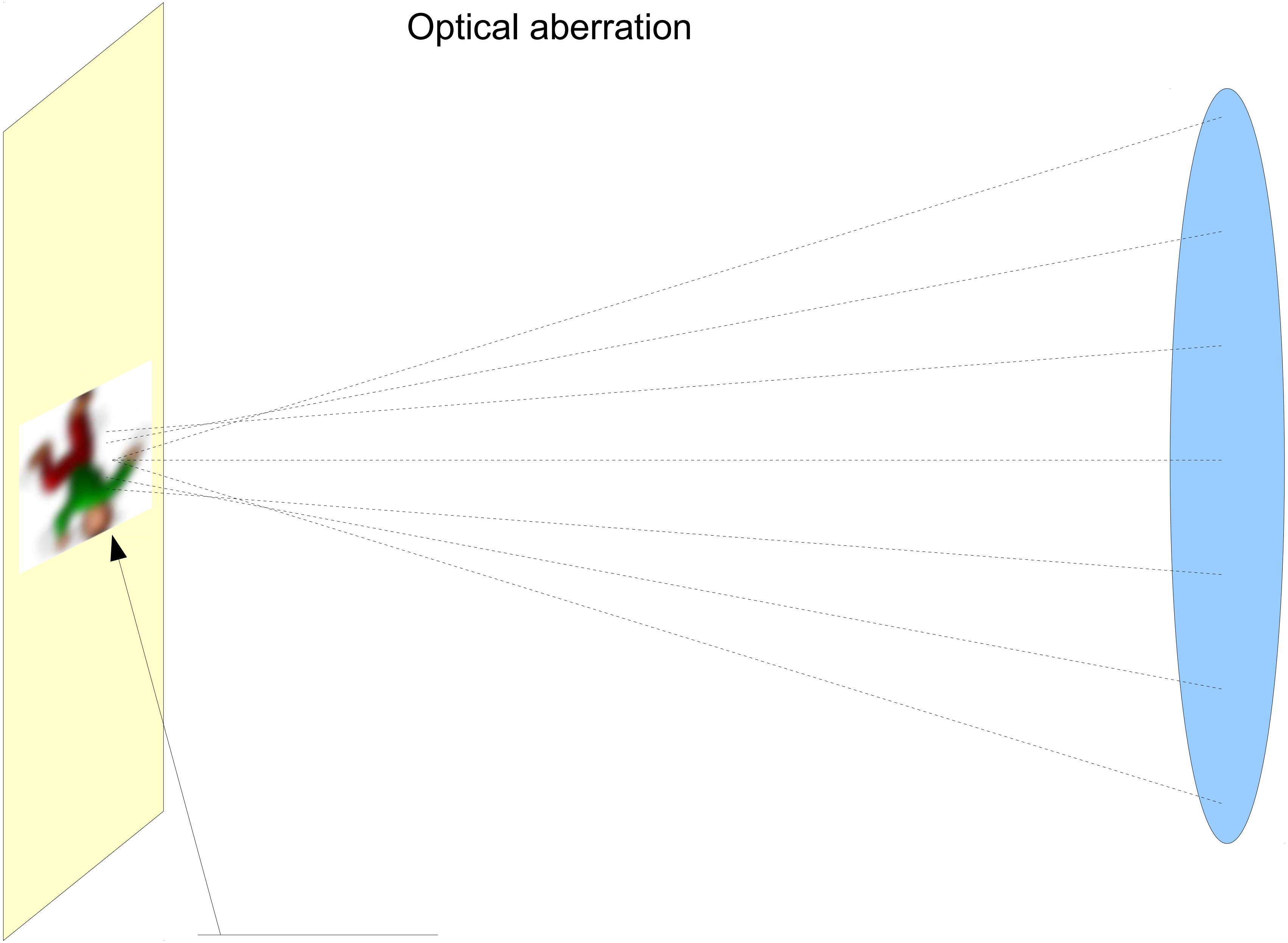


Applicative example : Correction of blurred images



Applicative example : Correction of blurred images

Optical aberration



Blurred image

Applicative example : Correction of blurred images

In some applications, it is necessary to post-process the image

Need of very precise pictures

Or

Very bad optics

Model the blur as a linear operator.
Invert that operator in order to recover the image

Problem :

This operator is Patologically Ill-conditionned

Other Problem :

There is also noise

Applicative example : Correction of blurred images

In normal applications, high quality images (~3000 X 3000 pix)

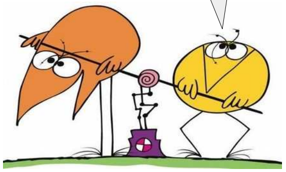
Full operator of size 9 10^6

High spec hardware & optimized (and compiled) code
(Matrix-free methods, GPU computing...)

We use laptops on an unoptimized code (pedagogically optimized)

Looks like someone here is coding with his ass...

Nice excuse



Our pictures are very small

⚠ The behaviour of real images wrt the algos
May be slightly different to the behaviour of
The proposed ones ⚠

Applicative example : Correction of blurred images

The gaussian blur model (one among many) :

$$\boxed{I_B(\underline{x})} = \frac{1}{\boxed{N(\underline{x})}} \boxed{\int_\Omega e^{\frac{-(\underline{x}-\underline{\tilde{x}})^2}{2\sigma^2}} \boxed{I(\underline{\tilde{x}})}\, d\underline{\tilde{x}}}$$

Fredholm integral

Trouble

Blurred image

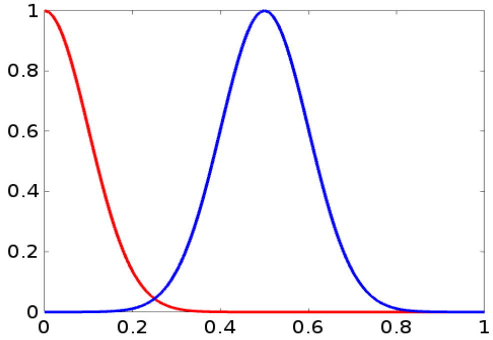
Normalization factor

Unblurred image

$$N(\underline{x}) = \int_\Omega e^{\frac{-(\underline{x}-\underline{\tilde{x}})^2}{2\sigma^2}} d\underline{\tilde{x}}$$

$$\boxed{\mathcal{B}(\underline{x},\underline{y}) = \frac{1}{N(\underline{x})} e^{\frac{-(\underline{x}-\underline{y})^2}{2\sigma^2}}}$$

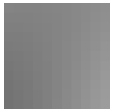
Attention :
This operator is not symmetric
N(x) ≠ N(y)
Because of the bounds



Applicative example : Correction of blurred images

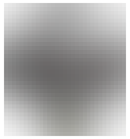
Proposed images :

1) 11 X 11 noiseless picture (for debugging), detail of picture 2



2) 62 X 65 noiseless picture

3) 31 X 34 0.1 % noised picture



4) 16 X 15 0.1 % noised picture (for calibrating the NL solver)

5) 16 X 15 noiseless picture the same as the previous one

Applicative example : Correction of blurred images

Proposed solvers :

1) Tikhonov regularization

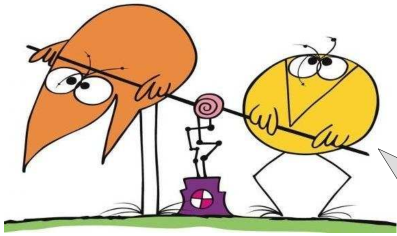
2) Total Variation regularization (non-quadratic Tikhonov)

3) Early-stopped Krylov solver

4) Truncated SVD

As it is, there are
too much unknwowns to do
Bayesian inversion

Unless we try to do
weird things that are
out of the scope
for today



And also out
of the scope
of competences
of the guy who
coded all
the stuff

And it’s not an
overdetermined system
So, no relaxation

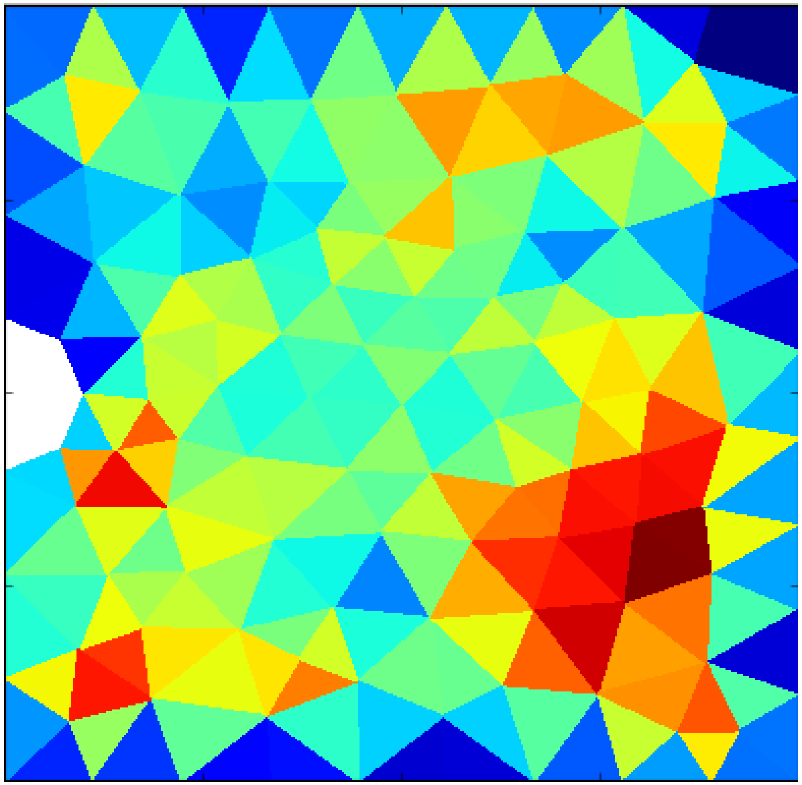
Applicative example : Correction of blurred images

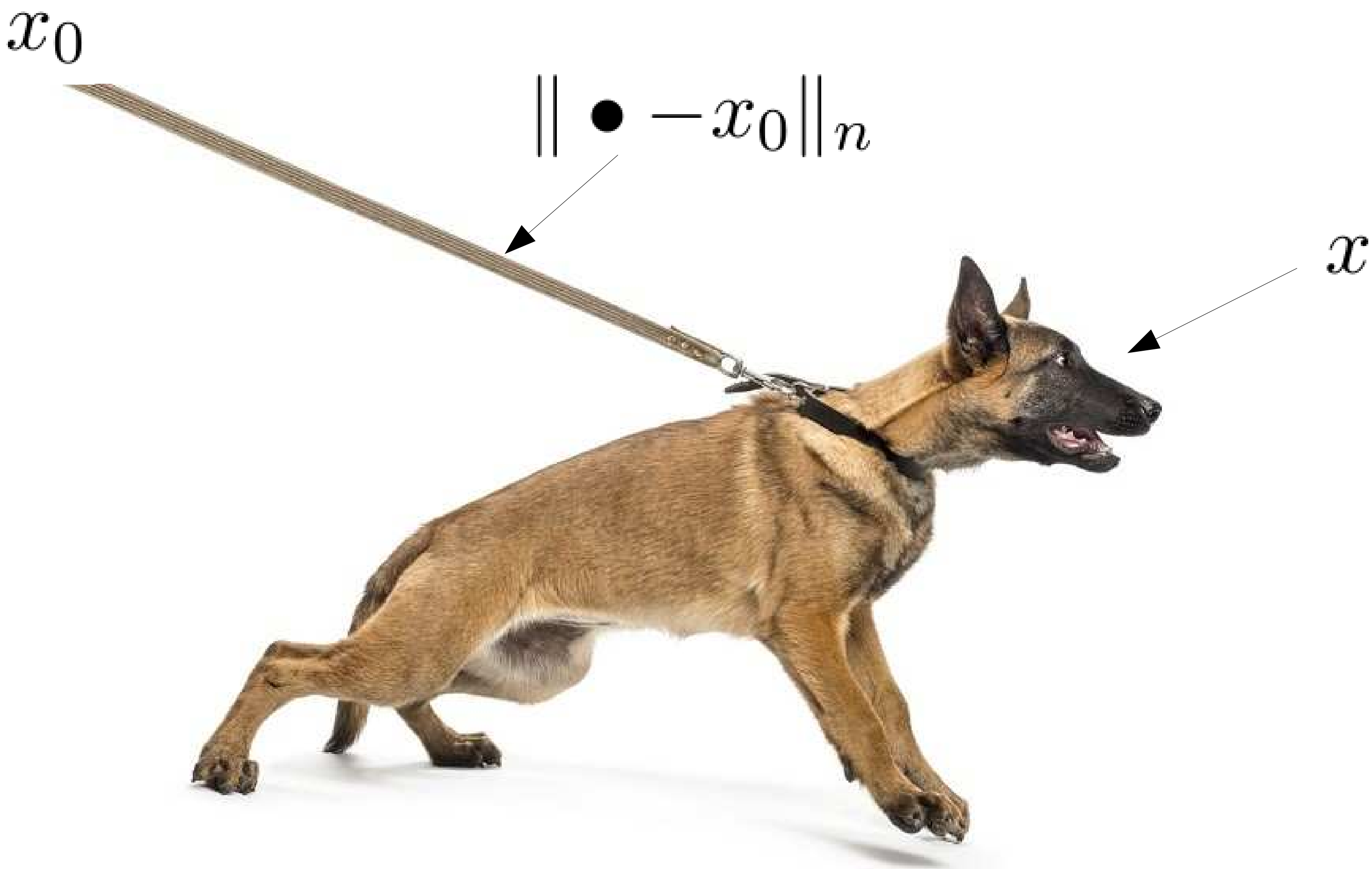
Suggestion of funny things to do (1/2) :	Fun points
1) For Tikhonov : compare the solutions of A'Ax = A'b and Ax = b	🫏 ½🫏
2) For all : Compare the deblurring from a picture and from a matrix Is it different ? Why ?	🫏
3) Test the Morozov method for noisy data $\|Ax-b\|^2=\|\delta b\|^2$	🫏🫏
4) For the picture etnono.png, understand why the blue is behaving so strange	½🫏
5) See what happends when the operator is modified (the model is false)	🫏🫏 ½🫏
6) For Krylov, try to use a congugate gradient on the normal equation. Compare it on the quality and CPU point of view.	🫏🫏
6 1/2) If you really want to compare, implement a MRHS CG	💀

Applicative example : Correction of blurred images

Suggestion of funny things to do (2/2) :	Fun points
7) Try using the data from a .png and compare with the .mat What is the difference ?	🐴
8) Modify the truncated SVD method to make a damped SVD Compare the result with Tikhonov	🐴🐴
9) On TSVD with the image 2-cr, does the function findPicard really choose the best number of modes ? Understand what's wrong and cure the function	🐴🐴💀
10) Why does the total variation tend to give better results than quadratic Tikhonov ?	🐴
11) By inverting the Morozov principle, estimate an equivalent noise level for unnoised pictures. Is it stable between the pictures ? Do the same for the normal equation	🐴🐴🐴

Thank you for your attention




x_0
$\| \bullet - x_0 \|_n$
x

You

The jury of your PhD

An ill-conditionned matrix

The regularization toolbox

The guy who has stupid questions, but you have to answer because he is part of the jury

Your parents : they don‘t understand anythong, but look happy