→ Two problems remain:

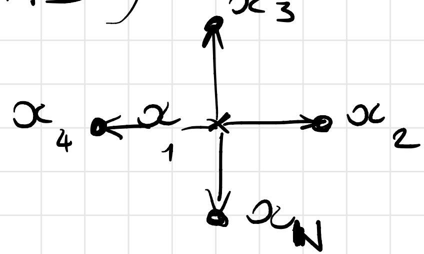
Q1 The cost of the first interpolation model

Q2 The possible non poised system that can be obtained

→ Answer to question 1:

Instead of taking and evaluating $p=\frac{(n+1)(n+2)}{2}$ point for the construction of the first Lagrange polynomial, a reduced number of points is chosen,

$n+1 \le N < p$ (often $N = 2n+1$)



The coefficients of the quadratic model:

$$\boxed{m(x+h) = c + \langle g, h\rangle + \frac{1}{2}\langle h, Hh\rangle}$$

$$p = 1 + n + \frac{n(n+1)}{2}$$

are imposed by:

→ first, an interpolation principle at N points: $f(x_i) = m(x_i)\ \forall i \in \{1,\dots,n\}$

→ m is minimal in the following sense:

$$|||H||| = \operatorname*{Inf}_{(c',g',H')} |||H'|||$$

with $\left|\begin{array}{l} m'(x+h) = c' + \langle g', h\rangle + \langle H'h, h\rangle \\ m' \text{ is interpolation polynom at the } N \text{ chosen points}\end{array}\right.$

and where $|||\ \ |||$ is the Frobenius norm of H:

$$|||H||| = \sum_{i,j} H_{ij}^2$$

(Least square problem with linear constraints)

→ Answer to question 2

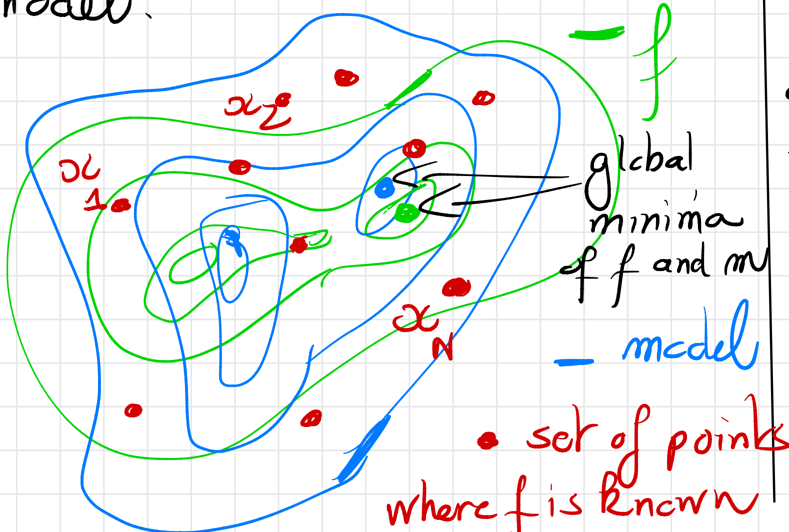
In order to have a poised set at each iteration, some steps are added to improve the geometry (or poisedness) of the interpolation set.

(see reference: Toint et al, 2010)

With all these tools, a very efficient DFO trust region method has been developped: NEWUOA (Powell) (extension BOBYQA for constrained problems)

2.5) Surrogate-based methods

The idea is to replace the cost function f, by a global approximation (or surrogate) model and to optimize this model.



The new model will have to be:

- → easy to compute
- → smooth

Lagrange interpolation is not adapted to this context.

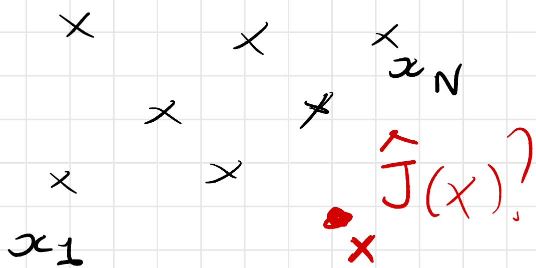
We assume that f is known on a set of N points $\{x_1, \dots, x_N\}$ and we try to build a model m:

↗ Kriging

↘ Radial Basis Function (RBF)

* Kriging

Origin (1950'): interpolation principle developped by Daniel Krige, south african mining engineer.



→ $(x_i, J(x_i))_{1 \leq i \leq n}$ known

→ Build an interpolation function $\hat{J}$ such that

$\forall i \in \{1, \dots N\}, \hat{J}(x_i) = J(x_i)$

The idea is to consider that J and $\hat{J}$ are random variables, with realizations at points x_i: $j(x_i)$ and $\hat{j}(x_i)$.

We assume that $\hat{j}$ (and $\hat{J}$) are linear functions of $(j(x_i))$ (and $J(x_i)$):

$$\boxed{\hat{j}(x) = \sum_{i=1}^{N} w_i(x) j(X_i)}$$

and that the covariance between $J(X)$ and $J(Y)$ is a known function:

$$cov(J(X), J(Y)) = \underbrace{c}_{\text{known}}(X, Y)$$

The coefficients $(\omega_i)_{1 \leq i \leq N}$ are obtained by minimizing : $Var(J(x)-\hat{J}(x))$ with the condition : $E(J(x)-\hat{J}(x)) = 0$

(exercice)

The coefficients are solution of the following linear system :

$$\boxed{C\begin{pmatrix}\omega_1\\ \vdots\\ \omega_N\end{pmatrix} = K}$$

where $C = [c(x_i, x_j)]$ (covariance matrix)

and $K = \begin{pmatrix} c(x_1,x)\\ \vdots \\ c(x_N,x)\end{pmatrix}$

Thus :

$$\boxed{\hat{J}(x) = {}^tK\, C^{-1} Z}$$

with $Z = \begin{pmatrix} J(x_1)\\ \vdots \\ J(x_N)\end{pmatrix}$.

Moreover,

$$V(\hat{J}(x) - J(x)) = c(x,x) - {}^tK\, C^{-1} K$$

We first notice that

$$\hat{J}(x_j) = \sum_{i=1}^{N} \delta_{ij}\, J(x_i) = J(x_j)$$

$\longrightarrow$ interpolation model.

The remaining part is to find the best covariance function f:
we first assume that c has a Gaussian type form:

$$\boxed{c(X,Y) = \theta_1 \exp\left(-\frac{1}{2}\sum_{i=1}^{n}\frac{(X_i-Y_i)^2}{r_i^2}\right)+\theta_2}$$

if $\mathbf{X} = (X_1, \dots X_n)$ and $Y = (Y_1, \dots Y_n)$

The parameters $(\theta_1, \theta_2, r_1, \dots r_n)$ are fixed with a maximum likelyhood principle, by maximizing

$$\boxed{\mathcal{L}(\theta_1, \dots r_n) = \frac{1}{\sqrt{(2\pi)^N \det C}} \exp\left(-\frac{1}{2} z^t C z\right)}$$

→ Implementation with Scilab //

→ Include Kriging into optimization algorithm: iteratively by improving the model at each step with a new point: DACE
EGO
⋮

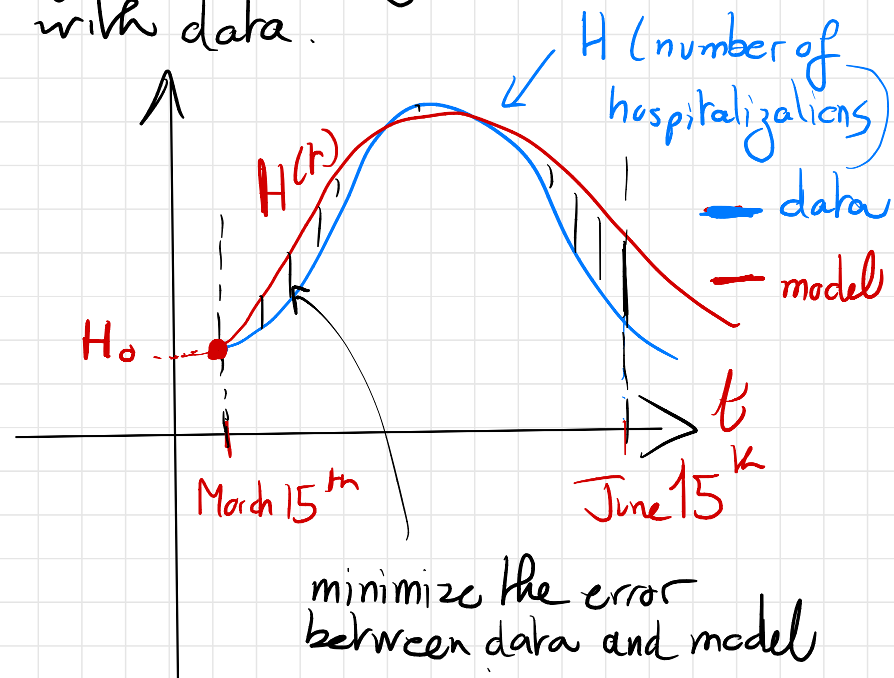
→ 12/02 (last course)

* Small quizz 2 (must region methods and surrogate models)

* Applicative example:

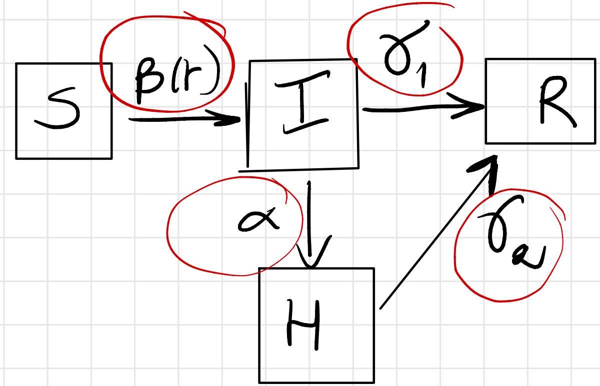
Study the first wave of COVID epidemy in various French regions (between 15-03 and 15-06)

→ with a simple SIHR model after optimizing its parameters with data.



→ Model SIHR

$$\begin{cases} S(t) : \text{number of susceptible} \\ I(t) : \quad \text{"} \quad \text{infected} \\ H(t) \quad \text{"} \quad \text{hospitalized} \\ R(t) \quad \text{"} \quad \text{recovered (or deceased)} \end{cases}$$



$$\begin{cases} \dfrac{dS(t)}{dt} = -\beta(t) I(t) S(t) \\[2ex] \dfrac{dI(t)}{dt} = \beta(t) I(t) S(t) - \alpha I(t) - \gamma_1 I(t) \\[2ex] \dfrac{dH(t)}{dt} = \alpha I(t) - \gamma_2 H(t) \\[2ex] \dfrac{dR(t)}{dt} = \gamma_1 I(t) + \gamma_2 H(t) \end{cases}$$

with initial condition $(S(0), I(0), H(0), R(0))$

* Parameters :

$$\begin{cases} * \ \beta(t) = C \exp(-\lambda t) \quad \text{(decreasing with } t) \\ \left.\begin{array}{l} * \ \alpha \\ * \ \gamma_1 \\ * \ \gamma_2 \end{array}\right\} \text{constants} \end{cases}$$

* Function to minimize :

$$(c, \lambda, \alpha, \gamma_1, \gamma_2) \xrightarrow{J} \|H_{model} - H_{data}\|$$

→ No Gradient information !

→ Use of a DFO optimization method

For Next course :

Part 1 → Implement function J in a language you can choose /

individually or in groups /

→ The data are different for each person (*)

Part 2 → Minimize J with one or two methods of your choice (CCMAES, NEWUOA, Nelder Mead, Pattern Search), written or available on the web. //

Part 3 → Give the values of $\beta, \alpha, \gamma_1, \gamma_2$ for the studied region of France //

→ Auvergne : Baudry, Yang
→ Bourgogne : Boutin, El Herichi
→ Grand Est : Castera, Sun
→ Hauts de France : El Aichi, Gao
→ Ile de France : Martinez, Levillain
→ Provence : NGuyen, Ben Ghalleb

* Data available at the adress :
data.gouv.fr/
(Donnés relatives à la COVID-19)

* Remarks :

→ The model can be adimensionalized :

$$S + I + H + R = 1$$

→ Initialization :

* $H(0)$ with the data (H_0/N) fraction of people
* $I(0) = 10\, H_0/N$
* $S(0) \approx 1$
* $R(0) \approx 0$

→ Approximations of the parameters :

* β is decreasing such that :

$$R_0 = \frac{\beta}{\alpha + \gamma_1}$$: goes from 3 to 0.6
(March 15) (May 15)

* $\gamma_1 \approx \frac{1}{12}$ (12 days of infection)
* $\alpha \approx \frac{1}{10}$ ($\in [0,1]$)
* $\gamma_2 \approx \frac{1}{10}$ ($\in [0,1]$)

(rough values, as starting point for optimization)