Société de Calcul Mathématique, S. A.
Algorithmes et Optimisation

L'Hypersurface Probabiliste

Algorithme de recherche de zones particulières
- Looking for dangerous zones-

Rapport adressé à

l'Institut de Radioprotection et de Sûreté Nucléaire

Direction de la Sûreté des Réacteurs

par la

Société de Calcul Mathématique SA

en application de votre commande DA 79160/CA 11032655 du 4 mars 2008

Novembre 2008

Siège social et bureaux : 111, Faubourg Saint Honoré, 75008 Paris. Tel. : 01 42 89 10 89. Fax : 01 42 89 10 69
Société Anonyme au capital de 56 200 Euros. RCS : Paris B 399 991 041. SIRET : 399 991 041 00035. APE : 7219Z

Résumé opérationnel

Ce rapport est consacré à la recherche de zones dangereuses : ce sont les zones de l'espace des configurations où la température est susceptible d'être élevée (c'est à dire de dépasser un seuil fixé à l'avance). La connaissance préalable de cette information, au moyen de l'Hypersurface Probabiliste, est très importante : à partir d'un petit nombre de "runs" du code de calcul, l'EPH permet de diagnostiquer les zones à risque ; ces zones étant caractérisées, on pourra concentrer les runs suivants du code de calcul pour les explorer. En d'autres termes, l'EPH permet une caractérisation préliminaire, qui fait économiser des runs : on les concentre là où c'est important.

Un point essentiel est le suivant : les zones à risque sont d'abord caractérisées par une propriété mathématique (proximité plus grande des points chauds que des points froids), qui est très semblable à celle des "diagrammes de Voronoï" (voir [2]). Mais ces diagrammes ne peuvent pas être construits, de manière explicite, si la dimension d est grande et le nombre de points n important : la complexité est en $n^{d/2}$.

Pour trouver les points situés à distance maximale d'un point chaud donné, à l'intérieur d'un diagramme, nous sommes donc amenés à réaliser une construction de nature probabiliste : on choisit une direction au hasard, et on mesure la distance réalisée, en restant dans l'hypercube.

Mais cette notion de "choix de direction au hasard" n'est nullement une évidence : le choix dépend de la norme qui est retenue. Dans un premier temps, nous montrons quelles erreurs sont fréquemment commises (le problème est très général) ; nous donnons ensuite la construction correcte pour la loi de probabilité, et nous l'appliquons à la recherche du point le plus éloigné. Nous caractérisons alors les températures obtenues.

Une réitération de l'algorithme de recherche est très intéressante en soi : la seconde recherche se fait à l'intérieur d'un hyperplan ; la troisième dans un espace de codimension 2, et ainsi de suite. Nous montrons comment les algorithmes doivent être adaptés.

La conclusion numérique de la présente étude est que, avec la valeur actuelle des différents paramètres, la probabilité de dépasser 960°C est nulle, quelle que soit la configuration. Il serait donc légitime d'augmenter la puissance, tout en restant au dessous des seuils de sûreté.

Executive Summary

This report is devoted to the search for dangerous zones : these are the zones, in the configuration space, where the temperature may be high (that is, above a certain threshold). The preliminary knowledge, for this information, using the EPH, is quite interesting : from a small number of runs of the computer code, the EPH allows us to find the dangerous zones. Once these zones are characterized, one may concentrate further runs of the code inside them. In other terms, the EPH allows a preliminary characterization, which will save time and efforts, in terms of number of runs : they will be concentrated where the true need is.

An essential point is as follows : the dangerous zones are first characterized by a mathematical property (greater proximity to hot points than to cold points), which is quite analogous to Voronoï diagrams (see [2]). But these diagrams cannot be explicitly constructed if the dimension d is high and the number of points n is large, since the complexity is in $n^{d/2}$.

In order to find the points situated at maximal distance to a given hot point, inside a diagram, we have to use a construction of probabilistic nature : we choose a random direction and we measure the distance attained, still remaining inside the hypercube.

But this notion of "random direction" is by no means obvious : it depends upon the norm which has been chosen. In a first part, we show what mistakes occur frequently (this is a fairly general problem) ; then we give the correct construction for the probability law, and we apply this law to the search for the farthest point. We finally characterize the temperatures which have been obtained.

The search algorithm may be iterated ; the second step occurs inside an hyperplane, the third inside a space of codimension 2, and so on. We show what modifications must be brought to the algorithms in order to execute further steps.

The numerical conclusion of the present study is that the probability to be above the threshold of 960°C is zero, for any possible configuration of the parameters. Therefore, it might be possible to increase the power of the reactor, still remaining below the safety threshold.

Table of Contents

In this report, we investigate the following question : how can we find "sensitive" zones in the EPH ? "sensitive" may mean for instance zones with high temperatures.

I. Presentation of the problem

In our previous report [1], we showed that the zones with high temperature must be "around" the measure points with high temperature. In fact, in a high dimensional space, the influence sent by each measure point depends only on the distance : take any point in the space, it receives its strongest influence from the closest point.

So, this leads to the concept of "Voronoi Diagram" (see for instance [2]). The Voronoi Diagram built upon a set of points A_k is a partition of the space ; each diagram D_k is a simplex (intersection of halfspaces) consisting in the points which are closer to A_k than to all other A_j.

However, the full construction of the Voronoi Diagram is impossible in high dimension : it requires a number of operations of $N^{K/2}$ (N is the number of points, and K is the dimension ; here $K=42$).

In order to present more suitable algorithms, let us first state the problem precisely.

We start with one point, the measure point with highest temperature. Let us call it H_0. In our case, this point is the 175th, and it gives the value 955.76°C. We have set a threshold $T_0 = 900°C$ and we have 307 measure points below this value. Let us call them C_m, $m=1,...,M$, with $M=307$ (H stands for hot, C for cold).

Our problem is as follows :

Find the point X in the space $\left[0,1\right]^K$ such that :

$$d(X,H_0) \text{ is maximal} \tag{1}$$

satisfing the constraints :

$$d(H_0,X) \le d(C_m,X) \text{ for all } m=1,...,M \tag{2}$$

Here, d denotes the usual euclidean distance. In fact, we will always work with the square of the distance, d^2, so this is a convex problem, with convex constraints. Some of them are linear : since the point X is in the space $\left[0,1\right]^K$, all its coordinates x_k must satisfy :

$$0 \le x_k \le 1,\ k=1,...,K. \tag{3}$$

In fact, as we already saw in our report [1], all constraints may be taken to be linear. Let us introduce the hyperplane E_m which is equidistant between H_0 and C_m. Let

$f_m(X) = e_m$ be the equation of this hyperplane, then the constraints (2) may be replaced by the set of linear constraints :

$$f_m(X) \geq e_m \text{ or } f_m(X) \leq e_m \text{ for all } m = 1, \dots, M \tag{4}$$

The constraints $0 \leq x_k \leq 1$ are of the same type ; they mean that the requested point must lie in some half space. So we have the following problem, of geometrical nature : find the point which is farthest from a given point, inside the intersection of a large number of half spaces. Here, we have 307 + 84 = 391 half spaces.

Of course, the solution occurs when the constraints are "saturated", which means that some of them become equalities. In a space of dimension $K = 42$, the intersection of K hyperplanes is a single point. So the solution should be sought among the intersections of 42 hyperplanes. But there are $\binom{391}{42}$ such intersections : this is a huge number : 5390837042940093792377338042827426351941584498933558620000, and these intersections cannot be enumerated in practice.

The problem : find the farthest point from a given point, in a simplex, is not feasable in practice, in high dimension, from a deterministic and computational point of view. We will need random algorithms.

Let us finally observe that the solution of problem (1) (2) (3) above (the farthest point) is not exactly the solution to our problem of highest temperature. Let X_0 be the solution of (1) (2) (3). Then X_0 will probably be on some boundary, which means that for some m we will have $d(H_0, X_0) = d(C_m, X_0)$; in other words, X_0 will receive the same influence from H_0 and from the cold point C_m. So we will have to "backup" a little bit, and the solution of our problem of high temperature will be a point X_1, in between H_0 and X_0, and very close to X_0. In other words : $\overrightarrow{H_0X_1} = \alpha \overrightarrow{H_0X_0}$, for some α, $0 < \alpha < 1$, very close to 1. The precise value of α will be determined experimentally. We now concentrate upon the search of X_0.

II. Random search

So, we will use a random search algorithm ; this algorithm will produce random directions, starting from the point H_0, and each of these directions will be explored in turn, in order to find the farthest point.

Since we have no a priori knowledge about the directions (none is privileged), we need a uniform algorithm, that is an algorithm which returns all directions with a uniform law. This is not as easy as it seems, as we now see.

A "direction" is defined by the choice of a point on the unit sphere, for the euclidean norm, centered at the point H_0. So we need to construct a uniform law on the unit sphere.

A. Preliminary warning

If you take a uniform law on the components of a vector, and then normalize the vector, you do not get a uniform law on the normalized vector.

Quite often, people do the following :

- They draw a sample for a random vector $(x_1,...,x_N)$; each x_n follows a uniform law, for instance on the interval $[0,1]$;

- And then they normalize, that is they replace each x_n by

$$y_n = \frac{x_n}{\sqrt{\sum_{i=1}^{N} x_i^2}} \text{, so that indeed } \sum_{n=1}^{N} y_n^2 = 1.$$

People often think that, if you take the vector $X = (x_1,...,x_N)$ with a uniform law, and then normalize it, the normalized vector $Y = (y_1,...,y_n)$ will have a uniform law on the unit sphere, or, if one prefers, that all portions of the unit sphere have the same probability. This is wrong.

In mathematical terms, the radial projection of the uniform law on the unit hypercube is not the uniform law on the unit sphere.

Let us see this on a very simple example, dimension 2.

One takes two random numbers x_1 and x_2 with uniform law between 0 and 1. This means that we have a uniform law in the unit square $[0,1]^2$.

For each x_1 and x_2, we consider the normalized vector :

$$y_1 = \frac{x_1}{\sqrt{x_1^2 + x_2^2}}$$

$$y_2 = \frac{x_2}{\sqrt{x_1^2 + x_2^2}}$$

and we want to find the law of this vector, which is on the unit circle.

Fix any θ, $0 \le \theta \le \frac{\pi}{2}$:

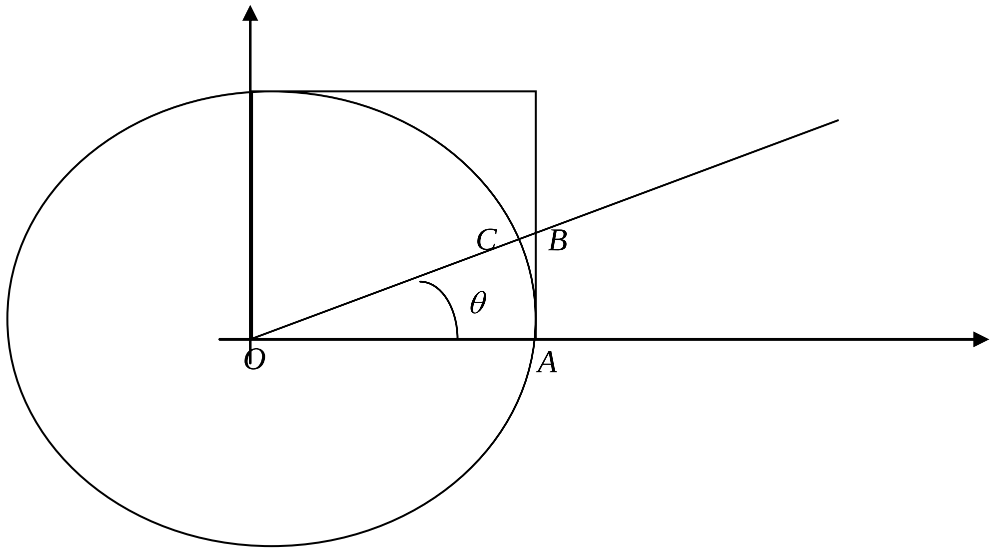


Figure 1 : Uniform law in a square and on a circle

The point Y is on the arc AC if and only if the point X is inside the triangle OAB. The area of this triangle is :

$$\text{If } 0 \le \theta \le \frac{\pi}{4},\ area(OAB) = \frac{\tan(\theta)}{2}$$

$$\text{If } \frac{\pi}{4} \le \theta \le \frac{\pi}{2},\ area(OAB) = 1 - \frac{\tan(\frac{\pi}{2} - \theta)}{2} = 1 - \frac{1}{2\tan(\theta)}$$

Let $f(\theta)$ be the density function of the vector Y and $F(\theta)$ its repartition function. We get :

$$\text{If } 0 \le \theta \le \frac{\pi}{4},\ F(\theta) = \frac{\tan(\theta)}{2}$$

$$\text{If } \frac{\pi}{4} \le \theta \le \frac{\pi}{2},\ F(\theta) = 1 - \frac{1}{2\tan(\theta)}$$

Since $f(\theta) = F'(\theta)$, we obtain :

$$\text{If } 0 \le \theta \le \frac{\pi}{4},\ f(\theta) = \frac{1}{2\cos^2(\theta)}$$

$$\text{If } \frac{\pi}{4} \le \theta \le \frac{\pi}{2},\ f(\theta) = \frac{1}{2\sin^2(\theta)}$$

Here is the graph of this density of probability :

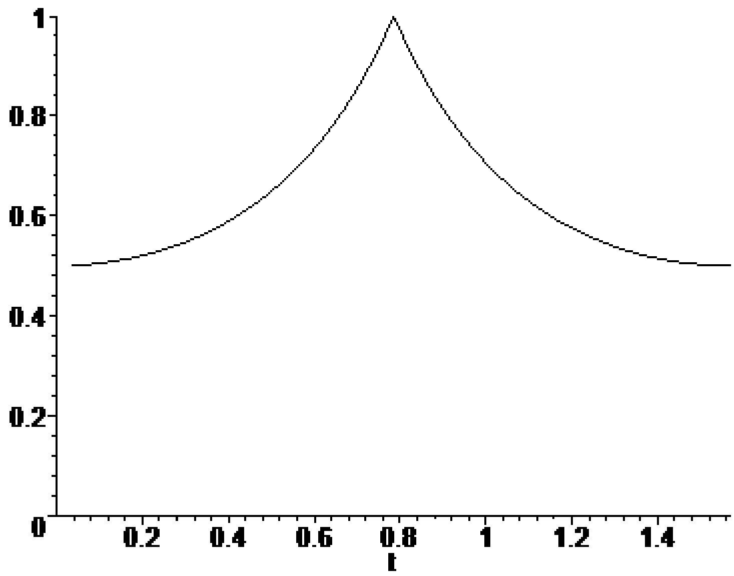


Figure 2 : Graph of the projected density of probability

We see that it is not at all uniform !

Let us compute the probability of the intervals

$$I_1 = \left\{ 0 \le \theta \le \frac{\pi}{10} \right\}$$

$$I_2 = \left\{ \frac{2\pi}{10} \le \theta \le \frac{3\pi}{10} \right\}$$

Both have the same width, namely $\frac{\pi}{10}$, and the second one is centered around $\frac{\pi}{4}$. We get:

$$P\{I_1\} = F\left(\frac{\pi}{10}\right) = \frac{\tan(\pi/10)}{2} \approx 0.1625$$

$$\begin{aligned} P\{I_2\} &= F\left(\frac{\pi}{4} + \frac{\pi}{20}\right) - F\left(\frac{\pi}{4} - \frac{\pi}{20}\right) \\ &= 2\left(F\left(\frac{\pi}{4}\right) - F\left(\frac{\pi}{4} - \frac{\pi}{20}\right)\right) \\ &= 1 - \tan\left(\frac{\pi}{4} - \frac{\pi}{10}\right) \approx 0.4905 \end{aligned}$$

So there is a considerable difference between both probabilities, despite the fact that both intervals have same width.

The same mistake would occur if we would project upon a hyperplane, instead of the unit circle, and would occur as well with other laws (not uniform) : the ratio between the area of the sector (on the circle or on a hyperplane) and the area of the triangle is not constant.

We would have the same warning, or mistake, for the l_1 norm, that is $\sum |x_i|$:

To generate uniform variables X_k on $[0,1]$ and then replace them by $\dfrac{X_k}{\sum_{j=1}^{K} X_j}$ is wrong, because we do not obtain this way a uniform law on the unit sphere. This is clear on the following example (dimension 2) :

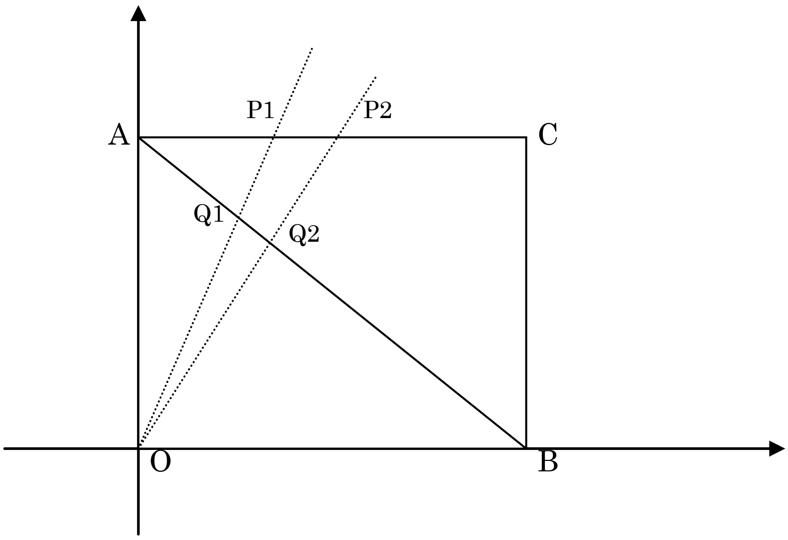


Figure 3 : projection of the uniform law in the square to the segment

Assume we take random variables (X_1, X_2) with uniform law on the segment $[0,1]$. Then, the number of times the point with coordinates (X_1, X_2) falls into the triangle OP_1P_2 is proportional to the area of that triangle. But the normalized point $\dfrac{X_1}{X_1+X_2}, \dfrac{X_2}{X_1+X_2}$ is in the segment $Q_1 - Q_2$ if and only if the point (X_1, X_2) is in the triangle OP_1P_2. But the area of the triangle depends upon its orientation : for given length of $Q_1 - Q_2$, we have a larger triangle if it contains the point C, and smaller at the extremities, when the triangle contains A or B. So this construction does not lead to a uniform law on the segment AB.

These remarks are quite general : a uniform law in the square does not project to a uniform law on the unit sphere.

B. Uniform distribution on the unit sphere

In this paragraph, we now show how to build a uniform distribution on the unit sphere.

1. The case of the l_1 norm

We start with the simple case of $\mathbb{R}^K$, equipped with the l_1 norm, that is :

$$\|X\| = \sum_{i=1}^{K} |x_i| \quad \text{if } X = (x_i)_{i=1,\dots,K}$$

We will discuss other situations later.

For this case, we follow the book by Luc Devroye [3], chapter 5, theorem 2.2.

Let :

$$C_K = \left\{ (x_1,\dots,x_K);\, x_k \geq 0\ \forall k, \sum_{k=1}^{K} x_k = 1 \right\}$$

be the positive part of the l_1-sphere.

Theorem. - *Let $X_1,\dots,X_K$ be independent random variables with exponential law. Then, the variables $Y_1,\dots,Y_K$ defined by :*

$$Y_1 = \frac{X_1}{X_1+\dots+X_K}, \dots, Y_K = \frac{X_K}{X_1+\dots+X_K}$$

are positive, have sum 1, and follow a uniform law on C_K.

Proof of the Theorem. Let $S = X_1+\dots+X_K$. Let $Z = (X_1,\dots,X_{K-1})$, and $z = (x_1,\dots,x_{K-1})$, to simplify the notation.

Let $f_{Z,S}(z,s)$ be the density of the joint law of the $K-$uple (Z,S). This density can be computed as a conditional probability, knowing Z, that is :

$$f_{Z,S}(z,s) = f_{S|Z}(s) \times f_Z(z) \tag{1}$$

where $f_{S|Z}(s)$ denotes the conditional probability density of S knowing Z and $f_Z(z)$ is the density of Z.

Since the variables $X_1,\dots,X_{K-1}$ are independent and follow an exponential law, we have :

$$f_Z(z) = e^{-(x_1+\dots+x_{K-1})} \tag{2}$$

The law of S knowing $X_1 = x_1,\dots,X_{K-1} = x_{K-1}$ is easy to find. Indeed,

$$P\{S \leq a \mid X_1 = x_1,\dots,X_{K-1} = x_{K-1}\} = P\{X_K \leq a - x_1 - \dots - x_{K-1} \mid X_1 = x_1,\dots,X_{K-1} = x_{K-1}\}$$

and therefore :

$$f_{S|Z}(s) = e^{-s+x_1+\cdots+x_{K-1}} \tag{3}$$

for $s \geq x_1 + \cdots + x_{K-1}$, 0 otherwise.

We deduce from (1), (2), (3) that :

$$f_{Z,S}(z,s) = e^{-s} \tag{4}$$

for $s \geq x_1 + \cdots + x_{K-1}$, 0 otherwise.

Now, we compute the joint density of $\left(\frac{X_1}{S},...,\frac{X_{K-1}}{S},S\right)$. This is obtained by a change of variable in the density (4). We set :

$$y_1 = \frac{x_1}{s},..., y_{K-1} = \frac{x_{K-1}}{s}, s = s .$$

Let $U = \left(\frac{X_1}{S},...,\frac{X_{K-1}}{S}\right)$; we get :

$$f_{U,S}(u,s) = s^K e^{-s} \tag{5}$$

The density of U can be obtained from the above formula, integrating with respect to s. We get :

$$f_U(u) = \int_0^{+\infty} f_{U,S}(u,s)ds = \int_0^{+\infty} s^K e^{-s} ds = K! \tag{6}$$

and we see that this value is constant on the whole set $x_1 \geq 0,..., x_{K-1} \geq 0,\ x_1 + \cdots + x_{K-1} \leq 1$.

But now, since the density of $\frac{X_1}{S},...,\frac{X_{K-1}}{S}$ is constant, so is the density of

$$\left(\frac{X_1}{S},...,\frac{X_{K-1}}{S},1-\frac{X_1}{S}-...-\frac{X_{K-1}}{S}\right) = \left(\frac{X_1}{S},...,\frac{X_{K-1}}{S},\frac{X_K}{S}\right) \tag{7}$$

This concludes the proof of the Theorem.

Remark

A package in Matlab, due to Roger Stafford, generates random numbers with uniform law, and fixed sum : it can be found on the Interned under the name of randfixedsum.m.

This package uses another approach, more complicated than the approach described here (using exponential variables), but it can solve more general problems : find x_i with $a \le x_i \le b$ and $\sum_{i=1}^{n} x_i = S$.

Remark

In order to generate random variables with exponential law, on should generate random variables with uniform law Y_k and then take $X_k = Log \frac{1}{Y_k}$.

2. Application

Random sampling using the l_1 norm is mostly used for sampling of proportions.

So, assume we want to generate a sample of $N = 1000$ proportions for $K = 40$ goods, we do the following :

for $n = 1$ to N
for $k = 1$ to K
generate $Y_1, \ldots, Y_K$ independent variables with uniform law on $[0,1]$
take $X_k = Log \frac{1}{Y_k}$, for $k = 1, \ldots, K$
compute $\sum_{j=1}^{K} X_j$
replace each X_k by $\frac{X_k}{\sum_{j=1}^{K} X_j}$
next k
next n

3. The case of the l_2 norm

Here, the space R^K is equipped with the l_2 norm, that is :

$$\|X\| = \left(\sum_{i=1}^{K} x_i^2 \right)^{1/2} \quad \text{if } X = (x_i)_{i=1,\ldots,K}$$

This is the Euclidean norm, which is the one we are using in the construction of the Hypersurface.

We have a similar result :

Theorem 2. *-Let* $K \geq 1$ *and let* $X_1,...,X_K$ *be independent normal variables (mean 0, variance 1). Then the variables :*

$$\frac{X_1}{\sqrt{X_1^2+\cdots+X_K^2}},\ldots,\frac{X_K}{\sqrt{X_1^2+\cdots+X_K^2}}$$

are independent and follow a uniform law on the unit sphere of the $K-$*dimensional Euclidean space.*

In other words, in order to obtain a uniform law on the sphere, one should not start with uniform variables, but with Gaussian.

A proof of this result can be found in the book by R. J. Muirhead, page 37 (communicated by Paul Deheuvels). This proof uses the fact that the normal law on R^K is invariant under all orthogonal transformations of the space into itself. So, its projection upon the unit sphere is itself invariant under the projections of the transformations. The unit sphere is a locally compact group for these transformations, and a general result says that there is only one invariant Haar measure, which must be the uniform law.

However, we will give a specific proof, following the arguments used in the l_1 case.

Proof of Theorem 2

Let now $S=\sqrt{X_1^2+\cdots+X_K^2}$, $Z=\left(X_1,...,X_{K-1}\right)$, and $z=\left(x_1,...,x_{K-1}\right)$.

Let $f_{Z,S}(z,s)$ be the density of the joint law of the $K-$uple $\left(Z,S\right)$. This density can be computed as a conditional probability, knowing Z , that is :

$$f_{Z,S}(z,s)=f_{S|Z}(s)\times f_Z(z) \tag{1}$$

where $f_{S|Z}(s)$ denotes the conditional probability density of S knowing Z and $f_Z(z)$ is the density of Z .

Since the variables $X_1,...,X_{K-1}$ are independent and follow a normal law, we have :

$$f_Z(z)=e^{-\frac{1}{2}x_1^2}\times...\times e^{-\frac{1}{2}x_{K-1}^2} \tag{2}$$

The law of S knowing $X_1=x_1,...,X_{K-1}=x_{K-1}$ is easy to find. Indeed,

$$P\left\{S\leq a\,|\,X_1=x_1,...,X_{K-1}=x_{K-1}\right\}=P\left\{X_K\leq a^2-x_1^2-...-x_{K-1}^2\,|\,X_1=x_1,...,X_{K-1}=x_{K-1}\right\}$$

and therefore :

$$f_{S|Z}(s)=e^{-\frac{1}{2}(s^2-x_1^2-\cdots-x_{K-1}^2)} \tag{3}$$

for $s\geq x_1+\cdots+x_{K-1}$, 0 otherwise.

We deduce from (1), (2), (3) that :

$$f_{Z,S}(z,s) = e^{-\frac{1}{2}s^2} \tag{4}$$

for $s \geq x_1 + \cdots + x_{K-1}$, 0 otherwise. We see that this is independent of $z = \left(x_1, ..., x_{K-1}\right)$ and the rest of the proof goes as before.

We compute the joint density of $\left(\frac{X_1}{S}, ..., \frac{X_{K-1}}{S}, S\right)$. This is obtained by a change of variable in the density (4). We set :

$$y_1 = \frac{x_1}{s}, ..., y_{K-1} = \frac{x_{K-1}}{s}, s = s\,.$$

Let $U = \left(\frac{X_1}{S}, ..., \frac{X_{K-1}}{S}\right)$; we get :

$$f_{U,S}(u,s) = s^K e^{-\frac{1}{2}s^2} \tag{5}$$

The density of U can be obtained from the above formula, integrating with respect to s. We get :

$$f_U(u) = \int_0^{+\infty} f_{U,S}(u,s)ds = \int_0^{+\infty} s^K e^{-\frac{1}{2}s^2} ds \tag{6}$$

and we see that this value is constant on the whole set $x_1 \geq 0, ..., x_{K-1} \geq 0, x_1 + \cdots + x_{K-1} \leq 1$.

But now, since the density of $\frac{X_1}{S}, ..., \frac{X_{K-1}}{S}$ is constant, so is the density of

$$\left(\frac{X_1}{S}, ..., \frac{X_{K-1}}{S}, 1 - \frac{X_1}{S} - ... - \frac{X_{K-1}}{S}\right) = \left(\frac{X_1}{S}, ..., \frac{X_{K-1}}{S}, \frac{X_K}{S}\right) \tag{7}$$

This concludes the proof of Theorem 2.

We observe that the same result holds for all the l_p norms, defined by :

$$\|X\| = \left(\sum_{i=1}^{K} x_i^p\right)^{1/p} \quad \text{if } X = \left(x_i\right)_{i=1,...,K}, \text{ for } 1 \leq p \leq \infty$$

We would take :

$$S = \left(\sum_{i=1}^{K} X_i^p\right)^{1/p}$$

4. Case of a general norm

Let us now investigate the general case of a norm on R^K. Of course, this norm has to be invariant under permutation of the variables :

$$\left\|\left(x_{\sigma(1)},...,x_{\sigma(K)}\right)\right\|=\left\|\left(x_1,...,x_K\right)\right\|,$$

for any permutation σ of the set $\{1,...,K\}$.

The question is : can we find a probability law, given by its density f or by the repartition function F such that, if $X_1,...,X_K$ are independent variables following this law, the law of $\dfrac{\left(X_1,...,X_K\right)}{\left\|\left(X_1,...,X_K\right)\right\|}$ will be the uniform law on the unit sphere of $\mathbb{R}^K$, equipped with this norm ?

The answer is "yes", under a restriction for the norm, which is of geometrical nature. In order to explicit this condition, let us define a function $\varphi\left(s\,;x_1,...,x_{K-1}\right)$ as follows :

The condition

$$\left\|\left(x_1,...,x_{K-1},x_K\right)\right\|\leq s$$

is equivalent to the fact :

$$0\leq x_K\leq\varphi\left(s\,;x_1,...,x_{K-1}\right).$$

For instance, for the Euclidean norm,

$$\varphi\left(s\,;x_1,...,x_{K-1}\right)=\left(s^2-x_1^2-...-x_{K-1}^2\right)^{1/2}$$

Theorem 3. - *Let $\|\ \|$ be a norm on $\mathbb{R}^K$, invariant under permutation of the variables. Then, there is a probability density on $\mathbb{R}$ such that, if $X_1,...,X_K$ are independent variables following this law, the law of $\dfrac{\left(X_1,...,X_K\right)}{\left\|\left(X_1,...,X_K\right)\right\|}$ is the uniform law on the unit sphere of $\mathbb{R}^K$, equipped with this norm, if and only if the values of $\dfrac{\partial\varphi}{\partial x_1}\left(s\,;z\right)$ depend only on the values of x_1 and of $\varphi\left(s\,;z\right)$. In other words, the answer is positive if and only if there is a function G such that :*

$$\frac{\partial\varphi}{\partial x_1}\left(s\,;z\right)=G\left(x_1,\varphi\left(s\,;z\right)\right)\qquad (*)$$

Proof of Theorem 3.

We set :

$$S = \left\| \left(X_1, ..., X_K \right) \right\|$$

and, as previously,

$$Z = \left(X_1, ..., X_{K-1} \right), \text{ and } z = \left(x_1, ..., x_{K-1} \right)$$

We compute the joint law of the couple (Z, S). As before, we have :

$$f_{Z,S}(z,s) = f_{S|Z}(s) \times f_Z(z)$$

If we can obtain :

$$f_{Z,S}(z,s) = f_{S|Z}(s) \times f_Z(z) = c(s)$$

independent of $z = (x_1, ..., x_{K-1})$, then the same argument as before will show that we have a uniform law on the unit sphere.

So we are looking for an identity of the following form :

$$f_{S|Z}(s) \times f_Z(z) = c(s) \tag{1}$$

independent of $z = (x_1, ..., x_{K-1})$.

We have :

$$f_Z(z) = f(x_1) \times \cdots \times f(x_{K-1})$$

and

$$F_{S|Z}(s) = P\left\{ \left\| \left(X_1, ..., X_K \right) \right\| \le s \mid X_1 = x_1, ..., X_{K-1} = x_{k-1} \right\}$$

that is :

$$F_{S|Z}(s) = P\left\{ \left\| \left(x_1, ..., x_{K-1}, X_K \right) \right\| \le s \right\}$$

This probability is entirely determined by the law on X_K, since the norm is known and the values of $x_1, ..., x_{K-1}$ are fixed.

The condition :

$$\left\| \left(x_1, ..., x_{K-1}, X_K \right) \right\| \le s$$

is equivalent to the fact that X_K is in the interval :

$$0 \leq X_K \leq \varphi(s\,;x_1,\ldots,x_{K-1})$$

by definition of the function φ.

For instance, for the Euclidean norm, the condition

$$\|(x_1,\ldots,x_{K-1},X_K)\| \leq s$$

is equivalent to :

$$0 \leq X_K \leq \left(s^2 - x_1^2 - \ldots - x_{K-1}^2\right)^{1/2}$$

that is, we have :

$$\varphi(s\,;x_1,\ldots,x_{K-1}) = \left(s^2 - x_1^2 - \ldots - x_{K-1}^2\right)^{1/2}$$

So the identity (1) is equivalent to :

$$f\left(\varphi(s\,;x_1,\ldots,x_{K-1})\right) \times f(x_1) \times \cdots \times f(x_{K-1}) = c(s) \tag{2}$$

independent of $(x_1,\ldots,x_{K-1})$.

Since all variables are equivalent, it is enough to ensure this for x_1. Computing the derivative with respect to this variable, we get :

$$\begin{aligned} & f'\left(\varphi(s\,;x_1,\ldots,x_{K-1})\right)\frac{\partial\varphi}{\partial x_1}(s\,;x_1,\ldots,x_{K-1}) \times f(x_1) \times \cdots \times f(x_{K-1}) + \\ & \quad + f\left(\varphi(s\,;x_1,\ldots,x_{K-1})\right) \times f'(x_1) \times f(x_2) \times \cdots \times f(x_{K-1}) = 0 \end{aligned}$$

that is,

$$f'\left(\varphi(s\,;x_1,\ldots,x_{K-1})\right)\frac{\partial\varphi}{\partial x_1}(s\,;x_1,\ldots,x_{K-1}) \times f(x_1) = -f\left(\varphi(s\,;x_1,\ldots,x_{K-1})\right) \times f'(x_1) \tag{3}$$

This can be written :

$$\frac{f'\left(\varphi(s\,;z)\right)}{f\left(\varphi(s\,;z)\right)} = -\frac{1}{\frac{\partial\varphi}{\partial x_1}(s\,;z)}\frac{f'(x_1)}{f(x_1)} \tag{4}$$

We are looking for a density function f satisfying equation (4). Set :

$$g(x) = \frac{f'(x)}{f(x)}$$

This is the logarithmic derivative of the function f. From the identity (4), we deduce :

$$g\left(\varphi\left(s\,;z\right)\right)=-\frac{1}{\dfrac{\partial\varphi}{\partial x_1}\left(s\,;z\right)}g(x_1) \tag{5}$$

In the case of the Euclidean norm, we have :

$$\frac{\partial\varphi}{\partial x_1}=-\frac{x_1}{\sqrt{s^2-x_1^2-...-x_{K-1}^2}}$$

and the identity (5) becomes :

$$g\left(\left(s^2-x_1^2-...-x_{K-1}^2\right)^{1/2}\right)=\frac{\sqrt{s^2-x_1^2-...-x_{K-1}^2}}{x_1}g(x_1) \tag{6}$$

The identity (5) allows the construction of the function g : fix any value for x_1, for instance $x_1=1$. Let c be the unknown value of $g(x_1)$. Then, varying all $s,x_2,...,x_{K-1}$, using identity (5) provides a definition of $g(u)$ for any u.

Indeed, fix u and find $s,x_2,...,x_{K-1}$ such that $\varphi(s\,;1,x_2,...,x_{K-1})=u$: then the identity (5) defines $g(u)$.

In the case of the Euclidean norm, one deduces from (6) :

$$g\left(\left(s^2-1\right)^{1/2}\right)=c\sqrt{s^2-1}$$

that is :

$$g\left(u\right)=cu$$

which gives

$$Log\left|f\right|=c\frac{u^2}{2}$$

and :

$$f(u)=\exp\left(c\frac{u^2}{2}\right).$$

From the definition of g, one deduces f, up to a multiplicative constant, which is computed by normalization (the integral of f must be 1). This concludes our construction.

However, there is a compatibility condition ; the identity :

$$g\left(\varphi\left(s\,;z\right)\right)=-\frac{1}{\dfrac{\partial\varphi}{\partial x_1}\left(s\,;z\right)}g(x_1)$$

must give the same value to g when the value of $\varphi(s,z)$ is fixed. This means that the value of the derivative $\frac{\partial\varphi}{\partial x_1}(s\,;z)$, for fixed x_1, must depend only on the value of $\varphi(s\,;z)$. In other words, there must exist a function G such that :

$$\frac{\partial\varphi}{\partial x_1}(s\,;z)=G\left(x_1,\varphi(s\,;z)\right)$$

This is the case for the Euclidean norm, since :

$$\frac{\partial\varphi}{\partial x_1}(s\,;z)=\frac{-x_1}{\varphi(s\,;z)}$$

This concludes the proof of our Theorem.

We note that the condition (*) does not need to be satisfied for any norm on the space R^K. Indeed, a norm on R^K is defined by a symmetric convex body in this space (see for instance B. Beauzamy [5]).

Condition (*) means that the value of the derivative $\frac{\partial\varphi}{\partial x_1}(s\,;z)$, for fixed x_1, depends only on the value of $\varphi(s\,;z)$. Let us see what this means geometrically, in 3 dimensions

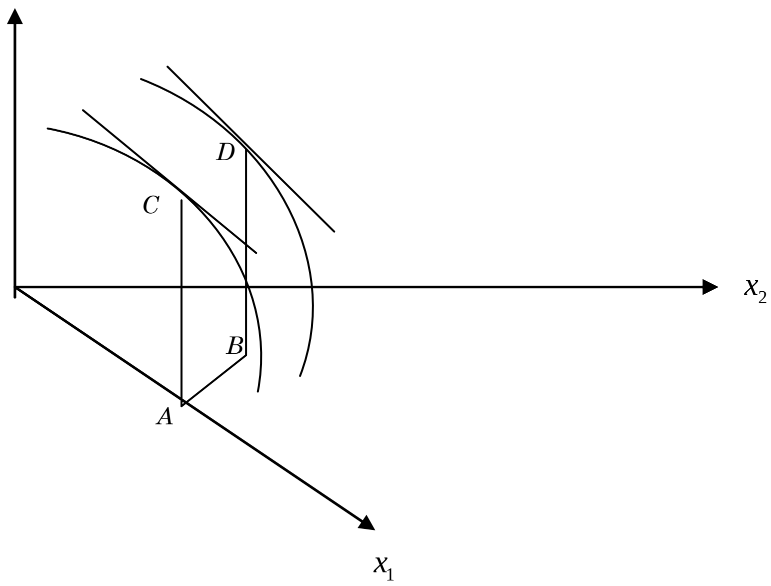


Figure 4 : slopes of the tangent to the unit sphere at two different points

On this picture, the two points A and B have the same value for the first coordinate x_1 and the same value for $\varphi(1;x_1,x_2)$, which is the length $AC=BD$. But the tangent to the unit sphere does not need to be have the same slope in C and in D.

III. Search for dangerous zones

We now apply the previous tools to the search of dangerous zones. We start from the point with highest temperature ; this is the point corresponding to the 175th run, and the observed temperature is 955.76°C. We call H_0 this point.

A. *First step of the iteration*

We fix a threshold, say $T_0 = 900°C$ and we want to find a point where the probability to be above this threshold is highest.

1. Theory

There are 307 measure points where the temperature is below this threshold ; we denote them as C_n, $n = 1,...,NT = 307$ (C stands for "cold").

First, we recall the construction from our first report. For every cold point C_n, we consider the hyperplane E_n equidistant from C_n and H_0, and we consider the half-space F_n determined by this hyperplane, containing H_0 : this is the set of the points in $\mathbb{R}^K$ which are closer to H_0 than to C_n. So we have 307 such half-spaces.

We also consider the half-spaces $x_j \geq 0$, $j = 1,...,K$, and the half-spaces $x_j \leq 1$, $j = 1,...,K$: so we have a total of $H = 391$ half-spaces (since $K = 42$).

The intersection A of all these 391 half-spaces is non-empty, since it contains the point H_0. It is a simplex (intersection of half-spaces), and we want to find the point in it which is farthest from H_0.

The first task is to generate $MT = 5000$ random directions, in $\mathbb{R}^K$, with uniform law. A "random direction" means simply a point on the unit sphere of $\mathbb{R}^K$, equipped with the Euclidean distance. We want a uniform law, because all directions should be considered equally.

Using Theorem 2 above, this is done the following way :

For $mu = 1$ to MT
For $k = 1$ to KT ($KT = K$ is introduced because VBA does not distinguish capitals)
Generate random vectors X_k following a normal law (Gauss with mean 0 and variance 1),
Replace them by

$$\left(\frac{X_1}{\left(X_1^2 + \cdots + X_K^2\right)^{1/2}}, ..., \frac{X_K}{\left(X_1^2 + \cdots + X_K^2\right)^{1/2}} \right)$$

2. VBA code

Here is a VBA code :

```
For mu = 1 To MT 'random sampling
For k = 1 To KT '=42
Randomize
V(k) = Application.WorksheetFunction.NormInv(Rnd(), 0, 1)
Next k

For k = 1 To KT
sum = sum + V(k) ^ 2
Next k

For k = 1 To KT
V(k) = V(k) / Sqr(sum)
Sheets(3).Cells(mu, k + 1) = V(k)
Next k
```

Here is the beginning of this table, which has $MT = 5000$ rows and $K + 1 = 43$ columns :

1	-0,051	-0,086	-0,229	-0,016	-0,091	-0,196	-0,053	0,023	-0,111	-0,114
2	-0,403	-0,053	-0,105	-0,256	0,077	0,176	0,184	0,161	0,244	0,059
3	-0,253	0,014	0,096	0,049	0,329	0,208	-0,045	0,113	-0,087	-0,168
4	-0,324	0,259	0,050	-0,131	-0,104	0,099	-0,137	-0,086	0,048	0,015
5	0,211	-0,008	0,053	0,387	-0,222	0,289	0,018	-0,027	-0,022	-0,038
6	-0,273	-0,044	0,056	-0,042	-0,064	-0,122	0,037	0,013	0,312	0,067
7	0,181	-0,047	0,217	0,008	-0,114	-0,161	0,199	-0,072	-0,168	0,235
8	0,092	-0,068	0,041	0,273	-0,049	0,086	0,043	-0,028	-0,264	-0,116
9	0,164	-0,119	0,220	-0,319	0,004	-0,304	-0,105	-0,173	-0,011	-0,032
10	0,181	0,233	0,116	-0,098	0,019	-0,035	-0,192	0,104	0,127	-0,130
11	-0,090	0,094	0,051	-0,129	-0,110	0,164	0,318	-0,058	0,003	-0,077
12	0,239	-0,005	0,164	-0,013	0,218	-0,090	0,002	-0,088	-0,111	-0,183
13	-0,222	-0,033	0,001	0,026	-0,051	0,218	-0,032	-0,374	0,110	0,066
14	0,268	0,327	-0,163	-0,064	-0,111	0,146	-0,195	-0,010	0,179	-0,119
15	0,108	-0,110	-0,122	0,084	-0,045	0,075	-0,022	0,005	-0,305	0,007

Table 5 : the beginning of the table of random directions

Our random directions will be the vectors $H_0 + V_m$, where V_m is the m-th vector in this list.

We denote by $H(k)$, $k = 1,...,K$ the coordinates of the point H_0 and by $C(n,k)$, $n = 1,...,FT$ $(FT = 307)$, $k = 1,...,K$, the coordinates of the cold points.

Here is the VBA code which lists the cold points ; recall that $NT = 307$ is the number of measurements.

```
For n = 1 To NT
If Sheets(2).Cells(n + 1, 43) < 900 Then 'the value of the measurements
For k = 1 To KT
C(j, k) = Sheets(2).Cells(n + 1, k)
Next k
j = j + 1
End If
Next n
Dim FT As Integer
FT = j - 1
'the number of cold points
```

We now find the middle between H_0 and each cold point C_n ; we call it M_n. Its coordinates are :

$$M(n,\ k)\ =\ \frac{1}{2}\big(H(k)\ +\ C(n,\ k)\big)$$

The vector $\overrightarrow{HC_n}$ from H_0 to C_n has components :

$$HC(n,k) = C(n,k) - H(k)$$

The hyperplane E_n is orthogonal to the vector $\overrightarrow{HC_n}$, so its equation must be of the form :

$$\sum_{k=1}^{K} HC(n,k)x_k = D_n$$

for some number D_n,

and since this hyperplane contains the middle point M_n, we get :

$$\sum_{k=1}^{K} HC(n,k)M(n,k) = D_n$$

So the equation of the hyperplane is :

$$\sum_{k=1}^{K} HC(n,k)x_k = \sum_{k=1}^{K} HC(n,k)M(n,k) \qquad (1)$$

Let $H_0 + \lambda V_m$ be any of the directions defined above, with coordinates $H(k) + \lambda V(m,k)$. Its intersection with the hyperplane E_n is obtained for :

$$\sum_{k=1}^{K} HC(n,k)\big(H(k) + \lambda V(m,k) - M(n,k)\big) = 0$$

that is :

$$\lambda = \frac{\sum_{k=1}^{K} HC(n,k)\left(M(n,k) - H(k)\right)}{\sum_{k=1}^{K} HC(n,k)V(m,k)} \tag{2}$$

and (since V_m is normalized), the distance between H_0 and the intersection with the hyperplane is $|\lambda|$.

The intersection of the line with the hyperplane $x_k = 0$ is :

$$\lambda = -\frac{H(k)}{V(m,k)} \tag{3}$$

And the intersection with the hyperplane $x_k = 1$ is :

$$\lambda = \frac{1 - H(k)}{V(m,k)} \tag{4}$$

Each vector V_m generates one line and two directions, that is $\pm V_m$. For each of these two directions, we consider all the intersections with the hyperplane, and we keep the closest to H_0 :

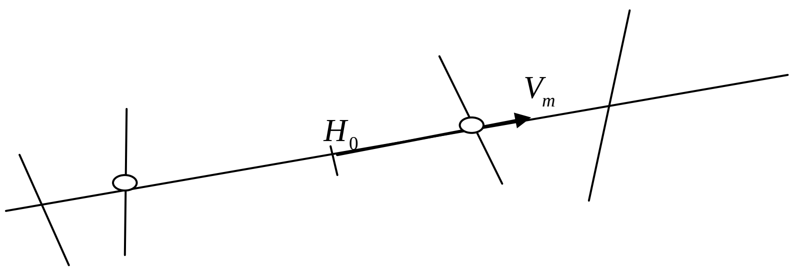


Figure 6 : the line generated by the vector V_m and several intersections with hyperplanes

The procedure goes as follows :

For each n (number of cold points, and of associated hyperplanes), compute the coefficient λ using formula (2). Do the same for the hyperplanes $x_j \geq 0$ and $x_j \leq 1$ (so we have a total of $H = 391$ values for λ). For each of them, check whether λ is positive or negative. Keep :

$\lambda_{\min}^{+} = \min\{\lambda > 0\}$

$\lambda_{\max}^{-} = \max\{\lambda < 0\}$

Here is the associated VBA code :

```
Dim mu As Integer
Dim V(1 To MT, 1 To KT) As Double 'the chosen vector
For mu = 1 To MT
For k = 1 To KT
 V(mu, k) = Sheets(3).Cells(mu, k + 1)
'the place where the random vectors have been stored
Next k
Next mu
```

Note here that one has to generate all random vectors and store them, because later we will have to recall which one was the best. But in fact it would be enough to store just two : the current vector, and the best up to current.

```
Dim a(1 To KT) As Double 'intermediate products


Dim sum1, sum2, sum3, lambda, lambda1,  lambda2, lambda3, lambda1max, lambda2max, As Double
Dim lambda_f As Double 'final
Dim mu1, mu2 As Integer
Dim Mfin() As Double
ReDim Mfin(1 To KT)

For mu = 1 To MT
lambda1 = 100
lambda2 = -100
sum1 = 0
sum2 = 0
For n = 1 To FT
For k = 1 To KT
sum1 = sum1 + HC(n, k) * (M(n, k) - H(k))
sum2 = sum2 + HC(n, k) * V(mu, k)
Next k
lambda = sum1 / sum2
'this gives the intersection point between the line generated by V_m, going through H
'and the hyperplane, middle between H_0 and C_n
'Sheets(5).Cells(n, 1) = lambda
sum1 = 0
sum2 = 0

If lambda > 0 And lambda < lambda1 Then
lambda1 = lambda
'this is the smallest of all lambda >0
End If

If lambda < 0 And lambda > lambda2 Then
lambda2 = lambda
'this is the largest of all lambda <0
End If
Next n

For k = 1 To KT 'taking into account the boundary hyperplanes
If -H(k) / V(mu, k) > 0 And -H(k) / V(mu, k) < lambda1 Then
lambda1 = -H(k) / V(mu, k)
End If

If (1 - H(k)) / V(mu, k) > 0 And (1 - H(k)) / V(mu, k) < lambda1 Then
lambda1 = (1 - H(k)) / V(mu, k)
End If
```

```
If -H(k) / V(mu, k) < 0 And -H(k) / V(mu, k) > lambda2 Then
lambda2 = -H(k) / V(mu, k)
End If

If (1 - H(k)) / V(mu, k) < 0 And (1 - H(k)) / V(mu, k) > lambda2 Then
lambda2 = (1 - H(k)) / V(mu, k)
End If

Next k
'end of boundary hyperplanes

If lambda1max < lambda1 Then
lambda1max = lambda1
mu1 = mu
End If

If Abs(lambda2max) < Abs(lambda2) Then
lambda2max = lambda2
mu2 = mu
End If

'we keep the greatest of lambda1, lambda2, in absolute value

Sheets(5).Cells(mu, 1) = mu
Sheets(5).Cells(mu, 2) = lambda1max
Sheets(5).Cells(mu, 3) = lambda2max
Next mu

Sheets(5).Cells(1, 5) = mu1
Sheets(5).Cells(1, 6) = lambda1max
Sheets(5).Cells(1, 7) = mu2
Sheets(5).Cells(1, 8) = lambda2max

If lambda1max > Abs(lambda2max) Then
lambda_f = lambda1max
For k = 1 To KT
Mfin(k) = H(k) + lambda_f * V(mu1, k)
Next k
Else
lambda_f = lambda2max
For k = 1 To KT
Mfin(k) = H(k) + lambda_f * V(mu2, k)
Next k
End If
For k = 1 To KT
Sheets(5).Cells(k, 10) = Mfin(k)
Next k
'Mfin is the final point we were looking for
Sheets(5).Cells(1, 11) = Abs(lambda_f)
'the distance between H0 and Mfin.
```

Here are the 42 coordinates of the point M_1 (Mfin in the above VBA code) that we obtain.

0,944	0,994	0,516	0,195	0,805	0,448	0,109	0,985	0,561	0,000	0,815	0,514
0,186	0,918	0,426	0,251	0,685	0,532	0,093	0,915	0,347	0,062	0,945	0,231
0,048	0,943	0,443	0,046	0,876	0,364	0,096	0,945	0,241	0,234	0,903	0,464
0,290	0,978	0,224	0,233	0,977	0,224						

Table 7 : the coordinates of the farthest point

The distance to the hot point H_0 is 0.598, and this distance, in our case, is obtained for the 791th vector.

A verification is made, in order to ensure that the point M_1 is on the proper side of all hyperplanes. It must verify :

$$dist(H_0, M_1) \le dist(M_1, C_n) \text{ for all } n$$

The point M_1 must be closer to the hot point than to any of the cold points.

$$0 \le M_1(k) \le 1 \text{ for all } k$$

The point M_1 must be inside the hypercube.

3. Computing the probability to be above the threshold at the point M_1

The point M_1 gets its influence only from the point H_0 ; the observed temperature at H_0 was $\theta = 955.76°C$.

Recall (see our previous report) that the probability discrete distribution at the point M_1 is given by the formula :

$$p_j(M_1) = \frac{c\tau}{\sigma\sqrt{2\pi}} \cdot \exp\left(-\frac{\left(t_j - \theta\right)^2}{2\sigma^2}\right) \qquad (1)$$

where $\theta = 955.76°C$, σ is given by the formula :

$$\sigma = \frac{\tau e^{\lambda d}}{\sqrt{2\pi e}} \qquad (2)$$

where $\tau = 1$ (discretization by 1°C), $\lambda = 1.176$ (see our previous report), and $d = d(H_0, M_1) = 0.598$. We find :

$$\sigma = 0.489$$

And here is the distribution of probability we obtain at the point M_1 :

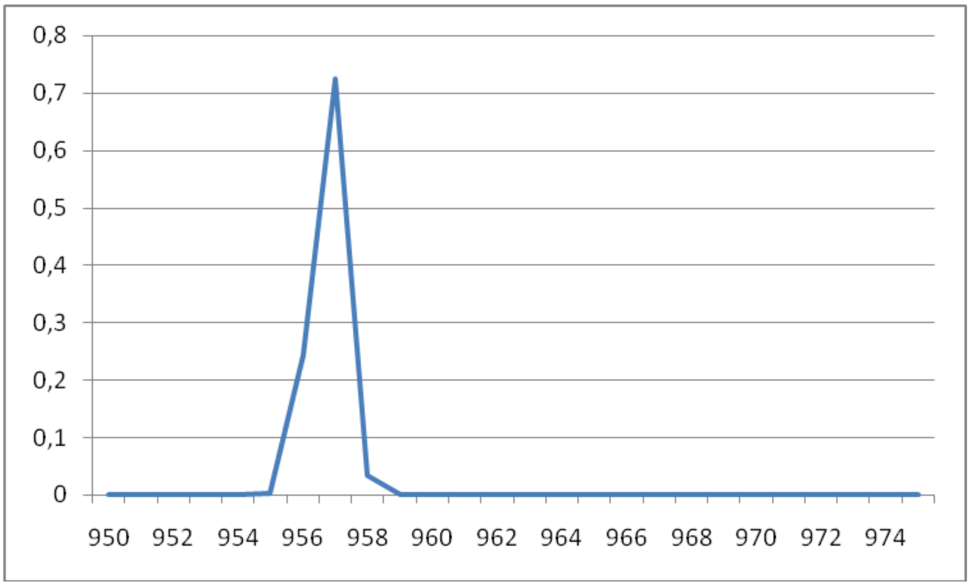


Figure 8 : The distribution of probability at the point M_1

This probability distribution gives very sharp estimates : for instance, the probability to be above 960°C is zero, and the probability to be above 954°C is 1.

B. *Next step in the iteration*

The point M_1 is not necessarily final : it is the farthest point from H_0 among the 5000 directions that we investigated. So a very natural idea is to have a second step : moving away from M_1 and trying to find a point M_2 which is still further from H_0. This will be done, again, using random sampling.

1. Theory

First, we observe that, by construction, the point M_1 is on the edge of our simplex. It must be on one of the limiting hyperplanes. In our case (see table above), we see that its 10-th coordinate is 0. But in other cases, the point might be at equal distance between H_0 and one of the C_n and this property has to be preserved under further moves : we want to stay inside the hyperplane where we are.

2. Random sampling inside a hyperplane

The random sampling we did before was inside the whole space. Let us see what modifications should be made if one wants to remain inside a given hyperplane, or more generally a low dimensional subspace.

The case where we are now is very simple, since one of the coordinates is 0 : all we have to do is to perform a random sampling of the other 41 coordinates, and keep the 10th to be 0.

But for an hyperplane in general position, this simple trick does not work. One possibility is of course to build an orthonormal basis (a new coordinate system), such that

$x_1 = 0$ is the equation of the hyperplane in this basis. Then one samples the other 41 coordinates, and comes back to the original coordinates. This construction may be used for any subspace, of any codimension.

For a general hyperplane, there is, however, a simpler trick : generate (as we did before) a sample in the whole space (42 dimensions) and project the vectors upon the hyperplane : the orthogonal projection of the uniform law in the whole space is a uniform law on the hyperplane.

This method, on the other hand, would be quite unefficient for low dimensional subspaces. Assume for instance that we want to build a uniform law on the unit sphere of a 5-dimensional subspace ; it would require to project from 42 to 41, then from 41 to 40, and so on, until 5. It is much simpler, in this case, to build an orthonormal basis for our 5 dimensional subspace and to complete it as a basis of the whole space.

Let us come back to our specific construction. We generate 5000 vectors V_m in a 41 dimensional space, with uniform law on the unit sphere. Here are, as before, the first 10 coordinates of the first 15 vectors :

-0,334	0,264	-0,041	0,138	-0,086	-0,130	-0,177	-0,060	0,070	0,268
-0,252	-0,208	0,026	-0,264	-0,182	-0,024	-0,059	-0,325	-0,099	0,175
-0,017	0,174	0,409	0,143	-0,049	-0,094	-0,088	-0,105	-0,057	-0,245
-0,025	0,078	-0,024	-0,046	-0,102	0,058	0,032	-0,393	0,090	0,227
0,054	-0,212	0,072	-0,112	0,280	0,189	0,064	-0,294	0,181	0,079
0,078	0,038	-0,138	-0,010	0,140	-0,112	0,033	-0,007	-0,086	0,175
0,046	-0,171	0,135	-0,186	0,176	0,251	-0,027	0,256	-0,166	-0,290
-0,188	-0,200	-0,032	-0,003	0,019	-0,048	0,182	-0,164	-0,353	0,091
-0,057	0,066	0,142	0,159	-0,397	-0,015	-0,058	0,249	0,309	0,040
0,073	-0,026	-0,024	0,273	-0,179	-0,053	0,206	-0,088	0,187	-0,060
-0,192	0,101	0,049	0,103	-0,044	0,156	0,034	0,177	0,280	-0,231
-0,260	0,015	0,065	0,032	0,142	0,184	-0,180	0,024	0,037	0,124
-0,128	-0,083	0,032	0,127	-0,154	-0,026	-0,146	0,246	-0,020	0,199
0,201	0,006	0,075	0,080	0,066	0,109	-0,122	-0,062	-0,007	-0,230
0,118	0,065	-0,309	0,077	-0,166	0,138	-0,287	-0,161	-0,225	-0,083

Table 9 : The first 10 coordinates of the first 15 vectors, dimension 41

Our random vectors V_m are obtained by inserting a 0 at the 10^{th} place (so now each vector has 42 components). This is what we get for the first 11 coordinates of the first 15 vectors :

-0,334	0,264	-0,041	0,138	-0,086	-0,130	-0,177	-0,060	0,070	0,000	0,268
-0,252	-0,208	0,026	-0,264	-0,182	-0,024	-0,059	-0,325	-0,099	0,000	0,175
-0,017	0,174	0,409	0,143	-0,049	-0,094	-0,088	-0,105	-0,057	0,000	-0,245
-0,025	0,078	-0,024	-0,046	-0,102	0,058	0,032	-0,393	0,090	0,000	0,227
0,054	-0,212	0,072	-0,112	0,280	0,189	0,064	-0,294	0,181	0,000	0,079
0,078	0,038	-0,138	-0,010	0,140	-0,112	0,033	-0,007	-0,086	0,000	0,175
0,046	-0,171	0,135	-0,186	0,176	0,251	-0,027	0,256	-0,166	0,000	-0,290
-0,188	-0,200	-0,032	-0,003	0,019	-0,048	0,182	-0,164	-0,353	0,000	0,091
-0,057	0,066	0,142	0,159	-0,397	-0,015	-0,058	0,249	0,309	0,000	0,040
0,073	-0,026	-0,024	0,273	-0,179	-0,053	0,206	-0,088	0,187	0,000	-0,060
-0,192	0,101	0,049	0,103	-0,044	0,156	0,034	0,177	0,280	0,000	-0,231
-0,260	0,015	0,065	0,032	0,142	0,184	-0,180	0,024	0,037	0,000	0,124
-0,128	-0,083	0,032	0,127	-0,154	-0,026	-0,146	0,246	-0,020	0,000	0,199
0,201	0,006	0,075	0,080	0,066	0,109	-0,122	-0,062	-0,007	0,000	-0,230
0,118	0,065	-0,309	0,077	-0,166	0,138	-0,287	-0,161	-0,225	0,000	-0,083

Table 10 : The first 11 coordinates of the first 15 vectors, dimension 42

The midpoints between H_0 and each cold point C_n are the same points M_n as previously, and the vector $\overrightarrow{HC_n}$ from H_0 to C_n is also the same.

The hyperplane E_n, orthogonal to the vector $\overrightarrow{HC_n}$, and passing through M_n has an equation :

$$\sum_{k=1}^{K} HC(n,k)x_k = \sum_{k=1}^{K} HC(n,k)M(n,k) \tag{1}$$

But now, the directions defined are of the type $M_1 + \lambda V_m$, with coordinates $M_1(k) + \lambda V(m,k)$. Their intersection with the hyperplane E_n is obtained for :

$$\sum_{k=1}^{K} HC(n,k)\left(M_1(k) + \lambda V(m,k) - M(n,k)\right) = 0$$

that is :

$$\lambda = \frac{\sum_{k=1}^{K} HC(n,k)\left(M(n,k) - M_1(k)\right)}{\sum_{k=1}^{K} HC(n,k)V(m,k)} \tag{2}$$

and (since V_m is normalized), the distance between M_1 and the intersection with the hyperplane is $|\lambda|$.

The intersection of the line with the hyperplane $x_k = 0$ is :

$$\lambda = -\frac{M_1(k)}{V(m,k)} \tag{3}$$

And the intersection with the hyperplane $x_k = 1$ is :

$$\lambda = \frac{1 - M_1(k)}{V(m,k)} \tag{4}$$

3. VBA Code

Here is the VBA code :

```
Dim H(1 To KT) As Double 'coordinates of the hottest point
Dim k As Integer

For k = 1 To KT
H(k) = Sheets(1).Cells(176, k)
Next k

Dim M1(1 To KT) As Double 'coordinates of the point M1
For k = 1 To KT
M1(k) = Sheets(3).Cells(k + 1, 1)
Next k

Dim C(1 To NT, 1 To KT) As Double 'the coordinates of the cold points
Dim n, j As Integer
j = 1

For n = 1 To NT
If Sheets(2).Cells(n + 1, 43) < 900 Then
For k = 1 To KT
C(j, k) = Sheets(2).Cells(n + 1, k)
Next k
j = j + 1
End If
Next n
Dim FT As Integer
FT = j - 1 'the number of cold points

Dim M() As Double 'middle between H and C
ReDim M(1 To FT, 1 To KT) As Double

For n = 1 To FT
For k = 1 To KT
M(n, k) = (H(k) + C(n, k)) / 2
Next k
Next n

Dim HC() As Double 'the vectors H_0 C_n
ReDim HC(1 To FT, 1 To KT) As Double

For n = 1 To FT
For k = 1 To KT
HC(n, k) = C(n, k) - H(k)
Next k
Next n
```

```
Dim mu, mu1 As Integer
Dim V(1 To MT, 1 To KT) As Double 'the chosen vector
For mu = 1 To MT
For k = 1 To KT
V(mu, k) = Sheets(2).Cells(mu, k)
Next k
Next mu

Dim sum1, sum2, alpha, beta, gamma, lambda, As Double

Dim M2() As Double
ReDim M2(1 To KT)

Dim d2, dmax2 As Double

For mu = 1 To MT
lambda = 1

sum1 = 0
sum2 = 0
For n = 1 To FT
For k = 1 To KT
sum1 = sum1 + HC(n, k) * (M(n, k) - M1(k))
sum2 = sum2 + HC(n, k) * V(mu, k)
Next k
alpha = sum1 / sum2
'this gives the intersection point between the line generated by V_m, going through M1
'and the hyperplane, middle between H_0 and C_n

sum1 = 0
sum2 = 0

If alpha > 0 And alpha < lambda Then
lambda = alpha
'this is the smallest of all alpha >0
End If

Next n

For k = 1 To 9  'taking into account the boundary hyperplanes
beta = -M1(k) / V(mu, k)
gamma = (1 - M1(k)) / V(mu, k)
If beta > 0 And beta < lambda Then
lambda = beta
End If

If gamma > 0 And gamma < lambda Then
lambda = gamma
End If
Next k

For k = 11 To KT  'taking into account the boundary hyperplanes, 2
beta = -M1(k) / V(mu, k)
gamma = (1 - M1(k)) / V(mu, k)
If beta > 0 And beta < lambda Then
lambda = beta
End If

If gamma > 0 And gamma < lambda Then
lambda = gamma
End If
Next k
```

```
'end of boundary hyperplanes

For k = 1 To KT
d2 = d2 + (H(k) - M1(k) - lambda * V(mu, k)) ^ 2
Next k

If dmax2 < d2 Then
dmax2 = d2
mu1 = mu
lambda1 = lambda
End If
'we keep the greatest distance

d2 = 0
Next mu

For k = 1 To KT
M2(k) = M1(k) + lambda1 * V(mu1, k)
Sheets(3).Cells(k + 1, 3) = M2(k)
Next k
Sheets(3).Cells(1, 3) = "M2"
Sheets(3).Cells(1, 5) = mu1
Sheets(3).Cells(1, 6) = lambda1
Sheets(3).Cells(1, 7) = Sqr(dmax2)
End Sub
```

Here are the coordinates of the point M_2 :

0,927	0,938	0,486	0,077	0,945	0,472	0,084	0,951	0,457	0,093
0,957	0,443	0,101	0,962	0,429	0,110	0,968	0,415	0,119	0,972
0,401	0,128	0,977	0,387	0,138	0,981	0,373	0,148	0,985	0,360
0,158	0,988	0,346	0,168	0,991	0,333	0,179	0,993	0,319	0,190
0,995	0,306								

Table 11 : The coordinates of the new point

The distance $d(H_0, M_2) = 0.986$, which shows a considerable improvement compared with step 1. This value is obtained for the 3961st vector.

As above, a verification is made, in order to ensure that the point M_2 is on the proper side of all hyperplanes. It must verify :

$$dist(H_0, M_2) \le dist(M_2, C_n) \text{ for all } n,$$

which means that the point M_2 must be closer to the hot point than to any of the cold points, and finally :

$$0 \le M_1(k) \le 1 \text{ for all } k$$

which means that the point M_2 must be inside the hypercube.

4. Computing the probability to be above the threshold at the point M_2

The point M_2 gets its influence only from the point H_0 ; the observed temperature at H_0 was $\theta = 955.76°C$.

As before, the probability discrete distribution at the point M_1 is given by the formula :

$$p_j(M_1) = \frac{c\tau}{\sigma\sqrt{2\pi}} \cdot \exp\left(-\frac{\left(t_j - \theta\right)^2}{2\sigma^2}\right) \quad (1)$$

where $\theta = 955.76°C$, σ is given by the formula :

$$\sigma = \frac{\tau e^{\lambda d}}{\sqrt{2\pi e}} \quad (2)$$

where $\tau = 1$ (discretization by 1°C), $\lambda = 1.176$, and $d = d(H_0, M_2) = 0.986$. We find :

$$\sigma = 0.77$$

And here is the distribution of probability we obtain at the point M_2:

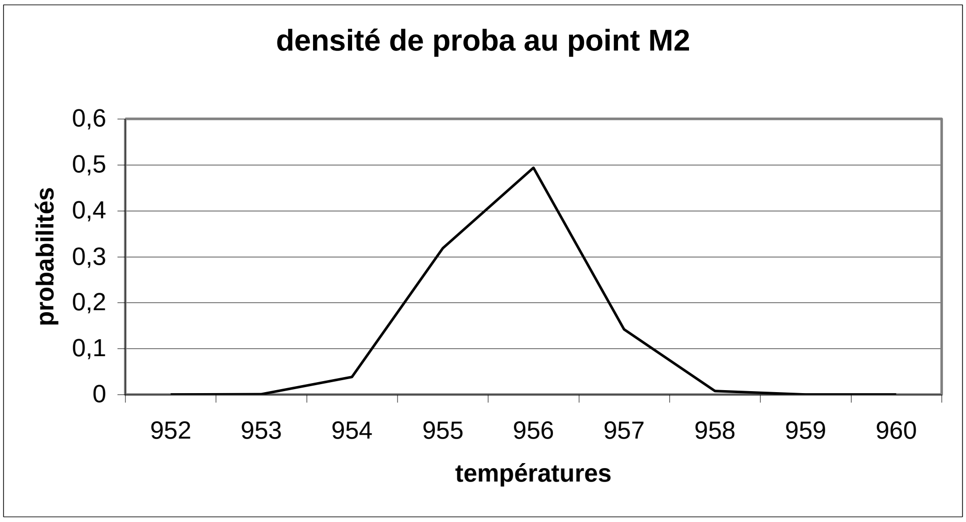


Figure 12 : Graph of the probability density above the point M_2

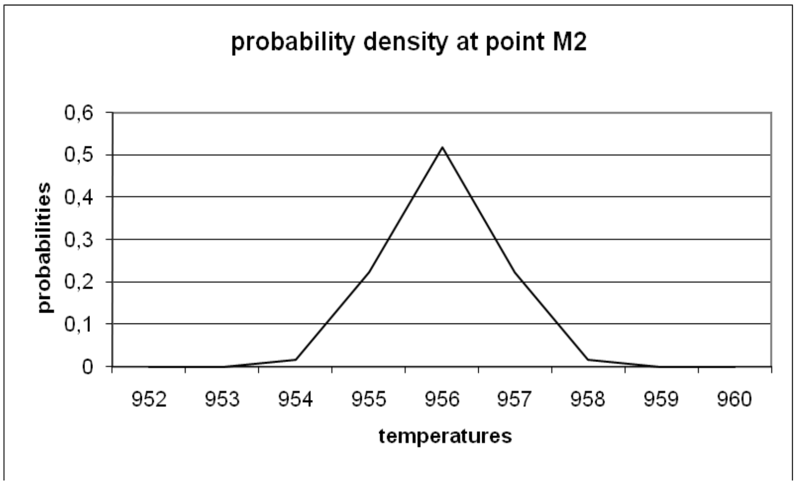
As the theory predicts, this density is less concentrated than the one we got previously ; however, it is still quite sharp, and we can conclude that the probability to be above 960°C is still 0.

The above graph is not gaussian ; in fact it is "discrete gaussian" ; here are the data :

temp	proba
952	3,4402E-06
953	0,00084043
954	0,03801328
955	0,31832827
956	0,49354087
957	0,14167038
958	0,0075291
959	7,4082E-05
960	1,3496E-07
961	4,5518E-11
962	2,8424E-15
963	3,2861E-20

Table 13 : The probability data above the final point

The graph is not symmetric, with respect to the peak. It would be, if the peak was at some integer value, but the peak is at 955.76, and this explains the lack of symmetry. Here is the same graph, taking the peak at 956°C :



Graph 14 : probability density above the final point, approximate value of the peak

Acknowledgements

We thank Paul Deheuvels for indicating the book by Luc Devroye, Luc Devroye for his comments about the proofs, and Roger Stafford for his comments about the Matlab program.

References

[1] Rapport SCM IRSN : applications de l'EPH, juin 2008.

[2] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf : *Computational Geometry*, 2nd revised edition, Springer-Verlag, 2000.

[3] Luc Devroye : Non Uniform Random Variate Generation, Springer-Verlag, New York, 1986.

[4] R. J. Muirhead : Aspects of Multivariate Statistical Theory, Wiley, New York, 1982.

[5] B. Beauzamy : Introduction to Banach Spaces and their geometry, North Holland. Collection "Notas de Matematica", vol. 68. Second edition, 1985.