# REGRESSION et EQUATION INTEGRALE.

# APPLICATION A LA DISTRIBUTION GAUSSIENNE

**Jean Jacquelin**

## 1. Introduction

La présente étude se situe dans le cadre général des problèmes de régression. Par exemple, on connaît les coordonnées d'une série de $n$ points : $(x_1 , y_1)$, $(x_2 , y_2)$, …, $(x_k , y_k)$, …, $(x_n , y_n)$ et l'on cherche à ce que la courbe représentative d'une fonction $y = F(a, b, c,…; x)$ passe au plus près de ces points, ceci en optimisant les valeurs des paramètres $a$, $b$, $c$, …

Le cas bien connu de la régression linéaire ne mérite qu'un bref rappel, ce qui sera fait dans le paragraphe suivant. Pour certains cas apparemment non linéaires et bien que cela puisse échapper à première vue, il est possible de revenir à une régression linéaire. Le cas de la fonction de répartition gaussienne en est un exemple : il sera traité au §.3.

Hors les cas simples précédents, on est confronté à un véritable problème de régression non linéaire. La littérature sur le sujet est très étendue. Une revue, même sommaire, nous éloignerait de l'objectif du présent papier. Nous n'en aurons pas besoin ici car notre parti est de ramener certains problèmes non linéaires à une régression linéaire sans processus itératif ou récursif (si non, où serait l'originalité par rapport à des méthodes couramment utilisées ? ).

A partir du §.4, on entre dans le vif du sujet : c'est-à-dire les possibilités de ramener un problème non linéaire à une forme linéaire grâce à une équation différentielle et/ou intégrale convenable. La discussion préliminaire montre que, sauf cas particuliers, une équation intégrale est mieux adaptée à la résolution par calcul numérique qu'une équation différentielle, dans le contexte de ce genre de problèmes. Le principe de l'utilisation d'une équation intégrale sera exposé et mis en pratique en prenant pour exemple la fonction de distribution gaussienne.

## 2. Régression linéaire

Lorsque la fonction $y = F(a, b, c,…; x)$ que l'on cherche à optimiser peut se mettre sous la forme : $y = a\ f(x) + b\ g(x) + c\ h(x) + …$ , selon le nombre de paramètres a, b, c, … et avec des fonctions $f(x)$, $g(x)$, $h(x)$, … connues, le processus est linéaire relativement aux paramètres à optimiser.

Encore plus généralement, si la fonction $y = F(a, b, c,…; x)$ peut être transformée et mise sous la forme : $F(x,y) = A\ f(x,y) + B\ g(x,y) + C\ h(x,y) + …$ avec des fonctions connues : $F(x,y)$, $f(x,y)$, $g(x,y)$, $h(x,y)$,… , $A(a,b,c,…)$, $B(a,b,c,…)$, $C(a,b,c,…)$, … le processus est encore linéaire relativement aux coefficients A, B et C, bien qu'il ne le soit plus relativement à $a$, $b$, $c$, … Mais il relève toujours d'une régression linéaire. En effet, la méthode "des moindres carrés" consiste à chercher le minimum de :

$$\begin{cases} \varepsilon^2{}_{(A,B,C,...)} = \sum_{k=1}^{n} \left( F_k - \left( A\ f_k + B\ g_k + C\ h_k + ... \right) \right)^2 \\ F_k \equiv F(x_k, y_k)\ ;\ f_k \equiv f(x_k, y_k)\ ;\ g_k \equiv g(x_k, y_k)\ ;\ h_k \equiv h(x_k, y_k) \end{cases}$$

Les dérivées partielles relatives à $A$, $B$, $C$, … conduisent au système d'équations dont les solutions $A_0$, $B_0$, $C_0$, … sont optimum :

$$\begin{cases} \left( \frac{\partial(\varepsilon^2)}{\partial A} \right)_{A_0, B_0, C_0, ...} = -\sum_{k=1}^{n} \left( F_k - \left( A_0\ f_k + B_0\ g_k + C_0\ h_k + ... \right) \right) f_k = 0 \\ \left( \frac{\partial(\varepsilon^2)}{\partial B} \right)_{A_0, B_0, C_0, ...} = -\sum_{k=1}^{n} \left( F_k - \left( A_0\ f_k + B_0\ g_k + C_0\ h_k + ... \right) \right) g_k = 0 \\ \left( \frac{\partial(\varepsilon^2)}{\partial C} \right)_{A_0, B_0, C_0, ...} = -\sum_{k=1}^{n} \left( F_k - \left( A_0\ f_k + B_0\ g_k + C_0\ h_k + ... \right) \right) h_k = 0 \\ ... \end{cases}$$

La résolution de ce système linéaire, écrit conventionnellement avec $\sum \equiv \sum_{k=1}^{n}$ conduit à :

$$\begin{pmatrix} A_0 \\ B_0 \\ C_0 \\ ... \end{pmatrix} = \begin{pmatrix} \sum f_k{}^2 & \sum f_k g_k & \sum f_k h_k & ... \\ \sum f_k g_k & \sum g_k{}^2 & \sum g_k h_k & ... \\ \sum f_k h_k & \sum g_k h_k & \sum h_k{}^2 & ... \\ ... & ... & ... & ... \end{pmatrix}^{-1} \begin{pmatrix} \sum F_k f_k \\ \sum F_k g_k \\ \sum F_k h_k \\ ... \end{pmatrix}$$

Ensuite, on obtient les valeurs optimum de $a$, $b$, $c$, … correspondantes par résolution du système suivant, dont les inconnues sont $a_0$, $b_0$, $c_0$, … :

$$\begin{cases} A(a_0, b_0, c_0, ...) = A_0 \\ B(a_0, b_0, c_0, ...) = B_0 \\ C(a_0, b_0, c_0, ...) = C_0 \\ ... \end{cases}$$

qui est un système d'équations non linéaires dans la mesure où les fonctions A($a$,$b$,$c$,…), B($a$,$b$,$c$,…), C($a$,$b$,$c$,…), … ne sont pas linéaires. Mais cela n'empêche pas que la régression qui a été faite est linéaire, donc que ce cas a bien sa place dans le présent paragraphe.

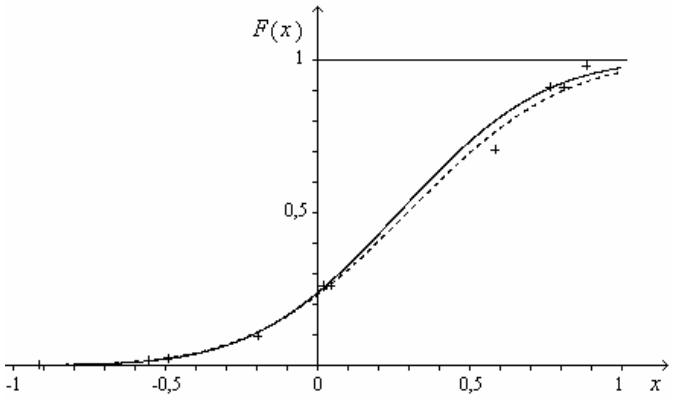
Bien sûr, ceci peut être encore étendu en considérant plus de variables, par exemple $x$, $y$, $z$, $t$, …, au lieu de seulement $x$, $y$ et donc de travailler en 3D., ou 4D., … au lieu de 2D.. Tout ce qui précède figure dans la littérature de façon plus détaillée et surtout mieux structurée, avec des présentations adaptées à une théorie générale. Ici, le propos était seulement un bref rappel, avec les notations spécifiques cohérentes avec celles utilisées par la suite.

**3. Application à la fonction de répartition gaussienne :**

Nous considérons la fonction de répartition gaussienne non centrée, à deux paramètres $\sigma$ et $\mu$, définie par :

$$F(x) = \frac{1}{\sqrt{2\pi}\,\sigma} \int_{-\infty}^{x} \exp\left( -\frac{1}{2}\left( \frac{t-\mu}{\sigma} \right)^2 \right) dt \qquad [1]$$

Un exemple est représenté sur la figure 1 (courbe en pointillés).




| $k$ | $x_k$ | $F_k$ | argErf(2 $F_k$ -1) |
|---|---|---|---|
| 1 | -0,914 | 0,001 | -2,18512 |
| 2 | -0,556 | 0,017 | -1,49912 |
| 3 | -0,49 | 0,021 | -1,43792 |
| 4 | -0,195 | 0,097 | -0,918416 |
| 5 | 0,019 | 0,258 | -0,459283 |
| 6 | 0,045 | 0,258 | -0,459283 |
| 7 | 0,587 | 0,704 | 0,378967 |
| 8 | 0,764 | 0,911 | 0,952429 |
| 9 | 0,81 | 0,911 | 0,952429 |
| 10 | 0,884 | 0,979 | 1,43792 |

$\sigma = 0,4$ $\quad \sigma_1 = 0,374462$

$\mu = 0,3$ $\quad \mu_1 = 0,266843$

Figure 1 : Exemple de régression, cas d'une fonction de répartition gaussienne.

Les données sont les points dit "expérimentaux" :

$(x_1 , F_1), (x_2 , F_2), \ldots, (x_k , F_k), \ldots, (x_n , F_n)$

qui, sur l'exemple de la figure 1, présentent une certaine dispersion par rapport à leur positions théoriques respectives ( $x_k$, $F(x_k)$ ) sur la courbe en pointillés représentative de $F(x)$.

Une forme équivalente d'écriture de $F(x)$ se réfère à la fonction Erf , dite "fonction d'erreur" et définie par :

$$\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z \exp\left(-\tau^2\right) d\tau \qquad [2]$$

Le changement de variable $t = \mu + \sqrt{2}\,\sigma\,\tau$ dans [1] donne la relation :

$$F(x) = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\frac{x-\mu}{\sqrt{2}\,\sigma}} \exp\left(-\tau^2\right) d\tau = \frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\left( \frac{x-\mu}{\sqrt{2}\,\sigma} \right) \qquad [3]$$

La fonction réciproque ou "inverse" de Erf est désignée par $\mathrm{Erf}^{(-1)}$ , ou Erfinv, ou argErf. Nous utiliserons cette dernière notation.

Ainsi, la relation réciproque de [3] s'écrit :

$$\frac{x-\mu}{\sqrt{2}\,\sigma} = \mathrm{argErf}\left(2F(x) - 1\right) \qquad [4]$$

Ce qui conduit à la relation linéaire relativement à $A$ et $B$ définis par :

$$y(x) = \text{argErf}\left(2F(x) - 1\right) = A\,x + B \qquad \begin{cases} A = \dfrac{1}{\sqrt{2}\,\sigma} \\ B = -\dfrac{\mu}{\sqrt{2}\,\sigma} \end{cases} \qquad [5]$$

Il s'agit donc d'une régression linéaire sous sa forme la plus élémentaire, relativement aux points ($x_k$ , $y_k$) avec les $y_k$ calculés préalablement par :

$$y_k = \text{argErf}\left(2F_k - 1\right) \qquad [6]$$

Les valeurs optimum $A_1$, $B_1$ sont les solutions du système suivant :

$$\begin{pmatrix} A_1 \\ B_1 \end{pmatrix} = \begin{pmatrix} \sum (x_k)^2 & \sum x_k \\ \sum x_k & n \end{pmatrix}^{-1} \begin{pmatrix} \sum y_k x_k \\ \sum y_k \end{pmatrix} \qquad [7]$$

Avec conventionnellement : $\sum \equiv \sum_{k=1}^{n}$ . On déduit ensuite $\sigma_1$ et $\mu_1$ d'après [5] :

$$\sigma_1 = \frac{1}{\sqrt{2}\,A} \quad ; \quad \mu_1 = -\frac{B}{A} \qquad [8]$$

Pour l'exemple traité, les valeurs numériques $\sigma_1$ et $\mu_1$ obtenues sont indiquées sur la figure 1, où la courbe représentative de la fonction correspondante est tracée en trait plein. Elle est voisine de la courbe "théorique" en pointillé.

En fait, l'exemple a été choisi intentionnellement avec un très faible nombre de points et une forte dispersion pour que les deux courbes soient bien distinctes l'une de l'autre, ce qui est plutôt dépréciatif et peu représentatif de la qualité de ce qui est obtenu le plus souvent.

En résumé, le processus de calcul numérique, très simple, est le suivant :

<u>Données</u> : ($x_1$ , $F_1$), ($x_2$ , $F_2$), …, ($x_k$ , $F_k$), …, ($x_n$ , $F_n$)

- Calcul des $y_k$, équation [6]
- Calcul de $\sum x_k$ , $\sum (x_k)^2$ , $\sum y_k$ , $\sum y_k x_k$
- Calcul de $A_1$ et $B_1$, système [7]
- Calcul de $\sigma_1$ et $\mu_1$ , relations [8]

<u>Résultat</u> : $\sigma_1$ et $\mu_1$ sont les valeurs approchées de $\sigma$ et $\mu$

Si l'on ne dispose pas de la fonction argErf implémentée dans le logiciel utilisé, un exemple de listing pour les fonctions Erf et argErf est donné en annexe.

**4. Principe de linéarisation par équation différentielle et/ou intégrale :**

Commençons par un sommaire concernant les approximations des dérivées et/ou des primitives par calcul numérique. Etant donnés $n$ points $(x_k, y_k)$ situés à proximité de la courbe représentative d'une fonction $y(x)$ et étant donnée une autre fonction $g(x)$, on peut calculer les approximations pour les dérivées et/ou intégrales suivantes, avec $g_k = g(x_k)$ :

$$D_k = \frac{g_{k+1}y_{k+1} - g_{k-1}y_{k-1}}{x_{k+1} - x_{k-1}} \simeq \left( \frac{d}{dx} g(x)y(x) \right)_{(x=x_k)}$$

$$DD_k = \frac{D_{k+1} - D_{k-1}}{x_{k+1} - x_{k-1}} \simeq \left( \frac{d^2}{dx^2} g(x)y(x) \right)_{(x=x_k)}$$

etc.

$$S_k \simeq \int_{x_1}^{x} g(u)y(u)du \quad \begin{cases} S_1 = 0 \quad \text{et pour } k = 2 \to n \text{ :} \\ S_k = S_{k-1} + \frac{1}{2}(g_k y_k + g_{k-1}y_{k-1})(x_k - x_{k-1}) \end{cases}$$

$$SS_k \simeq \int_{x_1}^{x} \left( \int_{x_1}^{v} g(u)y(u)du \right) dv \quad \begin{cases} SS_1 = 0 \quad \text{et pour } k = 2 \to n : \\ SS_k = SS_{k-1} + \frac{1}{2}(S_k + S_{k-1})(x_k - x_{k-1}) \end{cases}$$

etc.

Il va sans dire que les points doivent être préalablement ordonnés selon les $x_k$ croissants.

Il serait possible d'utiliser des méthodes de dérivation et/ou d'intégration numérique plus sophistiquées. Rien n'empêche non plus de prendre la (ou les) borne(s) inférieure(s) d'intégration autres que $x_1$ et même différentes entre elles pour les intégrations successives. Mais cela compliquerait les formules et alourdirait les explications. Pour faire simple, restons en aux formules les plus élémentaires possibles, du moins à ce stade de l'exposé.

Revenons maintenant à la formulation initiale du problème: optimiser les paramètres $a$, $b$, $c$, … d'une fonction $y(a, b, c, \ldots ; x)$ de telle sorte que sa courbe représentative passe au plus près de $n$ points donnés $(x_k, y_k)$. Bien évidemment, les expressions littérales des dérivées et des primitives de cette fonction dépendent de $a$, $b$, $c$, … Mais, leurs valeurs approchées calculées selon les formules précédentes, c'est-à-dire les valeurs numériques $D_k$ , $DD_k$ , …, $S_k$ , $SS_k$ , … sont obtenues uniquement à partir des données $(x_k, y_k)$ et **sans avoir besoin de connaître** $a$, $b$, $c$, … : cette observation est fondamentale dans la compréhension de la méthode qui va être exposée.

Supposons que la fonction $y(a, b, c, \ldots ; x)$ soit solution d'une équation différentielle et/ou intégrale linéaire telle que :

$$y(x) = A\,\Phi(x) + B\int G(x)y(x)dx + C\int\int H(x)y(x)dx^2 + \ldots + \alpha\frac{d}{dx}g(x)y(x) + \beta\frac{d^2}{dx^2}h(x)y(x) + \ldots$$

avec $\Phi(x)$, $G(x)$, $H(x)$, …, $g(x)$, $h(x)$, … des fonctions données ne dépendant pas de $a$, $b$, $c$, … et les coefficients $A$, $B$, C, …, $\alpha$, $\beta$, … dépendant de $a$, $b$, $c$, …

Les valeurs approximatives sont donc respectivement :

$$\Phi_k = \Phi(x_k)\,;\, G_k = G(x_k)\,;\, H_k = H(x_k)\,;\,...\,;\, \alpha_k = \alpha(x_k)\,;\, \beta_k = \beta(x_k)\,;...$$

$$D_k = \frac{g_{k+1}y_{k+1} - g_{k-1}y_{k-1}}{x_{k+1} - x_{k-1}}$$

$$DD_k = \frac{\Delta_{k+1} - \Delta_{k-1}}{x_{k+1} - x_{k-1}} \quad \text{avec} \quad \Delta_k = \frac{h_{k+1}y_{k+1} - h_{k-1}y_{k-1}}{x_{k+1} - x_{k-1}}$$

$$S_1 = 0 \quad ; \quad S_k = S_{k-1} + \frac{1}{2}(G_k y_k + G_{k-1}y_{k-1})(x_k - x_{k-1})$$

$$\begin{cases} SS_1 = 0 \quad ; \quad SS_k = SS_{k-1} + \frac{1}{2}(\Xi_k + \Xi_{k-1})(x_k - x_{k-1}) \\ \text{avec :}\ \Xi_k = 0 \quad ; \quad \Xi_k = \Xi_{k-1} + \frac{1}{2}(H_k y_k + H_{k-1}y_{k-1})(x_k - x_{k-1}) \end{cases}$$

Si l'on remplace les dérivées et/ou primitives littérales par leurs approximations, l'équation cesse d'être exactement vérifiée. On considère alors la somme des écarts quadratiques :

$$\sum_{k=1}^{n} \varepsilon_k^2 = \sum_{k=1}^{n} \left(-y_k + A\,\Phi_k + B\,S_k + C\,SS_k + ... + \alpha\,D_k + \beta\,DD_k + ...\right)^2$$

La relation est linéaire relativement à $A$, $B$, $C$, …, $\alpha$, $\beta$, … Ainsi, on se trouve ramené à une régression linéaire classique qui permet de calculer les valeurs optimums de $A_o$, $B_o$, $C_o$, …, $\alpha_o$, $\beta_o$, … Finalement, puisque $A$, $B$, $C$, …, $\alpha$, $\beta$, … sont des fonctions connues de $a$, $b$, $c$, … on aura à résoudre le système d'équations : $A(a,b,c,\ldots)=A_o$ ; $B(a,b,c,\ldots)=B_o$ ; … ; $\alpha(a,b,c,\ldots)=\alpha_o$ ; $\beta(a,b,c,\ldots)=\beta_o$ ; … pour obtenir les valeurs optimum des paramètres $a$, $b$, $c$, …

Des conditions complémentaires sont à prendre en considération, concernant le choix de l'équation différentielle et/ou intégrale. Outre qu'elle doit être linéaire relativement aux coefficients (mais non au sens des fonctions elles-mêmes, puisqu'on dispose du choix des $G(x)$, $H(x)$, …, $g(x)$, $h(x)$, …), l'équation doit, de préférence comporter autant de coefficients $A_o$, $B_o$, …, $\alpha_o$, $\beta_o$, … qu'il y a de paramètres initiaux $a$, $b$, $c$, … à optimiser. S'il y en a moins, une (ou des) régression supplémentaire serra nécessaire pour calculer les coefficients ne figurant pas explicitement dans l'équation.

De plus, pour ne pas surcharger l'exposé, on a considéré une forme réduite d'équation différentielle et/ou intégrale. En fait, elle pouvait aussi comporter plusieurs fonctions $\Phi(x)$ différentes, plusieurs dérivées différentes (correspondant à des $g(x)$ différents), plusieurs intégrales différentes (correspondant à des $G(x)$ différents) et ainsi de suite pour les dérivées multiples et intégrales multiples.

On voit donc que l'on dispose de possibilités très nombreuses pour adapter une équation différentielle et/ou intégrale au problème à traiter. Toutefois, des contingences pratiques limitent ce choix. L'une des principales pierres d'achoppement résulte des difficultés inhérentes aux dérivations numériques. En effet, dans les cas où les points donnés ne sont pas régulièrement répartis, s'ils sont peu nombreux et insuffisamment proches les uns des autres et si, pour aggraver encore la situation, les valeurs des $y_k$ ne sont pas assez précises, les dérivées calculées deviennent très fluctuantes, très dispersées, rendant inefficiente la régression linéaire qui s'en suit. Au contraire, même dans ces cas difficiles, les intégrations numériques conservent une bonne stabilité (ce qui ne veut pas dire que les inévitables déviations sont faibles, mais au moins elles restent amorties, ce

qui est essentiel pour la robustesse du procédé). Sauf cas particulier, il est donc largement préférable de s'orienter vers une équation intégrale plutôt qu'une équation comportant une fonction dérivée.

La généralité de la présentation qui vient d'être faite peut donner l'impression que la méthode est ardue et difficile à mettre en œuvre. Hors c'est tout le contraire lorsque l'on cesse de parler d'une façon abstraite, couvrant trop de cas différents et lorsque l'on s'applique à résoudre un cas concret.

L'un des exemples les plus spectaculaires est celui de la régression sinusoïdale (que nous évoquons seulement, sans approfondir ici ). Il s'agit d'optimiser les paramètres *a*, *b*, *c* et $\omega$ de l'équation :

$$y(x) = a + b\sin(\omega x) + c\cos(\omega x)$$

Cette fonction est solution de l'équation différentielle :

$$y(x) = A + B\frac{d^2y}{dx^2} \quad \text{avec : } A = a \text{ et } B = -\frac{1}{\omega^2}$$

C'est une équation linéaire relativement à *A* et *B*, qui sont eux-mêmes des fonctions (très simples) de *a* et $\omega$ . Qui plus est, les paramètres *b* et *c* n'interviennent plus directement. On est donc dans un cas typique et des plus aisés d'applicabilité de la méthode, sauf que s'agissant d'une dérivée seconde, il vaut mieux s'abstenir ! Heureusement, il n'y a pas de contre-indication à priori pour utiliser une équation intégrale dont la fonction sinusoïdale est solution. Ce n'est guère plus compliqué et donne en général des résultats largement satisfaisants (cette étude de la régression sinusoïdale a fait l'objet d'une proposition de publication en cours d'examen).

Un autre exemple, montrant très clairement le processus de calcul, est celui d'une régression appliquée à la fonction densité de probabilité de Gauss, que l'on verra traitée en détail dans la paragraphe suivant.

## 5. Cas de la fonction densité de probabilité de Gauss :

Nous considérons la fonction de densité de probabilité, à deux paramètres $\sigma$ et $\mu$, définie par :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right) \qquad [9]$$

La notation générale *y*(*x*) des paragraphes précédents se trouve donc être remplacée par *f*(*x*) en raison de la spécificité de ce cas.

L'intégration [10] conduit à l'équation intégrale [11] dont *f*(*x*) est solution :

$$\int_{x_1}^{x}(t-\mu)\,f(t)\,dt = -\sqrt{\frac{\pi}{2}}\,\sigma\left(f(x)-f(x_1)\right) \qquad [10]$$

$$\begin{cases} f(x)-f(x_1) = A\displaystyle\int_{x_1}^{x} f(t)\,dt + B\int_{x_1}^{x} t\,f(t)\,dt \\ \text{avec : } \quad A = \dfrac{\mu}{\sigma}\sqrt{\dfrac{2}{\pi}} \quad \text{et} \quad B = -\dfrac{1}{\sigma}\sqrt{\dfrac{2}{\pi}} \end{cases} \qquad [11]$$

C'est une équation intégrale linéaire avec la particularité de comporter deux intégrales simples, ce qui entre dans les extensions mentionnées à la fin du paragraphe précédent. On calcule les approximations respectives, la première étant notée $S$ avec $G(x) = 1$ et la seconde notée $T$ avec $G(x) = x$ :

$$\begin{cases} S_1 = 0 \\ S_k = S_{k-1} + \dfrac{1}{2}(f_k + f_{k-1})(x_k - x_{k-1}) \qquad k = 2 \to n \end{cases} \qquad [12]$$

$$\begin{cases} T_1 = 0 \\ T_k = T_{k-1} + \dfrac{1}{2}(x_k f_k + y_{k-1} f_{k-1})(x_k - x_{k-1}) \qquad k = 2 \to n \end{cases} \qquad [13]$$

En remplaçant $f(x_k)$ par $f_k$ , ainsi que $f(x_1)$ par $f_1$ et les intégrales par $S_k$ et $T_k$ respectivement, l'équation [11] n'est plus exactement vérifiée. On cherche à minimiser la somme des carrés des écarts :

$$\sum_{k=1}^{n} \varepsilon_k^2 = \sum_{k=1}^{n} \left( -(f_k - f_1) + A\, S_k + B\, T_k \right)^2 \qquad [14]$$

Remarquons que, si l'on avait choisi une autre borne inférieure d'intégration que $x_1$, cela aurait entraîné le changement de $f_1$, mais aussi des valeurs numériques différentes pour $S_k$ et $T_k$, le tout se compensant et ne modifiant pas le résultat final.

La relation [14] n'est autre que l'équation de base d'une régression linéaire dont on sait calculer la solution optimum $A_1$, $B_1$ :

$$\begin{pmatrix} A_1 \\ B_1 \end{pmatrix} = \begin{pmatrix} \sum (S_k)^2 & \sum S_k T_k \\ \sum S_k T_k & \sum (T_k)^2 \end{pmatrix}^{-1} \begin{pmatrix} \sum (y_k - y_1) S_k \\ \sum (y_k - y_1) T_k \end{pmatrix} \qquad [15]$$

Avec conventionnellement : $\sum \equiv \sum_{k=1}^{n}$ . On déduit ensuite $\sigma_1$ et $\mu_1$ d'après [11] :

$$\sigma_1 = -\frac{1}{B}\sqrt{\frac{2}{\pi}} \quad ; \quad \mu_1 = -\frac{A}{B} \qquad [16]$$

En résumé, le processus de calcul numérique est le suivant :

<u>Données</u> : $(x_1 , f_1)$, $(x_2 , f_2)$, …, $(x_k , f_k)$, …, $(x_n , f_n)$

- Calcul des $S_k$, équation [12]
- Calcul des $T_k$, équation [13]
- Calcul de : $\sum (S_k)^2 , \sum S_k T_k , \sum (T_k)^2 , \sum (y_k - y_1) S_k , \sum (y_k - y_1) T_k$
- Calcul de $A_1$ et $B_1$, système [15]
- Calcul de $\sigma_1$ et $\mu_1$ , relations [16]

<u>Résultat</u> : $\sigma_1$ et $\mu_1$ sont les valeurs approchées de $\sigma$ et $\mu$

Pour illustrer ce calcul ( figure 2 ), les données numériques ( Table 1 ) ont été générées de la manière suivante : Les $x_k$ ont été tirés au hasard sur la plage des abscisses considérées. A partir de valeurs "exactes" données $\sigma_e$ et $\mu_e$ , (définissant la fonction $f(x)$ dite "exacte" dont la courbe représentative est tracée en pointillés sur la figure 2 ), on a calculé les $f(x_k)$ exacts correspondants avec l'équation [9]. Ils ont été ensuite affectés de déviations dont l'amplitude a été tirée au hasard entre – et + 10% de $f(x_k)$, ce qui a donné, après arrondis, les valeurs numériques $f_k$ indiquées sur la Table 1. Cette modélisation outrancière de l'imprécision sur les ordonnées répond au souci de lisibilité de la figure, de telle sorte que les points dits "expérimentaux", figurés par des croix, soient assez éloignés de la courbe en pointillés. Dans le même esprit, un nombre exagérément faible de points a été choisi de façon à ce que les défauts soient mis en évidence sur la figure 2 par une différence nette entre les courbes "exactes" en pointillés et celles en trait plein représentatives des résultats de calculs intermédiaires et final. Le fait que les points ne soient pas répartis à intervalles constants selon les abscisses est aussi un facteur fortement aggravant la difficulté.

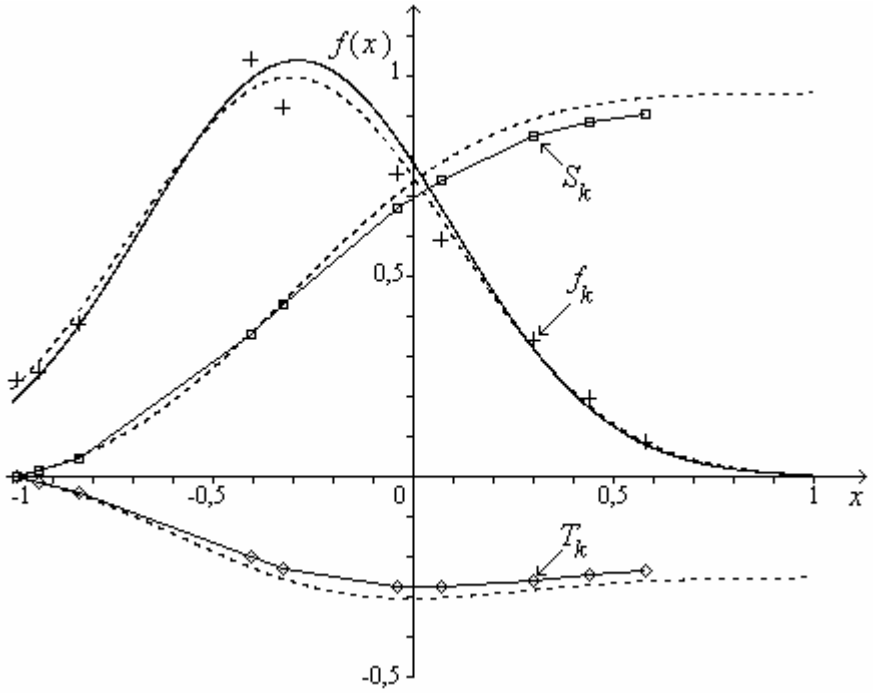



Figure 2 : Fonction densité de probabilité de Gauss, exemple de régression.

| $k$ | $x_k$ | $f_k$ | $S_k$ | $T_k$ | |
|---|---|---|---|---|---|
| 1 | -0,992 | 0,238 | 0 | 0 | |
| 2 | -0,935 | 0,262 | 0,01425 | -0,0137104 | $\sigma_e =$ 0,4 |
| 3 | -0,836 | 0,38 | 0,046029 | -0,0415616 | $\mu_e =$ -0,3 |
| 4 | -0,404 | 1,041 | 0,352965 | -0,201022 | |
| 5 | -0,326 | 0,922 | 0,429522 | -0,229147 | $\sigma_1 =$ 0,383915 |
| 6 | -0,042 | 0,755 | 0,667656 | -0,276331 | $\mu_1 =$ -0,289356 |
| 7 | 0,068 | 0,589 | 0,741576 | -0,275872 | |
| 8 | 0,302 | 0,34 | 0,850269 | -0,259172 | |
| 9 | 0,439 | 0,193 | 0,88678 | -0,246335 | |
| 10 | 0,58 | 0,083 | 0,906238 | -0,236968 | |

Table 1 : Valeurs numériques correspondantes à l'exemple de la figure 2.

Sur la figure 2, le tracé des courbes représentatives des intégrales "exactes" et des points ($x_k$ , $S_k$) et ($x_k$ , $T_k$) fait clairement apparaître la cause principale de déviations dans cette méthode de calcul : L'intégration numérique, bien que plus favorable que ne serait la dérivation, n'est pas sans défaut, loin de là. Cela entraîne des déviations sur le résultat $(\sigma_1 , \mu_1)$ .

Pour se faire une opinion objective des qualités et défauts de la méthode qui vient d'être exposée, il faudrait mener une étude expérimentale systématique sur un très grand nombre de cas et d'exemples. Ceci reste à faire, dans l'état d'avancement actuel de l'étude.

Il est à priori certain que les déviations, causées par le défaut inhérent aux intégrations numériques, seront considérablement réduites si les points sont assez nombreux et leurs abscisses réparties à intervalles assez réguliers.

**5. Commentaires :**

Il serait déraisonnable d'imaginer que la méthode présentée ici peut remplacer celles qui sont couramment utilisées, implantées dans les logiciels commerciaux et qui bénéficient d'une longue histoire d'études, d'expérimentations et de fiabilisation. On peut même prévoir avec quasi certitude que les méthodes de régression non linéaires qui ont fait leurs preuves, en travaillant par approximations successives, convergent vers un résultat plus précis qu'une méthode directe, sans calcul itératif. Alors on se demande bien quel peut être la motivation du présent travail.

Certes, en général les méthodes récursives nécessitent de connaître au départ une première approximation, au moins un ordre de grandeur, du résultat que l'on cherche. Ce n'est pas un handicap en général car le praticien ne part pas dans l'inconnu total. On pourrait penser à la méthode de régression avec équation intégrale pour, éventuellement, satisfaire ce besoin de première approximation. Mais c'est un besoin bien marginal, donc il ne faut pas voir là une motivation sérieuse.

Certes, une méthode de principe simple, aisée à programmer, telle que celle présentée ici, pourrait séduire quelques utilisateurs potentiels dans des situations particulières où l'on cherche à avoir la maîtrise totale des calculs que l'on exécute : L'utilisateur de logiciels commerciaux est bien satisfait des résultats qu'ils fournissent, mais peut parfois regretter de ne pas savoir ce que fait précisément le logiciel sophistiqué qu'il manipule. Néanmoins ce serait une piètre motivation pour la présente étude que de vouloir fournir un outil moins performant que ce qui existe, dans le seul but de répondre à un sentiment de frustration à l'usage d'outils dont on ne connaît pas exactement le mécanisme.

En fait, il faut voir dans ce papier, non pas une motivation utilitaire dans le cas spécifique de la distribution de Gauss, mais au contraire l'intention d'attirer l'attention sur une idée plus générale : les nombreuses possibilités offertes par les

équations intégrales pour transformer un problème de régression non linéaire en une régression linéaire et en déduire un processus de calcul de principe non itératif.

Il est hors de question de vouloir concurrencer ce qui a déjà l'avantage d'exister et qui mieux est, de bien fonctionner. Par contre, pour aider à résoudre de futurs problèmes, parmi les voies possibles il serait dommage d'en oublier une : celle qui fait l'objet du présent papier et dont le paragraphe 4 constitue l'essentiel de la présentation.

## ANNEXE : Listing pour les fonctions Erf et argErf :

Les valeurs approchées de Erf($x$) sont obtenues avec au moins huit chiffres significatifs après la virgule. On utilise le développement limité suivant :

$$\mathrm{Erf}(x) \simeq \frac{2\,x}{\sqrt{\pi}} \sum_{k=0}^{30} \frac{(-1)^k\, x^{2k}}{k!\,(2k+1)} \qquad \begin{cases} |x| < 2,7 \\ |\mathrm{Erf}(x)| < 0,999866 \end{cases}$$

complété par le développement limité asymptotique :

$$\mathrm{Erf}(x) \simeq \pm 1 - \frac{\mathrm{e}^{-x^2}}{x\sqrt{\pi}} \sum_{k=0}^{5} \frac{(-1)^k\,(2k+1)!!}{x^{2k}} \qquad \begin{cases} + \text{ si } x > 2,7 \ ; \ - \text{ si } x < -2,7 \\ (2k+1)!! = 1*3*\ldots*(2k+1) \\ 0,999865 < |\mathrm{Erf}(x)| < 1 \end{cases}$$

La fonction argErf($y$) est calculée par la méthode de Newton-Raphson. Le résultat argErf($y$) est obtenu avec au moins huit chiffres significatifs après la virgule si : $|y| < 0,999\,999\,999\,998 \rightarrow |\mathrm{argErf}(y)| < 5$. Au delà de ce domaine, le résultat n'est pas significatif.

Le listing (page suivante), écrit en langage Pascal, ne comporte que du vocabulaire et syntaxe élémentaires. Il ne devrait pas y avoir de difficulté pour le traduire dans tout autre langage souhaité.

On pourra le tester par comparaison des résultats du calcul avec les valeurs suivantes ( faire également le test avec les mêmes valeurs mais négatives ) :

| $x$<br>$argErf(y)$ | $Erf(x)$<br>$y$ |
|---|---|
| **0.001** | 0.001128378791 |
| **0.1** | 0.112462916 |
| **1.** | 0.8427007929 |
| **2.** | 0.995322265 |
| **2.699** | 0.9998648953 |
| **2.701** | 0.9998664351 |
| **4.** | 0.9999999846 |
| **5.** | 0.9999999999984 |

```
Function Erf(x:extended):extended;
var
   y,p:extended;
   k:integer;
begin
     y:=0;
     p:=1;
     if ((x>-2.7)and(x<2.7)) then
     begin
          for k:=0 to 30 do
          begin
               y:=y+p/(2*k+1);
               p:=-p*x*x/(k+1);
          end;
          y:=y*2*x/sqrt(pi);
     end else
     begin
          for k:=0 to 5 do
          begin
               y:=y+p;
               p:=-p*(2*k+1)/(2*x*x);
          end;
          y:=y*exp(-x*x)/(x*sqrt(pi));
          if x>0 then y:=1-y else y:=-1-y;
     end;
     Erf:=y;
end;
```

```
Function argErf(y:extended):extended;
var
   x:extended;
   k:integer;
begin
     x:=0;
     for k:=1 to 30 do
     begin
          x:=x+exp(x*x)*sqrt(pi)*(y-Erf(x))/2;
     end;
     argErf:=x;
end;
```