CHAPITRE 4

ENVIRONNEMENT INFORMATIQUE ET OUTIL DE CORRÉLATION

4.1 Environnement informatique

Le nombre d'essais pour les 8 cuves à différentes dates est de 87, générant à chaque fois 500 points ce qui donne environ 50 000 points de mesure si l'on compte les mesures fréquentielles. Un tel nombre de point ne peut être pris en charge que par un système d'information bien structuré. Pour ce faire on a opté pour la méthodologie Entité relation pour bâtir notre système d'information. La méthodologie Entité Relation a été fondée en 1970 par E.F.CODD. Elle consiste à modéliser le monde réel par des objets liés entre eux par des relations. Chaque objet se voit affecter un ensemble d'attributs dont au moins un servant à le référencer de façon unique et constituant sa clé. Les relations lient les objets via leur clé, elles peuvent aussi avoir des attributs lorsqu'elles sont porteuses d'information.

La mise en œuvre passe par :

1. La délimitation du périmètre à automatiser.
2. La construction du modèle conceptuel des données
3. l'implémentation

4.2 Périmètre à automatiser

À chaque essai les informations suivantes sont générées :

1. Le temps de vieillissement cumulé
2. Les mesures temporelles (courant de charge et de décharge vs temps)

3. Les mesures fréquentielles (tangente et capacité vs fréquence)
4. L'état des constituants (OIL PF, DP)

Elles constituent le périmètre à automatiser.

4.3 Modèle conceptuel des données "MCD"

Les entités recensées sont :

1. Les cuves
2. Les mesures temporelles
3. Les mesures fréquentielles

Les trois entités sont mises en relation via la relation expérience.

L'affectation des attributs ne se fait pas au hasard. Il est d'usage que le système construit respecte les trois premières formes normales suivantes. Pour illustrer de manière concrète ces principes, nous nous servirons de l'exemple classique suivant.

Prenons le cas d'un système de gestion de stock simple, où l'on veut consigner pour chaque article en plus de sa description, le nom et l'adresse de son fournisseur. Une approche naturelle serait de consigner toutes les informations dans une table Excel tel qu'illustré dans la figure 64.

Code Article	Description	Fournisseur	Adresse du fournisseur
KIT-A	Rouleaux thermiques	Comptoir du rouleau	180 - Bld lac Saint Jean Montral
KIT-B	Rouleaux INTERAC	Comptoir du rouleau	180 - Bld lac Saint Jean Montral
CE3	Cartouche d'encre	Comptoir des cartouches	116 - Bld Saguenay Montral
AN-10	Cartes auto nettoyantes	Comptoir des cartes	116 - Rue de Montral Iles de la Madelaine

Figure 64 Fichier articles

La première forme normale "atomicité" :

Sur la colonne "Adresse du fournisseur", on voit que la ville figure dans l'adresse et ne fait pas l'objet d'une colonne à part. Ainsi, on ne peut extraire de façon simple les fournisseurs d'une ville donnée. Cette insuffisance vient du fait que l'information adresse peut être encore décomposée.

Une relation est en première forme normale si tout attribut contient une valeur indécomposable.

La deuxième forme normale :

La colonne adresse est répétée pour le même fournisseur dans la ligne 1 et 2 de la figure 64. Cela va entrainer les trois anomalies suivantes :

1) Anomalie de stockage. Comme l'adresse est répétée plusieurs fois pour le fournisseur, de l'espace est occupé par une information redondante.

2) Anomalie de création. Si un nouveau produit fourni par le fournisseur "comptoir du rouleau", venait à être ajouté, il faudrait en plus du nom du fournisseur, ajouter même son adresse. Ce qui représente une saisie supplémentaire inutile.
3) Anomalie de mise à jour. Si le fournisseur "comptoir du rouleau", venait à changer d'adresse, il faudrait mette à jour toutes les lignes des articles commercialisés par ce fournisseur alors qu'il ne devrait y avoir qu'une seule mise à jour.
4) Anomalie de suppression. Si le produit de la ligne 4 venait à ne plus être commercialisé, on supprimerait la ligne 4. Mais en même temps on supprimerait une information importante à savoir l'adresse de ce fournisseur.

Si l'on suppose qu'un article peut être fourni par plusieurs fournisseurs et qu'un fournisseur peut fournir plusieurs articles. Alors la clé qui va identifier de façon unique une ligne c'est le couple (Code article, Fournisseur). L'adresse ne dépend que d'une partie de la clé à savoir la colonne "Fournisseur ". Cette dépendance partielle est à l'origine de toutes les anomalies ci-dessus. D'où la règle suivante qui porte le nom de deuxième forme normale.

Une relation est en deuxième forme normale si elle est en première forme normale et si chaque attribut dépend de la totalité de la clé.

La troisième forme normale :

Supposons maintenant que chaque article ne puisse être livré que par un fournisseur unique. Alors la clé qui va identifier de façon unique une ligne c'est la colonne Code article. La colonne adresse dépend de la colonne fournisseur qui ne fait pas partie de la clé. Cette dépendance d'un attribut ne faisant pas partie de la

clé, va générer les mêmes anomalies que précédemment. D'où la règle suivante qui porte le nom de troisième forme normale.

Une relation est en troisième forme normale si elle est en deuxième forme normale et si tout attribut ne dépend pas d'un attribut autre que la clé.

Un système d'information qui respecte la troisième forme normale ne présentera pas d'incohérence liée à la redondance des informations. De plus ce système ne présentera aucune anomalie de stockage, de création, de mise à jour et de suppression.

4.3.1 Implémentation du MCD

Pour l'implémentation, le choix s'est porté sur le logiciel de gestion des données Access pour les raisons suivantes :

- Le schéma relationnel obtenu par le MCD peut y être implémenté très facilement
- Une compatibilité avec MS Excel avec qui il partage le langage de programmation VBA
- La base de données obtenue est indépendante de MS ACCESS
- Le standard SQL pour l'interrogation des bases de données y est implémenté.

La figure 65, illustre le schéma relationnel issu de la modélisation de notre système d'information.

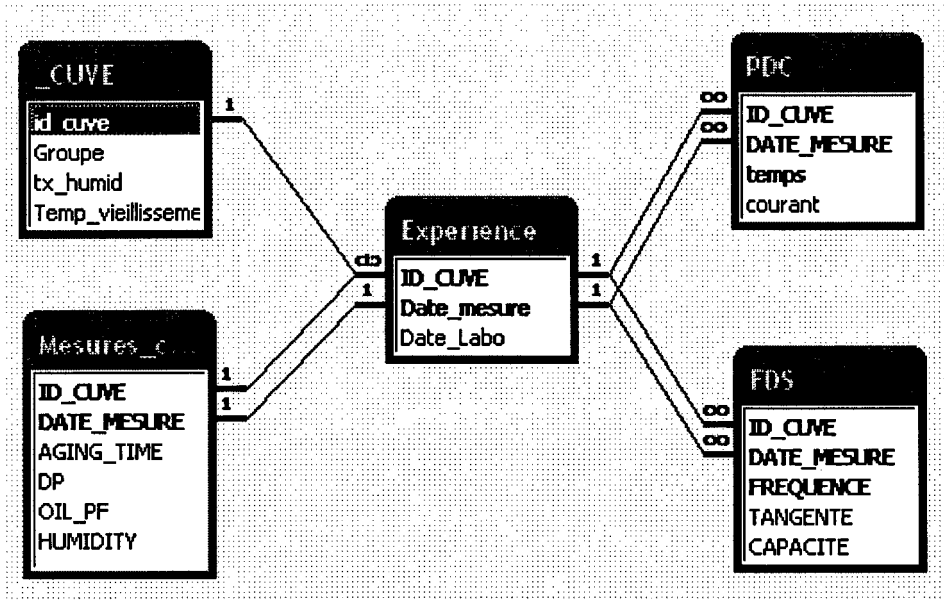

Figure 65 Modèle relationnel

On y voit clairement les entités suivantes :

- L'entité Cuve, qui contient toutes les informations propres à chaque cuve telles que le groupe de vieillissement, le taux d'humidité ainsi qu'un champ servant à référencer de façon unique chaque cuve.
- L'entité Mesures qui contient toutes les informations chimiques relevées à chaque expérience. On y retrouve l'identifiant de la cuve sur laquelle porte l'expérience, ainsi que les paramètres chimiques tels que le taux d'humidité, le degré de polymérisation. Chaque expérience est référencée par l'identifiant de la cuve correspondante ainsi que la date.
- L'entité PDC, contient les mesures du courant de charge et décharge recueillies au cours d'une expérience. Chaque mesure unitaire est référencée par l'identifiant de la cuve et la date de l'expérience ainsi que le temps où l'information est prise.

- L'entité FDS, est l'équivalent fréquentiel de l'entité PDC. Elle est en tout point identique à PDC sauf en ce qui concerne l'attribut temps qui est remplacé par celui de fréquence. Les mesures relevées sont la capacité et la tangente de l'angle des pertes.
- La relation Expérience, met en relation toutes les entités précédentes. À la lumière du schéma relationnel, on peut dire : « Une expérience c'est des données chimiques, une courbe temporelle et une courbe fréquentielle pour une cuve et à une date donnée ».

4.4 Organisation des traitements

Les traitements effectués sur notre base de données sont de deux sortes.

4.4.1 Chargement de la base de données

Les appareils de mesure génèrent leur sortie sous forme de fichier texte. Ces fichiers sont mis dans un répertoire. Une macro écrite sous VBA analyse le nom de chaque fichier, et charge son contenu dans la table appropriée.

4.4.2 Traitement de l'information

Pour la représentation graphique des données on a opté pour Excel. L'interfaçage entre la base de données se faisant soit via l'outil "couper coller", ou en intégrant un lien externe entre une requête et une plage de données sous Excel. Comme Access et Excel sont du même éditeur de logiciels, ils supportent tous les deux le langage de programmation VBA, ce qui donne une souplesse permettant de piloter la base de données à travers Excel ou vice versa.

4.4.3 Analyse de corrélation

Nous sommes ramené à écrire un programme de traitement classique à deux niveaux de rupture, qui va nous permettre de :

1. Afficher le nuage de points DP vs courant de charge à un temps donné pour toutes les cuves
2. Construire le tableau récapitulatif (temps, cuve, coefficient de corrélation).
3. Afficher le graphe montrant l'évolution de la corrélation en fonction du temps et ce pour chacune des cuves.

Concrètement la macro écrite attend en entrée :

- Une colonne contenant le premier niveau de hiérarchisation
- Une colonne contenant le second niveau hiérarchisation
- La colonne des valeurs de la variable indépendante X
- La colonne des valeurs de la variable dépendante Y

En sortie on obtient :

- Un graphe pour chaque valeur du premier niveau de hiérarchisation.
- Une courbe pour chaque valeur du second niveau de hiérarchisation.
- Un graphe montrant l'évolution de la corrélation pour chaque valeur du second niveau de hiérarchisation.

4.4.3.1 Ajustement par exponentielles décroissantes

Dans le but de rechercher d'autres corrélations, cette méthode a été appliquée sur toutes les courbes expérimentales soit un total de 87 courbes avec un total de plus de 50 000 points de mesure. Une telle opération ne peut être faite manuellement.

Comme on le verra toutes les courbes se trouvent dans une base de données relationnelle, qu'il est possible de piloter via Excel. Grace à un lien dynamique on peut faire correspondre à une feuille de calcul une requête Access, ce qui veut dire que toutes les opérations que nous faisons sur la feuille se font en réalité sur la requête. Ainsi le module VBA sous Excel choisit la cuve et la date, rafraichit le lien dynamique entre Excel et Access et lance le module d'ajustement, les résultats obtenus sont inscrits au fur et à mesure par le module de régression sous Excel dans une table Access. Le processus continue ainsi jusqu'à épuisement des 87 mesures.

4.5 Régression

4.5.1 Formulation générale du problème

Le problème de régression peut s'énoncer comme suit. Connaissant la courbe expérimentale, et ayant choisi une relation paramétrique, il faut trouver les paramètres qui minimisent la distance entre la courbe expérimentale et la courbe calculée. La distance choisie est la distance euclidienne. Si nous appelons X le vecteur des valeurs de la variable indépendante et Y le vecteur des variables dépendantes et f la relation paramétrique, alors le problème de régression linéaire revient à chercher la solution du problème suivant :

$$\begin{aligned} &\min\left(f(a_1,a_2,..a_pX)-Y\right)^t.\left(f(a_1,a_2,..a_pX)-Y\right) \\ &\text{où } (a_1,a_2,..,a_p)\in \square^{p} \end{aligned} \tag{4.1}$$

où X désigne le vecteur des valeurs de la variable indépendante, Y le vecteur des valeurs de la variable dépendante et f la fonction paramétrique ayant ($a_1 \ldots a_p$) comme paramètres réels.

4.5.2 Régression par une fonction paramétrique linéaire

Le problème devient :

$$\begin{aligned} &\min\left(aX+b-Y\right)^t.\left(aX-b-Y\right) \\ &\text{où } (a_1,b)\in \square^{2} \end{aligned} \tag{4.2}$$

Les valeurs minimales a_0 et b_0 vérifie :

$$\begin{cases} \dfrac{\partial}{\partial a}\varphi(a,b)\Big|_{(a_0,b_0)}=0 \\ \dfrac{\partial}{\partial b}\varphi(a,b)\Big|_{(a_0,b_0)}=0 \\ \varphi(a,b)=\left(aX+b-Y\right)^t.\left(aX-b-Y\right) \end{cases} \tag{4.3}$$

Après dérivation on obtient que :

$$\begin{cases} a.X.(aX+b-Y)=0 \\ (aX+b-Y)=0 \end{cases} \tag{4.4}$$

La résolution de (4.4) donne :

$$b = \frac{\text{cov}(X,Y)}{\text{var}(X)} = \frac{\sum(x_i - \bar{X})(y_i - \bar{Y})}{\sum(x_i - \bar{X})^2}$$
$$a = \bar{Y} - B.\bar{X} \tag{4.5}$$

Pour mesurer la qualité de l'alignement ou de la corrélation, on utilise le coefficient r donné par :

$$r = \frac{\sum(x_i - \bar{X})(y_i - \bar{Y})}{\sqrt{\sum(x_i - \bar{X})^2 \sum(y_i - \bar{Y})^2}} \tag{4.6}$$

4.5.3 Régression par la fonction paramétrique $Y = b.e^{a.X}$

On peut effectuer la transformation suivante :

$$Y = be^{a.X} \Leftrightarrow \ln(Y) = aX + \underbrace{\ln(b)}_{b'} \tag{4.7}$$

On est ramené à rechercher une régression linéaire entre ln(Y) et X.

Les coefficients a 'et b' sont déduit de la régression linéaire. On en déduit que :

$$a = a' \quad \text{et b} = e^{b'} \tag{4.8}$$

L'opération faite en (4.7) est rendu possible car la fonction logarithme est strictement croissante et conserve la propriété d'optimalité.

4.5.4 Régression par la fonction paramétrique $y = At^{-n}$

$$Y = aX^{-n} \Leftrightarrow \ln(Y) = \underbrace{\ln(a)}_{a'} + \underbrace{-n}_{n'}\ln(X) \tag{4.9}$$

On est ramené à rechercher une régression linéaire entre ln(Y) et ln(X).

Les coefficients a' et n' étant trouvé, les valeurs des paramètres sont :

$$a = e^{a'} \;; n = -n' \tag{4.10}$$

4.5.5 Régression non linéaire

Lorsque la fonction paramètre f est une fonction non linéaire, ses paramètres ne peuvent être trouvés analytiquement comme dans le cas linéaire. La recherche du minimum se fait de manière itérative en utilisant les nombreux algorithmes d'optimisation disponibles. Parmi les algorithmes les plus populaires, notons la descente du gradient, la matrice Hessienne et la méthode combinée de Marquardt.

4.5.5.1 Méthode du gradient «plus grande pente »

Cette méthode s'appuie sur la constatation simple qu'en un point donné, le gradient donne la direction de plus grande pente. Ce qui donne le schéma d'optimisation suivant [20] :

$$\vec{x}_{n+1} = \vec{x}_n - \eta \nabla f^T \tag{4.11}$$

Cette méthode a comme inconvénient que le pas de descente η est empirique. S'il est trop petit la convergence est trop lente, si la valeur est grande on peut tomber dans des oscillations. Le deuxième inconvénient est illustré sur la figure 66. On peut trouver un minimum qui est local. On peut cependant résoudre ce problème en choisissant avec soin la valeur initiale. Le choix de la valeur initiale peut être déterminé par la spécificité et la connaissance que l'on a du problème.

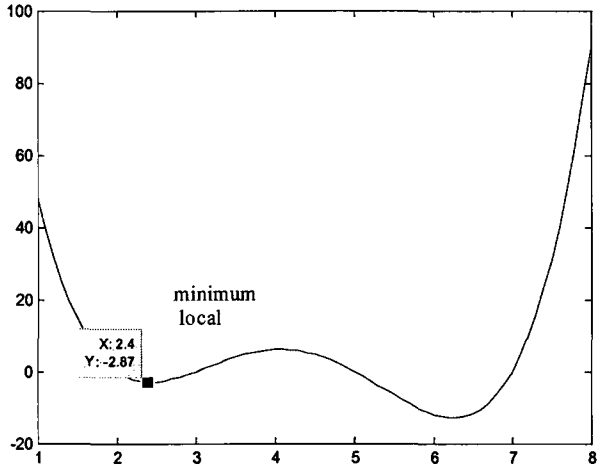

Figure 66 Minimum local

4.5.5.2 Méthode de la matrice Hessienne

Une condition nécessaire d'optimalité est que la dérivée s'annule au point optimum. Cela revient à résoudre le problème de la recherche de zéro d'une fonction.

Soit la fonction g de $\square^n$ vers $\square^n$, on cherche un x tel que g(x) = $O_{\square n}$. Le développement de Taylor à l'ordre 1 de g donne :

$$\vec{g}(x_0 + h) \approx \vec{g}(x) + \nabla g(x)\vec{h} \tag{4.12}$$

Si on veut que ce terme s'annule on doit avoir :

$$\vec{h} \approx -\nabla g(x)^{-1}\vec{g}(x) \tag{4.13}$$

Ce qui donne le schéma suivant :

$$\vec{x}_{n+1} \approx \vec{x}_n - \nabla g(x_n)^{-1}\vec{g}(x_n) \tag{4.14}$$

En remplaçant g par la dérivée, on obtient le schéma d'optimisation suivant [21] :

$$\vec{x}_{n+1} \approx \vec{x}_n - \underbrace{\nabla^2 f(x_n)^{-1}}_{H} \nabla^T f(x_n) \tag{4.15}$$

La matrice H est appelée matrice Hessienne. Cette méthode présente une erreur quadratique et une grande vitesse de convergence. Cependant, il faut calculer un inverse de matrice à chaque itération, sachant que la matrice H peut ne pas être inversible.

4.5.5.3 Méthode de Gauss Newton

La distance euclidienne entre la courbe expérimentale et la courbe paramétrique peut s'écrire :

$$\begin{aligned} & d : \square^n \longmapsto \square \\ & \quad \vec{a} \longmapsto d(\vec{a}) = \sum_1^p \left(fp(\vec{a})_i - fe_i \right)^2 \\ & (x.....x_p) : \text{points de mesure} \\ & fe_i = fe(x_i),\ fp_i = fp(x_i) \end{aligned} \tag{4.16}$$

où p désigne le nombre de points de mesure, n le nombre de paramètres, fe et fp désignent respectivement la courbe expérimentale et la courbe paramétrique.

En appliquant les règles de dérivation sur les puissances on obtient :

$$\nabla(d(a)) = \sum_1^m \nabla fp_i(\vec{a}).\left(fp(\vec{a})_i - fe_i \right) \tag{4.17}$$

En appliquant les règles de dérivation d'un produit de fonction on aura :

$$\nabla^2(d(a)) = \sum_1^m \left[\nabla fp_i(\vec{a}).\nabla fp_i(\vec{a})^T + \underbrace{\nabla^2 fp_i(\vec{a}).\left(fp(\vec{a})_i - fe_i \right)_i}_{\approx 0} \right] \tag{4.18}$$

En tenant compte qu'au voisinage de la solution optimale f(X) est nulle on obtient :

$$H(a) = \nabla^2 \left(d(a)\right) \approx \sum_{1}^{m} \nabla fp_i(a).\nabla fp_i(a)^T \tag{4.19}$$

En plus de la simplicité du calcul, la matrice qui approche la Matrice Hessienne H est symétrique et définie positive ce qui garantit l'existence de son inverse. On obtient le schéma d'optimisation suivant [22] :

$$\begin{aligned} G &= \nabla f \\ x_{n+1} &= x_n - \eta \left(G^T G\right)^{-1} G^T \end{aligned} \tag{4.20}$$