APPENDICE A :

AJUSTEMENT PAR MOINDRE CARRÉS

Soit une série de **N** points expérimentaux (x_i,y_i) avec des incertitudes σ_i. À ces données, on associe un modèle $\mathbf{F(p_k,x)}$ qui contient **n** paramètres $\mathbf{p_k}$. Une procédure d'optimisation consiste à trouver le jeu de paramètres $\mathbf{\{p_k\}_{opt}}$ qui minimise le χ^2, avec

$$\chi^2 = \sum_i \frac{1}{\sigma_i^2}\left[y_i - F(x_i)\right]^2 \quad . \tag{1}$$

Bien entendu, dans le cas où $\mathbf{F(p_k,x)}$ dépend linéairement des paramètres ajustables $\mathbf{p_k}$, une procédure de régression linéaire telle que décrite dans le cas d'un ajustement polynomial peut être appliquée. Sinon, la procédure d'optimi-sation consiste à partir d'un jeu de paramètres $\mathbf{\{p_k\}_o}$ initiaux et de les varier jusqu'à ce qu'un minimum du χ^2 ait été obtenu (voir Bevington, chap. 8). Il est à noter que toutes les méthodes utilisées ne peuvent trouver qu'un minimum local du χ^2. Il revient à l'utilisateur de s'assurer que ce minimum est effectivement global.

Une fois le jeu de paramètres optimal $\mathbf{\{p_k\}_{opt}}$ trouvé, les incertitudes $\mathbf{\Delta p_k}$ peuvent encore être obtenues de la matrice d'erreur $\boldsymbol{\varepsilon}$:

$$\Delta p_k = \sqrt{\varepsilon_{kk}} \quad , \tag{2}$$

avec

$$\varepsilon = \alpha^{-1}$$

et

$$\alpha_{ij} = \frac{1}{2}\frac{\partial^2 \chi^2}{\partial p_i \partial p_j}\bigg|_{\{p_k\}_{opt}} \cong \sum_l \frac{1}{\sigma_l^2}\frac{\partial F(x_l)}{\partial p_i}\frac{\partial F(x_l)}{\partial p_j}\bigg|_{\{p_k\}_{opt}} \quad , \tag{3}$$

où α est la matrice de courbure. Celle-ci peut être calculée analytiquement si la fonction à ajuster est suffisamment simple, ou numériquement selon :

$$\frac{\partial F}{\partial p_i} \approx \frac{F(\ldots, p_i + \Delta p_i, \ldots) - F(\ldots, p_i, \ldots)}{\Delta p_i} \quad .$$

Avant d'aborder les routines d'optimisation disponibles dans Matlab, il vaut la peine de s'attarder sur deux points fondamentaux.

i) Si le modèle $\mathbf{F(p_k,x)}$ est approprié, le chi carré normalisé $\chi^2_N = \chi^2/(N-n-1)$ doit être voisin de 1. Cela signifie tout simplement que si le modèle est approprié, l'écart entre celui-ci et les points expérimentaux devrait être de l'ordre de σ_i, en autant bien sûr que les incertitudes soient réalistes.

ii) Si les incertitudes σ_i ne sont pas a priori connues, on peut quand même effectuer l'optimisation en posant

$$\chi^2 = \sum \left[y_i - F(p_k, x_i) \right]^2$$

et $$\sigma^2 = \chi_N^2 = \chi^2 / (N - n - 1) .$$

On ne pourra par contre tester la validité du modèle puisque χ_N^2 / σ^2 vaut d'office 1.

Dans Matlab 5.3, la routine la plus simple à utiliser est ***fminunc*** (vient de «function, minimisation et unconstrained»; dans Matlab 5.2, cette fonction s'appelle ***fminu***).

Pour fixer les idées, soit le cas où on cherche à trouver les valeurs de C, L et R_L d'un circuit RLC série dont on a mesuré la courbe de résonance. Les données sont sous la forme (x_i, y_i, σ_i), où x_i est la pulsation, y_i le rapport V/E et σ_i l'incertitude sur y_i. Outre les trois paramètres ajustables C, L et R_L, on désire ajouter au modèle un quatrième paramètre fixe R, qui correspond à la valeur (connue) de la résistance sur laquelle V a été mesuré.

On commence par définir le modèle :

```
Function y=vse(p,x,r)
% p=[c*1e9,l,rl]
% c est en nanofarad pour que les valeurs à optimiser soient à
% quelques décades les unes des autres.

c=p(1)*1e-9;
l=p(2);
rl=p(3);
z=r+rl+i*(x*l-1./(x*c));
y=abs(r./z);
```

Les paramètres ajustables **doivent** être regroupés à l'intérieur d'une seule variable, ici le vecteur p, qui **doit** être le premier argument de la fonction modèle. On définit ensuite une fonction qui calcule le χ^2 :

```
Function chi2=chivse(p,x,y,sig,r)
% mettre sig=1 si sig est inconnu.

chi2= sum(((y-vse(p,x,r))./sig).^2);
```

Finalement, dans Matlab, on définit un vecteur de valeurs initiales p_o et on appelle ***fminunc*** :

```
» [p,chi2,a,a,a,h]=fminunc('chivse',p0,[ ],x,y,sig,r);
```

fminunc utilise certaines valeurs défauts contenues dans la structure *options.* Généralement, ces valeurs défauts sont convenables et on substitue la matrice vide [ ] à la structure *options* dans la liste des arguments d'entrée de ***fminunc***. La liste des paramètres qui suit cette matrice vide doit être dans le même ordre que lors de la définition de la fonction *chivse*. Finalement, ***fminunc*** retourne plusieurs diagnostics dont on peut la plupart du temps se passer. C'est pourquoi ces différentes valeurs sont stockées dans une variable *«a»* qui ne nous intéressera pas. Par contre, la matrice h, appelée «hessien» dans Matlab, vaut 2α. Les incertitudes sur les paramètres p optimisés seront donc trouvés en introduisant la fonction suivante :

```
function dp=erreur(h,chi2,p,x)
% si des incertitudes sigma ont été utilisées, seule la matrice h est
% nécessaire; sinon, il faut normaliser h.

if nargin>1
        h=h/chi2*(length(x)-length(p));
end
dp=sqrt(diag(inv(h/2)));
```

On peut bien sûr se simplifier la vie dans des cas simples en se définissant sa propre fonction d'ajustement non linéaire :

```
Function [p,dp,chi2n]=ajusnl(fonction,param0,x,y,sig,varargin)

[p,chi2,a,a,a,h]=fminunc(fonction,param0,[ ],x,y,sig,varargin{:});
if sig==1
        dp=erreur(h,chi2,p,x);
else
        dp=erreur(h);
end
chi2n=chi2/(length(x)-length(p));
```

Puis, dans Matlab,

```
» [p,dp,chi2n]=ajusnl('chivse',p0,x,y,sig,r);
```

Lorsque vous serez à l'aise avec ***fminunc***, examinez la fonction ***lsqnonlin***. Cette dernière permet de fixer des bornes à la plage de variation des différents paramètres. Elle ne retourne toutefois pas le «hessien» mais une matrice appelée «jacobien» dans Matlab. Ce «jacobien» est simplement la matrice des dérivées partielles du modèle par rapport aux paramètres d'optimisation. On peut donc en déduire la matrice de courbure α à partir de l'équation 3. Une autre routine intéressante, quoique plus complexe, est ***fmincon*** («function», «minimisation» et «constrained»). Cette routine permet d'établir des relations entre et des contraintes sur les différents paramètres.