

Feuille d'ajustement de courbe par minimisation d'un χ^2

introduction

- De nombreux laboratoires de recherche en physique utilisent le logiciel de minimisation MINUIT pour ajuster des modèles théoriques sur des données expérimentales.

Aux environs de 1970, au fur et à mesure de ses développements successifs, le logiciel MINUIT s'était toutefois déjà progressivement transformé en "usine à gaz" : un logiciel hyper complet, mais d'utilisation très lourde pour effectuer des actions basiques. La situation a encore abominablement empiré depuis (l'usine à gaz est devenue carrément "intergalactique"... les fous d'informatique peuvent le vérifier facilement avec une petite recherche sur internet : rien que pour en comprendre les rouages élémentaires, il faut se plonger dedans pendant trois jours).

Dès le début des complications, Berthon et Portes avaient choisi de simplifier la tâche des utilisateurs "ordinaires" en en programmant (en fortran) une version très basique : MINCON ; beaucoup plus simple à mettre en oeuvre, mais possédant de nombreuses fonctionnalités dont une prise en compte efficace des calculs d'incertitudes.

- Dans les années 1980, faute de logiciels équivalents sur Macintosh, j'en avais reprogrammé une version en pascal ; encore plus simplifiée, mais j'y avais joint une interface rudimentaire avec un interpréteur de formules et un traceur de graphiques (MINGRAPH ; dont le code est sur mon site).

L'évolution des ordinateurs et de leurs systèmes d'exploitation nécessite toutefois une mise à niveau incessante des logiciels. Je n'ai hélas que très peu de temps pour assurer cela. Or, bien que plusieurs logiciels récents disposent de fonctionnalités semblables, aucun (ni même Maple ou Mathematica, semble-t-il) ne gère efficacement les calculs d'incertitudes et des corrélations. L'enseignement de ces dernières s'est d'ailleurs un peu perdu, à tel point qu'il semble que de nombreux enseignants eux mêmes n'en maîtrisent plus très bien certains aspects.

- Je tente ici de mettre au point une version Maple de la procédure de minimisation MINIMI.

J M Laffaille - mars 2014

```
> restart
> # pour faciliter les mises à jour (et éviter d'encombrer les fichiers d'utilisation), Minimi est
   placée dans une bibliothèque
   # cette dernière doit être placée dans le dossier où se trouve le fichier utilisateur et chargée ici
   libname := (libname, "MinimiLib_42.mla")
libname := "/Library/Frameworks/Maple.framework/Versions/16/lib",
           "/Library/Frameworks/Maple.framework/Versions/16/toolbox/NAG/lib", ".",
           "MinimiLib_42.mla"
(1)
> # pour voir le code :
   # showstat(fonc) # cela peut servir d'exemple
   # showstat(FCN) # cela peut servir d'exemple
   # showstat(SorInt)
   # showstat(descente)
```

```

# showstat(parabole)
# showstat(errors)
# showstat(incertitudes)
# showstat(minimi)

```

fonc

```

> # cette fonction est prédéfinie dans la bibliothèque (pour compilation), mais doit être
    redéfinie ici pour l'usage souhaité
fonc := proc(NPar :: integer, p :: Array(1..30, datatype = float), x :: float) :: float;
local
    f :: float;
description "définit la fonction théorique ajustée sur les données expérimentales";

# les paramètres utilisés dans l'expression ajustée sont ici notés p[i]
# des noms plus explicites sont définis dans le programme appelant minimi, mais non
    utilisés ici
# (ceci n'a rien d'obligatoire, fonc n'est utilisé que par FCN qui peut être écrit autrement)

# le nombre maximum de paramètres est fixé à 30, ceci est imposé dans minimi
# le nombre de paramètres effectivement actifs est NPar, les suivants sont ignorés
# (NPar n'est transmis ici que pour vérification éventuelle)

# l'abscisse est notée x

f :=  $\frac{p[1]}{1 + p[2] \cdot \cos(x - p[3])}$ ; # ici on ajuste une ellipse en coordonnées polaires
return (f);
    # non nécessaire dans Maple (il retourne par défaut la dernière quantité calculée)
end proc:

```

FCN

```

> # cette fonction est prédéfinie dans la bibliothèque (pour compilation), mais peut être
    redéfinie ici pour l'usage souhaité
FCN := proc(NPar :: integer, xp :: Array(1..30, datatype = float)) :: float;
global NPts, xPts, yPts, dxPts, dyPts;

    # De façon générale, minimi ne sait pas a priori de quoi peut dépendre FCN en plus
    des paramètres ;
# ces quantités, si elles existent, ne sont donc pas transmises en arguments
# mais comme des variables globales du programme utilisateur de minimi
local
    f :: float,
    ii :: integer,
    ddd :: float;
description "définit la fonction minimisée (généralement un chi2)";
    # le nom FCN est réservé par minimi

f := 0;

```

```

# reste à définir la fonction fonc(NPar, params, abscisse) qui décrit le modèle ajusté
for ii from 1 to NPts do
# ici on ajoute quadratiquement l'incertitude de l'ordonnée des points expérimentaux...

# ...et la propagation sur l'ordonnée théorique de l'incertitude de l'abscisse des points
# expérimentaux
# (en supposant que les abscisses et les ordonnées sont des mesures indépendantes)
ddd := dyPts[ii]2
+ (  $\frac{1}{2}$  (fonc(NPar, xp, xPts[ii] + dxPts[ii] - fonc(NPar, xp, xPts[ii]
- dxPts[ii])) )2;
f := f +  $\frac{(\textit{yPts}[\textit{ii}] - \textit{fonc}(\textit{NPar}, \textit{xp}, \textit{xPts}[\textit{ii}]))^2}{ddd}$ ;
end do;
return (f);
# non nécessaire dans Maple (il retourne par défaut la dernière quantité calculée)
end proc:

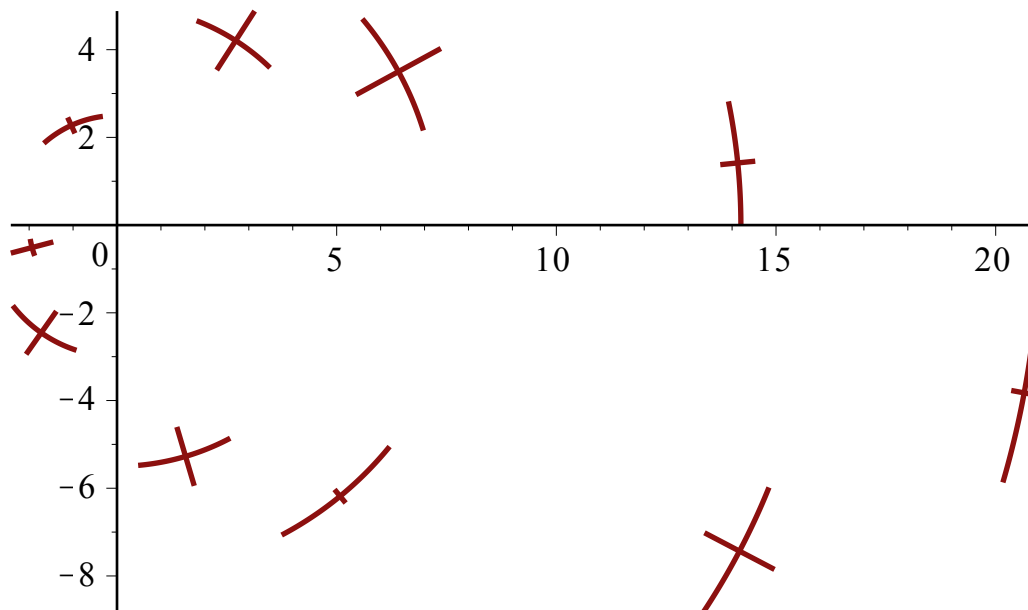
```

programme principal

```

>
> # il faut définir (ou importer) ici les points expérimentaux (abscisses x et ordonnées y) et
# leurs incertitudes
xPts := [0.1, 0.5, 1., 2., 3.4, 4.1, 5., 5.4, 5.8, 6.1] : # angle théta
yPts := [14.2, 7.3, 5., 2.5, 2., 3., 5.5, 8., 16., 21.] : # distance r( $\theta$ )
dxPts := [0.1, 0.2, 0.2, 0.3, 0.1, 0.3, 0.2, 0.2, 0.1, 0.1] :
dyPts := [0.4, 1.1, 0.8, 0.2, 0.5, 0.6, 0.7, 0.2, 0.9, 0.3] :
# il peut être prudent de vérifier le nombre de points
NPts := min(nops(xPts), nops(yPts), nops(dxPts), nops(dyPts))
NPts := 10 (4.1)
> # il faut préparer le graphique avec barres d'incertitudes (Maple ne semble pas savoir le
# faire)
# attention : avec plot, la première coordonnée du graphique est forcément r,
# correspondant à y (seul polaplot sait permuter)
gg := { } :
for ii from 1 to NPts do
gg := {op(gg), plot([yPts[ii], t, t = xPts[ii] - dxPts[ii]..xPts[ii] + dxPts[ii], coords
= polar, scaling = constrained, thickness = 2) };
gg := {op(gg), plot([t, xPts[ii], t = yPts[ii] - dyPts[ii]..yPts[ii] + dyPts[ii], coords
= polar, scaling = constrained, thickness = 2) };
end do:
plots[display](gg)

```



Maple ne sait pas définir les types "à chaud", on a prévu 30 paramètres et on indique combien sont utilisés

> # valeurs des paramètres

```
par := Array(1..30, datatype = float, fill = 0.) :
```

```
# estimations des pas d'optimisation (si zéro :
```

```
le paramètre est traité comme une constante imposée et n'est pas optimisé)
```

```
dPar := Array(1..30, datatype = float, fill = 0.) :
```

```
# noms des paramètres (utilisé dans les résultats pour mieux les reconnaître)
```

```
parN := Array(1..30, datatype = string, fill = "") :
```

```
parN[1] := "p" : par[1] := 2. : dPar[1] := 0.1 : # "paramètre" de l'ellipse
```

```
parN[2] := "e" : par[2] := 0.9 : dPar[2] := 0.1 : # excentricité
```

```
parN[3] := "θ0" : par[3] := 3. : dPar[3] := 0.1 : # angle au périégée
```

```
nnPar := 3;
```

```
# il faut indiquer sérieusement le nombre de paramètres utilisés (les suivants sont ignorés)
```

```
nnPar := 3
```

(4.2)

> stepp := 1.0 :

```
epsii := 0.0001 :
```

```
impp := 0 : # on fait comme on veut, mais on laisse ainsi minimi choisir au mieux
```

```
errorss := true :
```

> minimi(nnPar, par, parN, dPar, stepp, epsii, impp, errorss)

MINIMI (minimisation sans dérivées) MINIMI

Nombre de paramètres : 3

Nombre de paramètres effectifs : 3

Taille des pas : 1.0

Précision : .1e-3

Analyse des incertitudes pour un chi2

Valeurs initiales [Pas relatif]

p : 2. [Dp : .1]

e : .9 [De : .1]

θ0 : 3. [Dθ0 : .1]

Premier calcul de la quantité minimisée : Min = 36.079421519188

Paramètres pour le pas numéro : 4 (Min = 22.0461221226365)

p : 2.25999316609646

e : .8936

θ0 : 2.96884336504243

Paramètres pour le pas numéro : 8 (Min = 21.2459672622614)

p : 2.27912700128408

e : .892427338488456

θ0 : 2.96362151483127

Paramètres pour le pas numéro : 12 (Min = 7.13383747933046)

p : 3.01124988084132

e : .860135552056681

θ0 : 2.86774970652101

Paramètres pour le pas numéro : 16 (Min = 6.12737058199073)

p : 3.1948111968771

e : .854157258252527

θ0 : 2.87156574678941

Paramètres pour le pas numéro : 20 (Min = 4.50843633767328)

p : 3.36779369797295

e : .841674688371276

θ0 : 2.88004366138174

Paramètres pour le pas numéro : 24 (Min = 3.65904990490095)

p : 3.39201519101046
 e : .838097915175542
 θ_0 : 2.92203273642158

Paramètres pour le pas numéro : 28 (Min = 3.65847227259909)

p : 3.39535455880814
 e : .838009646327537
 θ_0 : 2.92209983054695

Paramètres pour le pas numéro : 32 (Min = 3.65845666443423)

p : 3.39542643535202
 e : .83801259047947
 θ_0 : 2.92192746266567

Paramètres pour le pas numéro : 36 (Min = 3.65845661496233)

p : 3.39541783021579
 e : .838013172777447
 θ_0 : 2.92191650103647

Paramètres pour le pas numéro : 40 (Min = 3.65845661496232)

p : 3.39541783521975
 e : .83801317262448
 θ_0 : 2.92191650107599

Paramètres pour le pas numéro : 44 (Min = 3.65845661496231)

p : 3.39541783313195
 e : .838013172729706
 θ_0 : 2.92191649966684

Paramètres pour le pas numéro : 48 (Min = 3.65845661496231)

p : 3.39541783316838
 e : .838013172727274
 θ_0 : 2.92191649970977

La minimisation est terminée

Le minimum n'a pas été amélioré de $.1e-3$ après chacune des 5 dernières étapes

La plus faible valeur est : Min = 3.65845661496231 (pour l'entrée : 286)

Calcul des incertitudes en 3 étapes

Paramètres [Déviations standard]

p : 3.39541783316838 [Dp : .302702889632306]

e : .838013172727274 [De : .0149879841153865]

Cov[e,p] : -.00440043558965444 ; Cor[e,p] : -.969919917057283

θ0 : 2.92191649970977 [Dθ0 : .0512285711445873]

Cov[θ0,p] : -.00194953708323467 ; Cor[θ0,p] : -.125719513272069

Cov[θ0,e] : 9.34905859033265e-005 ; Cor[θ0,e] : .121762179875073

Statistique de la minimisation : nombre d'entrées = 302 ; nombre de pas = 48

> # *minimi optimise les paramètres, mais indique en plus les incertitudes sur les valeurs ajustées...*

...ainsi que les taux de corrélation (entre -100% et +100%) et les covariances (corrélation × produit des incertitudes)

> # *on prépare la courbe théorique pour voir le résultat*

xPas := evalf $\left(\frac{2 \cdot \pi}{200} \right)$:

xtPts := [] :

ytPts := [] :

for i from 0 to 200 do

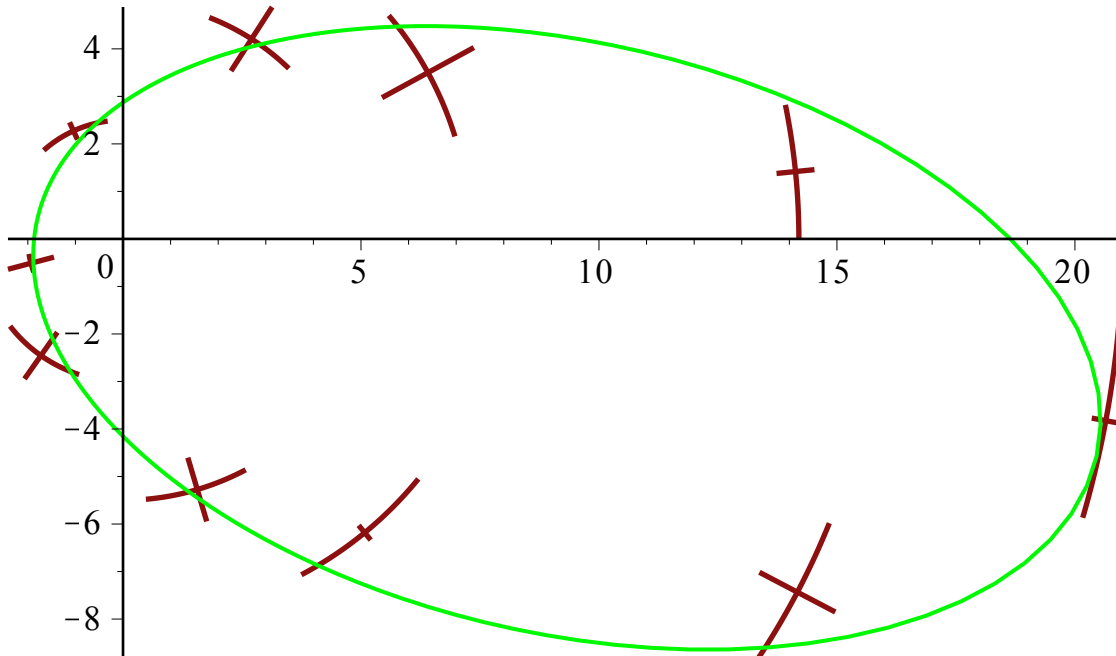
xT := i · xPas :

xtPts := [op(xtPts), xT] :

ytPts := [op(ytPts), fonc(nnPar, par, xT)] :

end do:

> *cbTh := plot(ytPts, xtPts, coords = polar, color = green) :*
plots[display](gg, cbTh);



```
> NDL := NPts - nnPar
```

```
NDL := 7
```

(4.4)

```
> chi2 := 3.65845661496231
```

```
chi2 := 3.65845661496231
```

(4.5)

```
> chi2 / NDL # il est en général souhaitable que chi2 sur NDL soit inférieur à 1
```

```
0.5226366593
```

(4.6)

```
> restart
```

```
> proba := (k, x) -> \frac{\Gamma\left(\frac{k}{2}, \frac{x}{2}\right)}{\Gamma\left(\frac{k}{2}\right)}
```

probabilité pour une valeur de chi2 avec k degrés de liberté

```
proba := (k, x) -> \frac{\Gamma\left(\frac{1}{2} k, \frac{1}{2} x\right)}{\Gamma\left(\frac{1}{2} k\right)}
```

(4.7)

```
> evalf(proba(7, 3.65845661496231))
```

0.49



0.8181681008

(4.8)