Feuille d'ajustement de courbe par minimisation d'un chi2

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 faliliter les mises à jour (et éviter d'encombrer les fichiers d'utilisation), Minimi est
      placée dans une bibliothèque
  # cette dernière doit etre 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",                    (1)
    "/Library/Frameworks/Maple.framework/Versions/16/toolbox/NAG/lib", ".",
    "MinimiLib_42.mla"
> # 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(voisinage)
# showstat(incertitudes)
# showstat(minimi)
```

fonc

```
> # cette fonction est prédéfinie dans la bibliothèque (pour compilation), mais doit etre
      redéfinie ici pour l'e modèle 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 etre 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 := p[1]·x^2 + p[2]·x + p[3]; # ici on ajuste une parabole
    return(f);
      # non nécessaire dans Maple (il retourne par défaut la dernière quantité calculée)
  end proc:
```

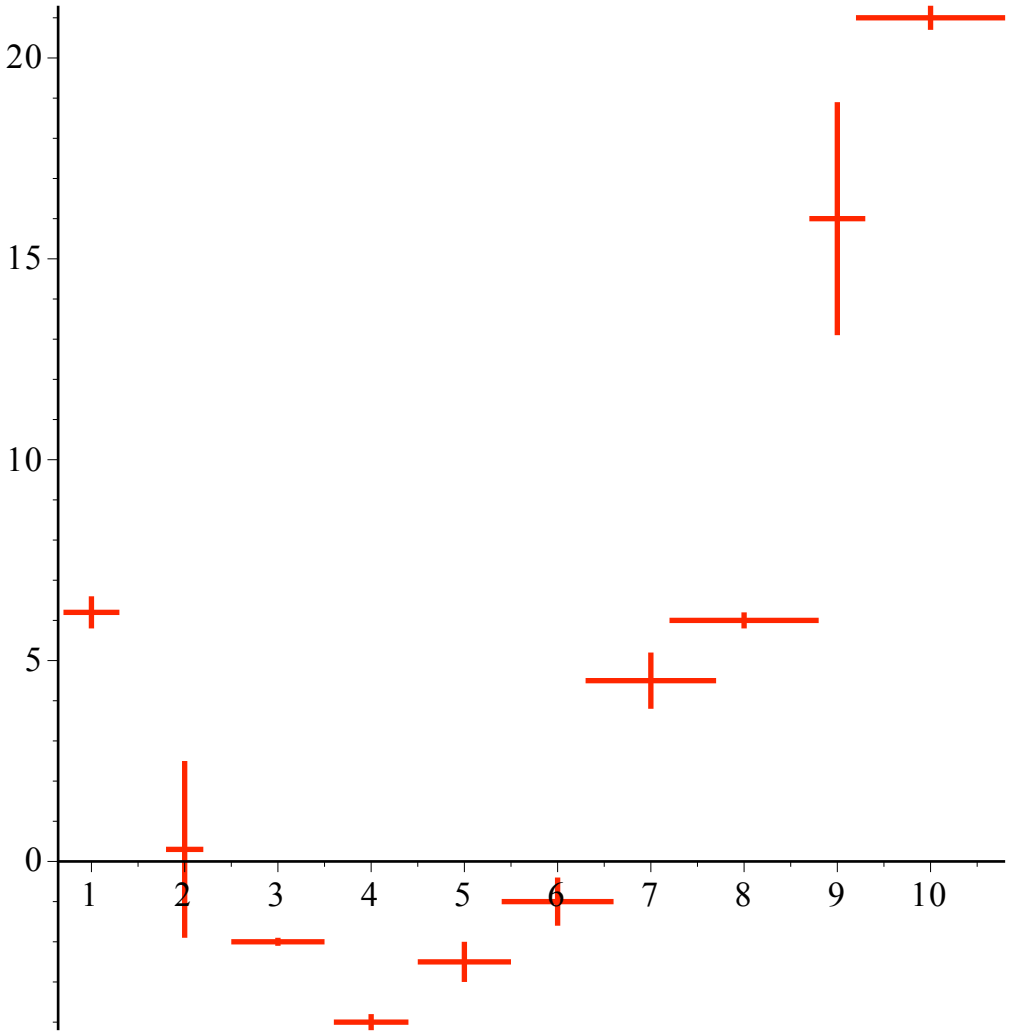
FCN

```
> # cette fonction est prédéfinie dans la bibliothèque (pour compilation), mais peut etre
      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
       + ( 1/2 (fonc(NPar, xp, xPts[ii] + dxPts[ii]) - fonc(NPar, xp, xPts[ii]
       - dxPts[ii])) )^2;
    f := f + (yPts[ii] - fonc(NPar, xp, xPts[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 := [1., 2., 3., 4., 5., 6., 7., 8., 9., 10.]:
  yPts := [6.2, 0.3, -2., -4., -2.5, -1., 4.5, 6., 16., 21.]:
  dxPts := [0.3, 0.2, 0.5, 0.4, 0.5, 0.6, 0.7, 0.8, 0.3, 0.8]:
  dyPts := [0.4, 2.2, 0.1, 0.2, 0.5, 0.6, 0.7, 0.2, 2.9, 0.3]:
  # il peut etre 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)
  bb := [ ]:
  for ii from 1 to NPts do
    bb := [op(bb), [[xPts[ii] - dxPts[ii], yPts[ii]], [xPts[ii] + dxPts[ii], yPts[ii]]]];
    bb := [op(bb), [[xPts[ii], yPts[ii] - dyPts[ii]], [xPts[ii], yPts[ii] + dyPts[ii]]]];
  end do:
  dExp := plot(bb, color = red, thickness = 2):
  plots[display](dExp)
```



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

  parN[1] := "a" : par[1] := 1. : dPar[1] := 0.1 :
  parN[2] := "b" : par[2] := -8. : dPar[2] := 0.1 :
  parN[3] := "c" : par[3] := 10. : dPar[3] := 0.1 :

  nnPar := 3;
        # il faut indiquer sérieusement le nombre de paramètres utilisés (les suivants sont
        ignorés)
```

$$nnPar := 3 \qquad \textbf{(4.2)}$$

```
> 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 **(4.3)**

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]

a : 1. [Da : .1]

b : -8. [Db : .1]

c : 10. [Dc : .1]

Premier calcul de la quantité minimisée : Min = 119.960723618308

Paramètres pour le pas numéro : 4 (Min = 17.0194057550887)

a : 1.06

b : -7.91056914641594

c : 10.4480144406427

Paramètres pour le pas numéro : 8 (Min = 2.58788828376737)

a : .905721568135798

b : -7.64912865326506

c : 12.2333733305871

Paramètres pour le pas numéro : 12 (Min = 2.58126058885274)

a : .905559495919812

b : -7.65472065504561

c : 12.2427337327968

Paramètres pour le pas numéro : 16 (Min = 2.5681087722112)

a : .912929279521931

b : -7.70920286901359

c : 12.3335522029869

Paramètres pour le pas numéro : 20 (Min = 2.55215711426972)

a : .921678606508094

b : -7.80740026003928

c : 12.593243932589

Paramètres pour le pas numéro : 24 (Min = 2.55215711351737)

a : .921678676035044

b : -7.80739842727309

c : 12.5932410716717

Paramètres pour le pas numéro : 28 (Min = 2.55215711351734)

a : .921678676180466

b : -7.80739842457576

c : 12.5932410678265

La minimisation est terminée

Le minimum n'a pas été amélioré à la dernière étape

La plus faible valeur est : Min = 2.55215711351734 (pour l'entrée : 153)

Calcul des incertitudes en 3 étapes

Paramètres   [Déviations standard]

a : .921678676180466   [Da : .103055532531597]

b : -7.80739842457576   [Db : .908459016318128]

Cov[b,a] : -.0891468163110796 ; Cor[b,a] : -.952202213009921

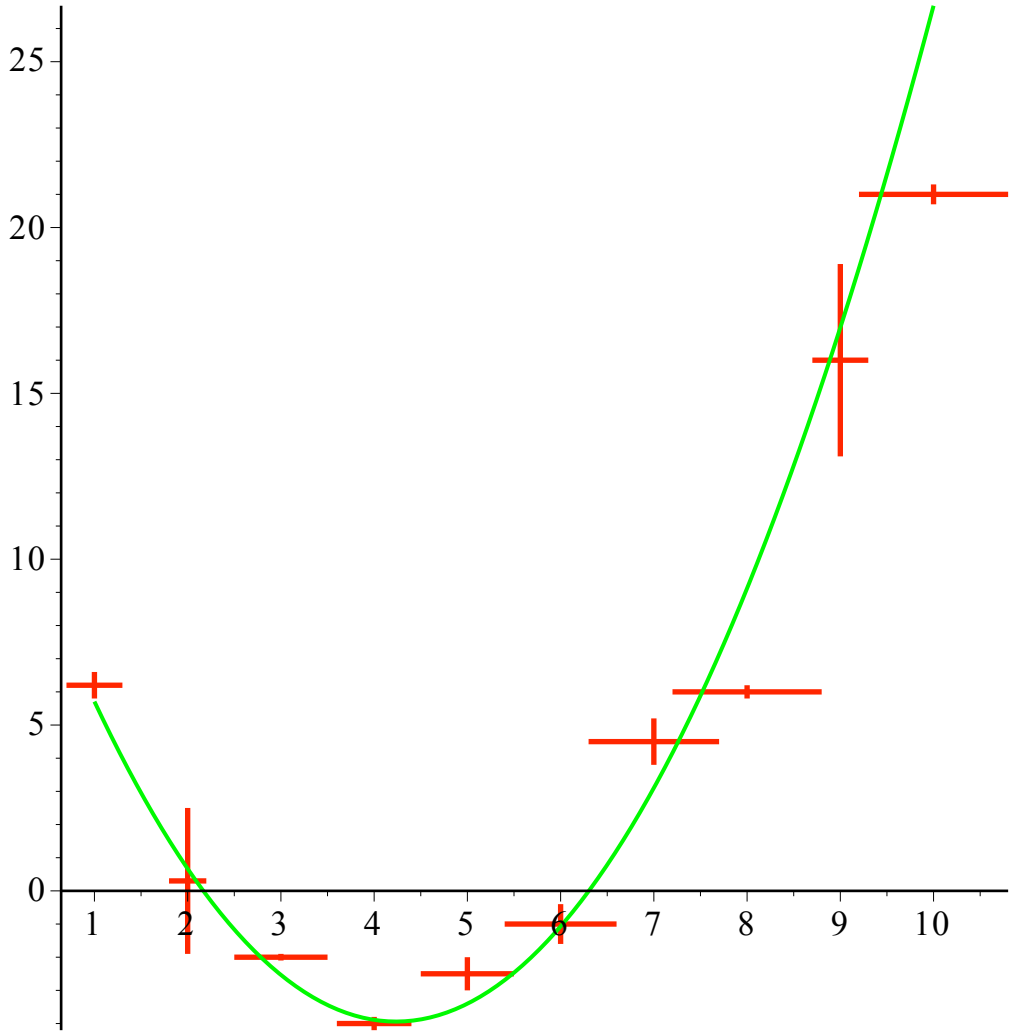
c : 12.5932410678265   [Dc : 2.12034924072015]

Cov[c,a] : .179891335506068 ; Cor[c,a] : .823249612784721

Cov[c,b] : -1.8373184650377 ; Cor[c,b] : -.95383158850563

Statistique de la minimisation : nombre d'entrées = 174 ; nombre de pas = 28

```
> # 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 := (max(xPts) - min(xPts))/100 :
  xtPts := [ ] :
  ytPts := [ ] :
  for i from 0 to 100 do
    xT := min(xPts) + i·xPas :
    xtPts := [op(xtPts), xT] :
    ytPts := [op(ytPts), fonc(nnPar, par, xT)] :
  end do:
> cbTh := plot(xtPts, ytPts, color = green) :
  plots[display](dExp, cbTh);
```



> $NDL := NPts - nnPar;$
> *# soustraire éventuellement le nombre de paramètres "bloqués" (avec dPar = 0)*

$$NDL := 7 \quad \textbf{(4.4)}$$

> $\chi 2 := 2.55215711351734$

$$\chi 2 := 2.55215711351734 \quad \textbf{(4.5)}$$

> $\frac{\chi 2}{NDL}$ *# il est en général souhaitable que chi2 sur NDL soit inférieur à 1*

$$0.3645938734 \quad \textbf{(4.6)}$$

> *restart*

> $proba := (k, x) \rightarrow \frac{\Gamma\left(\frac{k}{2}, \frac{x}{2}\right)}{\Gamma\left(\frac{k}{2}\right)}$
> *# probabilité pour une valeur de chi2 avec k degrés de liberté*

$$proba := (k, x) \rightarrow \frac{\Gamma\left(\frac{1}{2} k, \frac{1}{2} x\right)}{\Gamma\left(\frac{1}{2} k\right)} \quad \textbf{(4.7)}$$

> $evalf(proba(7, 2.55215711351734))$

$$0.9231186414 \quad \textbf{(4.8)}$$