Procédure d’ajout d’une nouvelle fonction de fit

Pour ajouter une nouvelle fonction de fit dans la bibliothèque *DataFitterLib*, il faut passer par différentes étapes :

* Créer une nouvelle classe héritant de *GenericFitter.*
* Y adapter les méthodes qui seront utilisées par la fonction générique de "fit".
* Mettre à jour les classes *FittingFunction* et *FittingFunctionFactory* pour permettre l'interfaçage du device *DataFitter* avec la nouvelle fonction.

# Introduction

Le device *DataFitter* offre une interface générique pour ajuster au mieux les paramètres d'une fonction sur un ensemble de mesures fournies (DataX et DataY). Il s'appuie sur l'algorithme générique "[Levenberg-Marquardt](https://fr.wikipedia.org/wiki/Algorithme_de_Levenberg-Marquardt)", qui utilise la bibliothèque spécialisée *DataFitterLib*. Cette dernière contient les méthodes spécifiques à chaque fonction utilisable, et s'appuie elle-même sur la bibliothèque [GSL](https://fr.wikipedia.org/wiki/GNU_Scientific_Library).

Le device *DataFitter* offre déjà une liste de paramètres possibles pour les différentes fonctions, sous forme d'attributs dynamiques :

* position
* width
* height
* background
* backgroundA
* backgroundB

Selon la forme de la fonction à ajuster, tous ces paramètres ne sont pas utilisés.

Les deux formes possibles sont:

, ou bien :

L'ajout d'une nouvelle fonction *correspondant à l'une de ces deux formes* ne nécessite donc que la mise à jour de la bibliothèque *DataFitterLib* : c'est le device qui indiquera quelle fonction employer (attribut "fittingFunctionType"), ce qui déterminera quels paramètres sont à ajuster.

# Procédure

Pour illustrer cette partie, nous nous baserons sur l'ajout de la fonction :

Afin de respecter les conventions utilisées pour les fonctions existantes, quelques règles de nommage seront détaillées au fur et à mesure de la description de la procédure.

## Ajout de la nouvelle classe

Ici, la classe à créer se nomme "***FittingFunctionKnifeEdgeWithBackground***". Elle doit être ajoutée dans **~/DataFitterLib/src** et **~/ DataFitterLib/include**.

|  |
| --- |
| ***Note***: le nom commence toujours par "***FittingFunction***", suivi du nom de la fonction (ex: "***KnifeEdge***"), et se termine par "***WithBackground***", car étant dans ce cas de la forme ***F1***(x) (cf. Introduction). Dans le cas où il se serait agi de la forme ***F2***(x), nous aurions dû employer "***WithFunctionBackground***". |

En se basant sur une autre classe du même type ( ***F1***(x) ou ***F2***(x) ), il faut mettre à jour différents éléments dans la classe "***FittingFunctionKnifeEdgeWithBackground***" :

### Fichier "FittingFunctionKnifeEdgeWithBackground.h"

* Concernant la structure définissant la fonction à ajuster :
  + *Son nom* (ex: "**KnifeEdgebFunction**") est constitué :
    - Du nom de la fonction (ex: **KnifeEdge**)
    - D'une abréviation indiquant le type de la fonction:
      * "**b**" pour "background" s'il s'agit de la forme ***F1***(x)
      * "**Fb**" pour "Function background" s'il s'agit de la forme ***F2***(x) (faisant référence à la fonction **a.x + b** ).
    - Du terme "**Function**" (faisant référence cette fois à la fonction à ajuster).
  + *Sa variable "size"* doit indiquer le nombre de paramètres (ex: 4)
  + *Son constructeur* permet de faire la liaison entre les paramètres fournis par DataFitter et les paramètres internes spécifiques à la fonction, sachant que l'ordre sera toujours le même (position, width, height, background, backgroundA, backgroundB).   
    Pour **KnifeEdge**, seuls les 4 premiers seront utilisés. Et les deux premiers seront "renommés" en interne ("position" 🡪 "mu", et "width" 🡪 "sigma").
  + *Sa fonction "operator"* correspond au calcul de la fonction : y=**KnifeEdge**(x) (qui utilise les paramètres qui sont en membre).
  + *Sa fonction "derivate"* correspond au calcul de la fonction dérivée de la précédente : dy/dx=**KnifeEdge'**(x) (qui peut utiliser les paramètres qui sont en membre).
  + *Sa fonction "partial"* correspond aux calculs des différentes dérivées partielles de la fonction, selon chacun des paramètres utilisés. Pour information, ce sont ces fonctions qui sont au cœur de l'algorithme générique utilisé par DataFitter ("[Levenberg-Marquardt](https://fr.wikipedia.org/wiki/Algorithme_de_Levenberg-Marquardt)").   
    Les résultats sont renvoyés dans un même tableau, de taille variable selon le nombre de paramètres, et sont placés dans le même ordre que celui du constructeur.

|  |
| --- |
| ***Note***: pour optimiser les calculs des fonctions décrites précédemment, il est fortement recommandé d'utiliser la bibliothèque **GSL** associée, que ce soit pour appeler des fonctions mathématiques ou employer des constantes usuelles (ex: PI, Sqrt(2), etc.). |

* Concernant la définition de la classe :
  + Elle hérite du template **GenericFitter<T>**, sur laquelle il faut adapter la structure définie précédemment (ex: **GenericFitter<*KnifeEdgebFunction*>**).

### Fichier "FittingFunctionKnifeEdgeWithBackground.cpp"

* Concernant le constructeur :
  + Initialiser la variable membre "mName" avec la description de la fonction.   
    Ex: "**KnifeEdge With Background**".
  + Initialiser la variable membre "mEquation" avec la définition de la fonction.   
    Ex: "**f(x) = Height/2 \* (1 + erf[(x - Mu)/(sqrt(2) \* Sigma)]) + Background**".
* Concernant la fonction "***computeResults***()" :
  + Calcul spécifique du FWHM.   
    Ex: "**FWHM = 2 \* Sigma \* sqrt(2 \* Ln(2))**".
* Concernant la fonction "***estimateInitialGuess***(…)" :
  + Si cela est demandé par l'utilisateur (attribut "Estimate Guess"=true), cette fonction est appelée pour calculer une estimation des paramètres à traiter. Celle-ci servira de point de départ pour l'algorithme d'ajustement.   
    Pour information, une bonne approximation permet non seulement de réduire le nombre d'itérations nécessaires pour trouver un résultat, mais réduit le risque de divergence.
  + Pour simplifier les calculs, certaines fonctions "génériques" sont disponibles dans **FittingFunction** :
    - gaussianEstimate(…),
    - gaussianEstimateWithFunctionBackground(…)
    - sigmoidEstimate(…)
  + Pour l'exemple de la fonction **KnifeEdge**(x), puisque cette fonction s'approche de la forme d'une sigmoïde, l'estimation de départ des paramètres a été basée sur la fonction ***sigmoidEstimate***(…).

## Mise-à-jour des classes existantes

### Classe "FittingFunctionFactory"

* Dans la fonction "***knownFitters***()": ajouter l'identifiant avec lequel le device **DataFitter** indiquera quelle est la fonction à ajuster. Son nom est constitué :
  + Du nom de la fonction (ex: KnifeEdge)
  + D'une abréviation indiquant le type de la fonction:
    - "b" pour "background" s'il s'agit de la forme **F1**(x)
    - "fb" pour "Function background" s'il s'agit de la forme **F2**(x) (faisant référence à la fonction a.x + b ).
  + Le tout en minuscules (ex: "***knifeedgeb***")
* Dans la fonction "***newFitter*()**": ajouter la création d'un objet de la nouvelle classe de fonction (ex: "**FittingFunctionKnifeEdgeWithBackground**"), dépendant de l'identifiant ajouté dans "***knownFitters***()".

### Classe "FittingFunction"

* Si nécessaire, il est possible d'ajouter ici des fonctions génériques utilisables par l'ensemble des fonctions spécialisées "**FittingFunctionXXX::*estimateInitialGuess***(…)", en plus de celles déjà disponibles. (cf. "***estimateInitialGuess***(…)" au §II.A.2.)

### Autres classes

Il est possible que la nouvelle fonction nécessite de nouveaux paramètres, autres que *position, width, height, background, background\_a* et *background\_b*. Dans ce cas, il faut :

* Ajouter les attributs correspondants (avec les descriptions) au niveau du device **DataFitter**,
* Mettre à jour la structure "params" dans **DataFitterTask**,
* Mettre à jour la fonction "**DataFitterTask::dofit**()", au niveau de l'envoi et de la lecture des paramètres à l'appel de "**DataFitterLib/FittingFunction::doFit(…)**".