



JMMC-MEM-2300-0002

Revision : 1.0

Date : 01/10/2003

JMMC

Groupe Modèles

RESSOURCES INFORMATIQUES

Jean-Michel Clausse (Jean-Michel.Clausse@obs-azur.fr)

OCA

Rossano Molfese

OCA

TABLE OF CONTENTS

1	<i>Introduction</i>	4
2	<i>Ressources de modèles analytiques et numériques</i>	4
2.1	Ressources présentes dans ASPRO	4
2.1.1	Modèles analytiques	4
2.1.2	Modèles numériques dans ASPRO	6
2.1.3	Commentaires	6
3	<i>Un code de calcul scientifique : SIMECA</i>	7
3.1	Présentation	7
3.2	Obtention de fichier au format FITS à partir des données issues de SIMECA	7
3.3	Commentaires	7
4	<i>Algorithmes d'ajustement</i>	8
4.1	Algorithme de Levenberg-Marquardt	8
4.2	Algorithme des Moindres carrés linéarisés itératifs	11
4.3	Commentaires	11
5	<i>Conclusion : synoptique des ressources disponibles</i>	12

1 Introduction

De la modélisation de sources à l'ajustement de données issues d'observations interférométriques - aspects principaux du projet de logiciel du groupe **JMMC-Modèles** -, de nombreuses contributions existent.

Aussi, sans être exhaustif, ce document a pour but de présenter les ressources informatiques susceptibles d'être utilisées dans le cadre du logiciel qui sera développé par le groupe.

2 Ressources de modèles analytiques et numériques

2.1 Ressources présentes dans ASPRO

2.1.1 Modèles analytiques

ASPRO dispose d'un module comportant un ensemble de modèles analytiques de distributions en intensité permettant de calculer, en fonction de plusieurs paramètres, les visibilité carrées correspondantes. Codés en langage **FORTRAN**, ces modèles décrivent des géométries différentes - dont la complexité varie - et sont donnés par les expressions générales suivantes :

Modèle de distribution en intensité (MI) en un point de coordonnées (x,y) : (1)

$$I(x,y) = I_0 \cdot f(x-x_0,y-y_0)$$

où : I_0 est l'amplitude maximale du flux,
 (x_0,y_0) sont les coordonnées de l'origine du repère d'observation R,
 (x,y) sont les coordonnées du point courant observé dans R.

Modèle de visibilité (MV), déduit de MI, en un point de coordonnées (u,v) : (2)

$$V(u,v) = I_0 \cdot \exp[-2p \cdot i \cdot (u \cdot x_0 + v \cdot y_0)] \cdot g(u,v)$$

où : i est le nombre complexe tel que $i^2 = -1$,
 (u,v) sont les coordonnées dans l'espace des fréquences spatiales,
 $g(u,v)$ est la transformée de Fourier de la fonction $f(x,y)$.

Détails des modèles analytiques disponibles dans ASPRO :

Ce paragraphe présente les différents modèles utilisables dans ASPRO ; les détails suivants sont donnés :

Nom original du module , [arguments]

Modèle de distribution en intensité

Prototype de la fonction informatique (MODEL(paramètre[1], ... , paramètre[n]))

Notations utilisées :

θ : angle d'observation du point-source de coordonnées (x,y) ;

r : distance [origine du repère – point-source] tel que $r^2 = x^2 + y^2$.

1 ► Source ponctuelle

POINT [R.A., DEC, Flux]

$I(x,y) = I_0 \cdot \delta(x - x_0, y - y_0)$, δ est la fonction de Dirac

MODEL(x,y,I₀)

2 ► Source gaussienne elliptique**E_GAUSS [R.A., DEC, Flux, Maj. diam., Min. diam., Pos. Ang.]**

$$I(r_1, r_2) = \frac{4 \cdot \ln(2) I_0}{\pi \cdot b_1 \cdot b_2} \cdot \exp \left[-4 \cdot \ln(2) \cdot \left(\left(\frac{r_1}{b_1} \right)^2 + \left(\frac{r_2}{b_2} \right)^2 \right) \right], \quad b_1 \text{ et } b_2 \text{ sont les axes de}$$

l'ellipse

MODEL(x, y, I₀, b₁, b₂, θ)**3 ► Source gaussienne circulaire****C_GAUSS [R.A., DEC, Flux, Diameter]**

$$I(r) = \frac{4 \cdot I_0}{\pi \ln(2) \cdot D^2} \cdot \exp \left[-4 \cdot \ln(2) \cdot \left(\frac{r}{D} \right)^2 \right], \quad D \text{ est le diamètre du disque}$$

MODEL(x, y, I₀, D)**4 ► Source circulaire uniforme****C_DISK [R.A., DEC, Flux, Diameter]**

$$I(\cos(\theta)) = \frac{4 \cdot I_0}{\pi \cdot D^2}, \quad D \text{ est le diamètre du disque}$$

MODEL(x, y, I₀, D)**5 ► Source annulaire****RING [R.A., DEC, Flux, Inner Diameter, Outer diameter]**

$$I(\cos(\theta)) = \frac{4 \cdot I_0}{\pi \cdot (D_e^2 - D_i^2)}$$

où : D_i est le diamètre intérieur de l'anneauD_e est le diamètre extérieur de l'anneau**MODEL(x, y, I₀, D_i, D_e)****6 ► Source exponentielle****EXPO [R.A., DEC, Flux, Diameter]**

$$I(r) = I_0 \cdot \exp \left[-2 \cdot \ln(2) \cdot \left(\frac{r}{D} \right)^2 \right], \quad D \text{ est le diamètre du disque}$$

MODEL(x, y, I₀, D)**7 ► Source en 1/r²****POWER-2 [R.A., DEC, Flux, Diameter]**

$$I(\cos(\theta)) = \frac{4 \cdot I_0}{\pi \cdot D^2} \left[\frac{1}{r} \right]^2, \quad D \text{ est le diamètre du disque}$$

MODEL(x, y, I₀, D)**8 ► Source en 1/r³****POWER-3 [R.A., DEC, Flux, Diameter]**

$$I(\cos(\theta)) = \frac{4 \cdot I_0}{\pi \cdot D^2} \left[\frac{1}{r} \right]^3, \quad D \text{ est le diamètre du disque}$$

MODEL(x, y, I₀, D)

9 ► Source elliptique uniforme**Elliptical Disk [R.A., DEC., Flux, Theta, Maj. diam., Min. diam.]**

$$I(\cos(\theta)) = \frac{4 \cdot I_0}{\pi \cdot (b_1^2 + b_2^2)}, \quad b_1 \text{ et } b_2 \text{ sont les axes de l'ellipse}$$

MODEL(x, y, I₀, θ, b₁, b₂)**10 ► Source annulaire non-résolue****U_RING [R.A., DEC., Flux, Diameter]**

$$I(r) = \frac{I}{2 \cdot \pi \cdot D} \cdot \delta(r - D), \quad \delta \text{ est la fonction de Dirac}$$

MODEL(x, y, I₀, D)**11 ► Source circulaire avec assombrissement quadratique centre-bord****LD_DISK [R.A., DEC, Flux, Diameter, 'c_u', 'c_v']**

$$I(\cos(\theta)) = I_0 \cdot (1 - c_u(1 - \cos(\theta)) - c_v(1 - \cos(\theta))^2) \quad \text{où } c_u \text{ et } c_v \text{ sont les coefficients d'assombrissement}$$

MODEL(x, y, I₀, D, c_u, c_v)**12 ► Source binaire (compagnons non-résolus)****BINARY [R.A., DEC, Flux, FluxRatio, Rho, Theta]****MODEL(x, y, I₁, rapport(I₁/I₂), ρ, θ)****Compléments :**D'autres modèles analytiques ont été proposés par **Denis MOURARD**, dont :

- disque avec surbrillance centre-bord ;
- disque avec tâches ;
- étoiles avec enveloppe ;
- étoiles avec disques ;
- étoiles avec jets ;
- étoiles à noyaux actifs avec disques et jets.

2.1.2 Modèles numériques dans ASPRO

En complément du module précédent, **ASPRO** dispose aussi d'un module permettant d'importer un modèle numérique de distribution en intensité (carte de brillance) issu d'un code scientifique externe quelconque. A la condition que le fichier représentant le modèle importé soit au format **FITS**, ce module calcule la visibilité carrée correspondante par le biais d'une transformée de Fourier.

Pour plus de précisions sur le format **FITS** (Flexible Image Transport System), voir le document suivant :

Definition of the Flexible Image Transport System (FITS)*, A&A 376, 359-380 (2001), R.J. Hanish, A. Farris, E. W. Greisen, W. D. Pence, B. M. Schlesinger, P. J. Teuben, R. W. Thompson and A Warnock III)

2.1.3 Commentaires

Les deux modules présentés dans les paragraphes précédents présentent donc un grand intérêt pour le projet de logiciel du groupe JMMC-Modèles : le premier car il fournit un ensemble important de modèles géométriques de base, le second du fait de sa

souplesse vis-à-vis de l'importation de résultats fournis par des codes de calculs scientifiques externes.

Il convient enfin de préciser d'ASPRO dispose d'une interface graphique la visualisation des visibilités carrées calculées.

3 Un code de calcul scientifique : SIMECA

3.1 Présentation

SIMECA (SIMulation d'Etoiles Chaudes et Actives), développé par **Philippe STEE** et **Jamal BITTAR** en langage *FORTRAN 77*, est un code de calcul complet, cohérent et flexible.

A partir de paramètres stellaires (PS*) tels que :

- le rayon ;
- la température effective ;
- la vitesse de rotation à l'équateur ;
- le degré d'inclinaison ;
- la densité photosphérique ;
- le rapport H/(H+He) ;
- les vitesses à la base de la photosphère ;
- le flux de masse,

SIMECA calcule les observables suivantes :

- profils de raies de l'hydrogène ;
- cartes de brillance dans les raies ou dans le continu ;
- courbes de flux pour l'étoile et pour l'enveloppe.

3.2 Obtention de fichier au format FITS à partir des données issues de SIMECA

Anthony MEILLAND a développé, sous *IDL*, un code permettant de convertir les résultats issus de SIMECA en fichiers au format FITS, format directement exploitable par les routines d'ASPRO présentées précédemment.

3.3 Commentaires

D'un point de vue algorithmique, la richesse du modèle de SIMECA peut se révéler problématique : nombreux sont les paramètres physiques (PS*) pris en compte par le code et l'ajustement d'un tel modèle à des données d'observations ne peut être possible que par le truchement d'un algorithme d'inversion performant.

En effet, la dimension de l'espace des paramètres à ajuster étant directement proportionnelle au nombre de paramètres, l'algorithme qui permettra le parcours de l'espace des paramètres de SIMECA devra réaliser un compromis entre précision (quel pas d'exploration pour tel ou tel paramètre?) et temps d'exécution.

4 Algorithmes d'ajustement

4.1 Algorithme de Levenberg-Marquardt

Le cœur du logiciel en cours de conception par le groupe JMMC-Modèles est un algorithme de recherche de minimum, développé par *Eric Thiebault*. Plus évolué qu'un algorithme de Levenberg-Marquardt classique, l'algorithme développé permet de prendre en compte un terme de régularisation nécessaire à la prise en compte d'a priori sur le modèle de l'objet et l'élimination de solutions non-physiques.

4.1.1 Rappels sur la méthode de Levenberg-Marquardt :

Soit une série de N données discrètes $Y(x_1), \dots, Y(x_N)$, en notation vectorielle : $\mathbf{Y} = \mathbf{Y}(\mathbf{X})$,

où $\mathbf{X} = [x_1, \dots, x_N]^T$ (T : transposée).

Soit le modèle \mathbf{F} à K paramètres a_1, \dots, a_K .

L'ajustement du modèle \mathbf{F} aux données \mathbf{Y} consiste à trouver le meilleur jeu de paramètres (a_1, \dots, a_K) tel que la distance vectorielle de \mathbf{F} à \mathbf{Y} soit minimale :

$$\forall X, \exists A_{opt} = [a_1, \dots, a_K]_{opt} / \min(\|F(X, A_{opt}) - Y(X)\|^2)$$

On note \mathbf{E} la fonction d'énergie définie par : $\|F - Y\|^2$, fonction de $\mathbf{A} = [a_1, \dots, a_K]^T$
 $\mathbf{E}(\mathbf{A})$ peut être approchée au point \mathbf{A}_j par un développement en séries de **Taylor** :

L'**algorithme de Newton** approche la fonction d'énergie par une fonction quadratique (développement de Taylor limité à l'ordre 2) et minimise cette fonction de manière exacte :

$$\mathbf{E}(\mathbf{A}_j) = \mathbf{E}(\mathbf{A}) + [\mathbf{A}_j - \mathbf{A}]^T \cdot \text{grad}[\mathbf{E}(\mathbf{A})] + \frac{1}{2} [\mathbf{A}_j - \mathbf{A}]^T \cdot \text{grad}[\text{grad}[\mathbf{E}(\mathbf{A})]] \cdot [\mathbf{A} - \mathbf{A}_j]$$

Le meilleur vecteur de paramètre \mathbf{A}^* est celui qui minimise la série tronquée ; \mathbf{A}^* vérifie donc :

$$\text{Grad}[\mathbf{E}(\mathbf{A})] + \text{grad}[[\text{grad}[\mathbf{E}(\mathbf{A})]] \cdot [\mathbf{A}^* - \mathbf{A}]] = 0$$

D'où, en utilisant les conditions initiales \mathbf{A}_0 :

$$\mathbf{A}^* = \mathbf{A}_0 - [\text{grad}[\text{grad}[\mathbf{E}(\mathbf{A}_0)]]^{-1} \cdot \text{grad}[\mathbf{E}(\mathbf{A}_0)]$$

L'algorithme de **Levenberg-Marquardt** (ALM) dérive de l'algorithme de **Newton**. La différence entre les deux méthodes réside dans le fait que ALM ajoute un terme dit de régularisation : sa formulation est la suivante :

$$\mathbf{A}^* = \mathbf{A}_0 - [\text{grad}[\text{grad}[\mathbf{E}(\mathbf{A}_0) + \lambda \cdot \mathbf{I}]]^{-1} \cdot \text{grad}[\mathbf{E}(\mathbf{A}_0)] \text{ (ELIN)},$$

où λ est un nombre réel positif et \mathbf{I} la matrice identité.

Le terme supplémentaire $\lambda \cdot \mathbf{I}$ est un terme dit de régularisation dans le sens où la matrice $[\text{grad}[\text{grad}[\mathbf{E}(\mathbf{A}_0) + \lambda \cdot \mathbf{I}]]$ est toujours une matrice symétrique non-singulière même si la matrice hessienne $[\text{grad}[\text{grad}[\mathbf{E}(\mathbf{A}_0)]]$ est singulière.

Autrement dit, il existe toujours une matrice inverse de $[\text{grad}[\text{grad}[\mathbf{E}(\mathbf{A}_0) + \lambda \mathbf{I}]]$ et donc toujours une solution \mathbf{A}^* , qui peut être mathématiquement exacte mais non-physique...

4.1.2 Déroulement de l'algorithme :

Etape 1 : Calculer $\mathbf{E}(\mathbf{A}_0)$, \mathbf{A}_0 étant choisi arbitrairement.

Etape 2 : Affecter une valeur faible à λ ; par exemple : $\lambda = 0.001$.

Etape 3 : Résoudre l'équation linéaire **ELIN** pour déterminer $\mathbf{A}^* = \mathbf{A}_0 + \delta \mathbf{A}$.

Test :

Si $\mathbf{E}(\mathbf{A}_0 + \delta \mathbf{A}) \geq \mathbf{E}(\mathbf{A}_0)$, augmenter la valeur de λ d'un ordre de grandeur et retour à Etape 3 ;

Sinon, diminuer la valeur de λ d'un ordre de grandeur et retour à Etape 3 en passant de $\mathbf{A}_0 + \delta \mathbf{A}$ à $\mathbf{A}_0 + 2\delta \mathbf{A}$,

Et ainsi de suite jusqu'à convergence, la condition d'arrêt de l'algorithme étant fixée préalablement par l'utilisateur (par exemple $\mathbf{E} < 0.01$).

4.1.3 Remarques sur l'algorithme de Levenberg-Marquardt :

Cet algorithme fait intervenir deux concepts mathématiques - la dérivation et l'inversion matricielle - qui ne sont pas des problèmes numériques triviaux.

4.1.3.1 Dérivation numérique.

Soit la fonction f , de variable réelle x , admettant un développement en séries de Taylor suivant :

$$f(x+h) = f(x) + h.f'(x) + 1/2.h^2.f''(x) + \dots \text{ avec } h \text{ telle que } h/x \ll 1.$$

Deux cas sont possibles :

f est une fonction définie à gauche et à droite de x (cas 1),

f est une fonction définie uniquement à droite de x (cas 2).

Cas (1) : différence centrée :

$$f(x+h) = f(x) + h.f'(x) + 1/2.h^2.f''(x) + \dots \quad (\text{a})$$

$$f(x-h) = f(x) - h.f'(x) + 1/2.h^2.f''(x) + \dots \quad (\text{b})$$

(a) - (b) $\Rightarrow f'(x) = (f(x+h) - f(x-h))/(2h)$: estimation¹ de la dérivée première.

(a) + (b) $\Rightarrow f''(x) = (f(x+h) + f(x-h) - 2f(x))/h^2$: estimation de la dérivée seconde.

Cas (2) : différence décentrée :

$$f(x+h) = f(x) + h.f'(x) + 1/2.h^2.f''(x) + \dots \quad (\text{a})$$

$$f(x+2h) = f(x) + 2h.f'(x) + 2.h^2.f''(x) + \dots \quad (\text{b})$$

4(a)-(b) $\Rightarrow f'(x) = (4.f(x+h) - f(x+2h) - 3.f(x))/(2h)$: estimation de la dérivée première

2(a)-(b) $\Rightarrow f''(x) = (2.f(x+h) - f(x+2h) - f(x))/h^2$: estimation de la dérivée seconde

¹ estimation car troncature du développement à l'ordre 2 et qu'il est possible, sinon souhaitable, de prolonger le développement à des ordres plus élevés.

Conclusion :

Le choix du pas de dérivation (h) est crucial car il conditionne la justesse de l'estimation des dérivées : en particulier, **h doit être choisi tel que x+h et x diffèrent toujours d'un nombre qui admette une représentation-machine.**

Note : Formule de dérivation numérique d'une fonction à deux paramètres :

$$d^2f/(dx^2dy) = [(f(x+h,y+h) - f(x+h,y-h)) - (f(x-h,y+h) - f(x-h,y-h))]/(4.h^2)$$

4.1.3.2 Inversion matricielle.

Une méthode de résolution d'un système linéaire de N équations à N inconnues est dite de "**Gauss avec pivot maximal**". Cette méthode consiste à mettre le système sous forme matricielle et à tenter de triangulariser la matrice le représentant en utilisant des combinaisons linéaires de ses lignes.

D'un point de vue numérique, pour obtenir la plus grande précision lors de ces calculs, c'est l'élément du système de plus grande valeur absolue qui est positionné, par le biais d'échanges des colonnes, sur la diagonale de la matrice avant combinaison de lignes.

Lorsque la matrice est triangulaire, les solutions du système s'obtiennent en partant de la dernière équation qui est du type $a.x = y$. Les valeurs ainsi trouvées sont reportées de proche en proche dans les équations précédentes.

Quelques remarques :

- 1) les solutions du système obtenu en permutant 2 lignes sont identiques à celles du système initial;
- 2) les solutions du système obtenu en effectuant une combinaison linéaire de plusieurs lignes sont identiques à celles du système initial;
- 3) les solutions du système obtenu en permutant 2 colonnes sont identiques à celles du système initial mais obtenues dans un ordre différent.

Pour triangulariser, il est nécessaire de commencer par annuler tous les coefficients de la 1ère colonne, excepté celui de la 1ère ligne. Pour cela, on réalise pour chaque ligne j, la combinaison linéaire suivante, à condition que le coefficient a_{11} soit non-nul :

$$\text{ligne}_j = \text{ligne}_j - \text{ligne}_1 \times a_{j1}/a_{11}$$

a_{11} est appelé pivot

Cette étape doit être itérée pour toutes les colonnes i ($i < N$) afin, à chaque fois, d'annuler les coefficients a_{ji} avec $i < j < N$, autrement dit :

$$\text{ligne}_j = \text{ligne}_j - \text{ligne}_i \times a_{ji}/a_{ii} \text{ avec } i < j < N$$

Pour des raisons liées à la précision des calculs (par exemple lorsque les opérations se font sur des nombres qui ne sont pas du même ordre de grandeur), le pivot choisi est le coefficient a_{ij} de valeur absolue la plus grande parmi les coefficients a_{ij} restants. Le calcul du coefficient a_{ji} est effectué ainsi :

$$a_{jk} = a_{jk} - a_{ik} \times a_{ji} / a_{ii} \text{ avec } i < j < N$$

Note :

Un pivot nul correspond alors à une sous matrice dont tous les coefficients a_{ij} sont nuls : le système est dit *indéterminé*.

3 cas se présentent :

- 1) a_{ii} est grand devant a_{ji} : la nouvelle valeur de a_{jk} sera du même ordre de grandeur que l'ancienne;
- 2) a_{ii} est très proche de a_{ji} . a_{ji}/a_{ii} est proche de 1 et la différence $a_{jk} - (a_{jk} \times a_{ji}/a_{ii})$ peut donner des résultats imprécis du fait de la représentation machine des nombres (10^{-7} et 0 ne peuvent être différenciés sur certaines machines);
- 3) a_{ii} est très petit devant a_{ji} . La nouvelle valeur de a_{jk} ne dépend pratiquement plus de l'ancienne.

Ce cas est le plus défavorable :

toutes les lignes deviennent proportionnelles car si $a_{jk} \ll (a_{jk} \times a_{ji}/a_{ii})$, le résultat informatique est $a_{jk} = - a_{ik} \times a_{ji}/a_{ii}$.

4.2 Algorithme des Moindres carrés linéarisés itératifs

Nicolas NARDETTO, avec l'aide de Philippe BERIO, a développé sous *IDL*, une procédure d'ajustement s'appuyant sur une méthode dite des **Moindres carrés linéarisés itératifs**.

Le principe de la méthode consiste en la recherche du jeu de paramètres qui minimise la différence linéaire (par opposition à la différence quadratique) entre les données et le modèle.

Avec les notations adoptées précédemment :

Modèle :

$$F(X, A) = F(X, A_0) + \Delta A \cdot \frac{\partial F(X, A)}{\partial A}$$

Grandeur minimisée :

$$E(X, A) = F(X, A) - Y(X)$$

Le programme nécessite ainsi la connaissance des grandeurs suivantes :

- valeurs du modèle au point de coordonnées **X**;
- valeurs de la dérivée du modèles au point de coordonnées **X**.

Limitations :

L'efficacité de l'algorithme présenté est décroissante avec le nombre paramètres à ajuster (nombre de paramètres du modèle). Au-delà de 2 paramètres, l'algorithme connaît des problèmes de convergence (voir études de Nicolas NARDETTO sur la détermination des diamètres angulaires d'étoiles par cette méthode).

4.3 Commentaires

La ressource présentée précédemment comporte à la fois des avantages et des inconvénients.

Avantage(s) :

Cette ressource, validée par Nicolas NARDETTO, pourrait être utilisée comme référence/point de comparaison lors des phases initiales de conception du logiciel, en

permettant la validation, sur des cas relativement simples, des algorithmes développés ;

Inconvénient(s) :

Outre le fait que ce code devrait être nécessairement traduit en langage C, la méthode utilisée semble inadaptée au logiciel que le groupe JMMC entend développer, du fait de son caractère non-généraliste : le nombre de paramètres ajustables par cet algorithme est faible - 2 au maximum - alors que la plupart des modèles envisagés en possèdent un nombre plus grand (cf. parties II et III).

5 Conclusion : synoptique des ressources disponibles

