

**UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES**

**MÉMOIRE PRÉSENTÉ À**

**L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES**

**COMME EXIGENCE PARTIELLE**

**DE LA MAÎTRISE EN ÉLECTRONIQUE INDUSTRIELLE**

**PAR**

**SHI HUANG**

**L'APPLICATION DES FONCTIONS SPLINE À UNE VARIABLE  
POUR L'ÉTALONNAGE STATIQUE DE SYSTÈMES DE MESURE**

**Avril 1997**

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

## RÉSUMÉ

Le traitement du signal mesuré dans un système de mesure consiste en deux phases: conversion et reconstitution. La conversion dans un système de mesure électrique consiste à transformer des signaux reçus d'un objet de mesure en signaux électriques standards, préférablement numériques. Ces derniers sont ensuite mis au traitement afin d'obtenir des estimés des valeurs des grandeurs à mesurer; cette opération est nommée "reconstitution de mesurandes".

La reconstitution de signaux à mesurer constitue un problème fondamental dans la métrologie. Elle consiste à estimer un signal  $x$ , qui n'est pas mesurable directement, à partir des résultats bruts du mesurage d'un autre signal  $y$ , qui est lié avec le premier de façon causative, et des grandeurs d'influence  $v$ . Pour effectuer la reconstitution, ces systèmes de mesure doivent être étalonnés. L'étalonnage consiste à identifier le modèle de la relation entre la grandeur mesurée et les résultats de la conversion et à estimer les paramètres de ce modèle. Nous envisageons une méthode numérique d'étalonnage des systèmes de mesure et de reconstitution de mesurandes; cette méthode est basée sur les fonctions spline et leurs performances en approximation des données discrètes. Elle consiste à trouver des paramètres  $p$  d'un opérateur non linéaire de la reconstitution  $\mathfrak{R}$ :

$$x = \mathfrak{R}(y, v; p) \quad (0-1)$$

où,  $x$  est le mesurande,  $y$  est le résultat brut de mesure,  $v$  est la grandeur d'influence. Les méthodes de reconstitution sont généralement basées sur certaines suppositions. Celles-ci concernent le modèle mathématique de la relation entre les signaux, l'information accessible *a priori* du signal à reconstituer et l'erreur qui perturbe les résultats de conversion  $y$  et l'influence  $v$ .

Les principaux éléments de la méthodologie de recherche proposée comportent l'élaboration et l'étude de l'algorithme de reconstitution en MATLAB et en C ainsi qu'en une synthèse des résultats de l'étude.

Les résultats essentiels de ce projet de recherche sont des algorithmes originaux de l'étalonnage et de la reconstitution de mesurandes basés sur l'application des fonctions spline pour l'estimation des paramètres de caractéristiques statiques.

L'étude des algorithmes développés permet de conclure:

- que les algorithmes donnent des résultats de qualité supérieure par rapport à d'autres méthodes comparables de l'étalonnage et de la reconstitution;

- que les algorithmes sont souples et faciles à utiliser;
- que l'exactitude des résultats de reconstitution obtenus par l'implantation MATLAB est satisfaisante pour différents niveaux d'erreur.

Ainsi, les algorithmes étudiés pourraient être utilisés dans d'autres travaux de recherche poursuivis au Laboratoire des systèmes de mesure de l'U.Q.T.R.

Les résultats de cette étude ont facilité, en particulier, le meilleur choix des algorithmes de la reconstitution pour une application spécifique. Les méthodes d'étalonnage et de reconstitution élaborées sont appliquées dans trois systèmes de mesure,

- avec un capteur ultrasonique pour mesurer la concentration d'alcool dans des solutions d'eau,
- avec un capteur à fibre-optique pour mesurer les hautes pressions,
- avec un capteur thermique à enzyme pour mesurer la concentration d'urée,

afin d'obtenir une évaluation. L'étude a été effectuée en utilisant des données synthétiques et réelles.

Pour évaluer les résultats d'étalonnage et de reconstitution, une comparaison avec les résultats obtenus par les méthodes basées sur les fonctions spline est proposée. Les critères de comparaison sont l'exactitude de reconstitution et le temps de calcul .

## REMERCIEMENTS

M. Andrzej Barwicz, professeur au Département d'ingénierie, Université du Québec à Trois-Rivières et mon directeur de mémoire, pour m'avoir donné la possibilité de travailler dans son laboratoire avec une bonne équipe et des équipements adéquats, et d'enrichir mes connaissances avec des appareils de pointe. Je le remercie pour la confiance qu'il m'a accordée et la liberté d'action qu'il m'a permise.

M. Roman Z. Morawski, professeur à l'Institut de Radioélectronique, École Polytechnique de Varsovie et mon co-directeur de mémoire, pour son appui technique et l'apport de ses idées constructives.

Messieurs Daniel Massicotte, Mohamed Ben Slima, Pierre Brouard, Louis Lemire, Leszek Szczecinski et Madame Sylvie Legendre.

M. Wojtek J. Bock, professeur au Laboratoire d'optoélectronique, Département d'informatique, Université du Québec à Hull.

Fondation du Centre d'Étude Universitaire de Trois-Rivières - Études supérieures.

Syndicat des professeurs de l'U.Q.T.R.

Conseil national de recherches du Canada (*National Research Council Canada*).

Hewlett-Packard Canada.

Je voudrais profiter de cette occasion pour remercier tous ceux qui m'ont aidé, appuyé et encouragé tout au long de mes études.

## TABLE DES MATIÈRES

RÉSUMÉ . . . . .	i
REMERCIEMENTS . . . . .	iii
TABLES DES MATIERES . . . . .	iv
LISTE DES SYMBOLES ET ABRÉVIATIONS . . . . .	ix
<b>1. INTRODUCTION . . . . .</b>	<b>1</b>
1.1. Objectifs de la recherche . . . . .	3
1.2. Problématique de la recherche . . . . .	4
1.3. Méthodologie de la recherche . . . . .	5
1.4. Structure du mémoire . . . . .	6
<b>2. SYSTÈMES DE MESURE . . . . .</b>	<b>7</b>
2.1. Notions de base . . . . .	7
2.1.1. <i>Description de la mesure</i> . . . . .	7
2.1.2. <i>Description du système de mesure</i> . . . . .	8
2.2. Sources des erreurs de mesure . . . . .	10
2.3. Étalonnage statique de systèmes de mesure . . . . .	11
2.3.1. <i>Étalonnage statique</i> . . . . .	11
2.3.2. <i>Méthodes d'étalonnage statique</i> . . . . .	11
2.4. Reconstitution de mesurandes . . . . .	13

<b>3. MÉTHODES NUMÉRIQUES D'APPROXIMATION PAR LES FONCTIONS SPLINE À UNE VARIABLE</b> . . . . .	16
3.1. Fonctions spline d'interpolation . . . . .	17
3.1.1. <i>Fonction spline parabolique d'interpolation</i> . . . . .	19
3.1.2. <i>Fonction spline cubique d'interpolation</i> . . . . .	23
3.2. Fonctions spline d'approximation de moindres carrés . . . . .	26
3.2.1. <i>Fonction spline parabolique d'approximation de moindres carrés</i> . . . . .	28
3.2.2. <i>Fonction spline cubique d'approximation de moindres carrés</i> . . . . .	30
3.3. Fonctions spline de lissage . . . . .	34
3.3.1. <i>Fonction spline parabolique de lissage</i> . . . . .	34
3.3.2. <i>Fonction spline cubique de lissage</i> . . . . .	38
3.3.3. <i>Fonction Beta-spline cubique</i> . . . . .	42
<b>4. MÉTHODES D'ÉTALONNAGE PROPOSÉES</b> . . . . .	48
4.1. Procédure PCR1 . . . . .	49
4.2. Procédure PCR2 . . . . .	50
4.3. Procédure PCR3 . . . . .	51
4.4. Procédure PCR4 . . . . .	52
<b>5. MÉTHODOLOGIE D'ÉTUDE</b> . . . . .	54
5.1. Synthèse et acquisition des données de mesure . . . . .	54
5.1.1. <i>Données synthétiques: un capteur ultrasonique pour mesurer la concentration d'alcool dans des solutions d'eau - CAS1</i> . . . . .	56
5.1.2. <i>Données synthétiques: un capteur à fibre-optique pour mesurer la haute pression - CAS2</i> . . . . .	58

5.1.3.	<i>Données synthétiques: un capteur thermique à enzyme pour mesurer la concentration d'urée - CAS3</i>	58
5.1.4.	<i>Données réelles: un capteur à fibre-optique pour mesurer les hautes pressions - CAS2</i>	58
5.2.	<i>Évaluation de la qualité de l'étalonnage</i>	60
5.2.1.	<i>Synthèse et acquisition des données pour l'évaluation</i>	60
5.2.2.	<i>Critères de l'évaluation</i>	62
5.2.3.	<i>Traitement postérieur des résultats d'évaluation</i>	63
<b>6.</b>	<b>RÉSULTATS NUMÉRIQUES DE L'ÉTUDE</b>	<b>65</b>
6.1.	<i>Programme des tests accomplis</i>	66
6.2.	<i>Résultats d'évaluation obtenus pour les données synthétiques</i>	67
6.3.	<i>Résultats d'évaluation obtenus pour les données réelles</i>	68
6.4.	<i>Discussion des résultats</i>	69
6.4.1.	<i>Influence du nombre de données sur la qualité de l'étalonnage</i>	69
6.4.2.	<i>Influence du nombre de données d'influence <math>N_v^{cal}</math></i>	71
6.4.3.	<i>Influence des erreurs sur le mesurande <math>\bar{\Delta}x^{cal}</math>, la grandeur d'influence <math>\bar{\Delta}v^{cal}</math> et le résultat brut de mesure <math>\bar{\Delta}y^{cal}</math></i>	71
6.4.4.	<i>Influence du degré de la fonction spline</i>	73
6.4.5.	<i>Sensibilité des fonctions spline au niveau d'erreur</i>	77
6.4.6.	<i>Rapidité des fonctions spline</i>	78
6.4.7.	<i>Analyse des résultats d'évaluation de la qualité de l'étalonnage pour les CAS1, CAS2 et CAS3</i>	79



---

<b>7. CONCLUSION</b> . . . . .	86
7.1. Contribution algorithmique . . . . .	87
7.2. Contribution méthodologique . . . . .	89
7.3. Application des résultats des travaux . . . . .	90
7.4. Perspectives de développements futurs . . . . .	91
7.4.1. Procédures des fonctions spline à deux variables . . . . .	91
7.4.2. Développement d'autres procédures d'approximation . . . . .	91
7.4.3. Implantation matérielle . . . . .	92
<b>ANNEXES</b> . . . . .	93
A.1. Estimation du paramètre de lissage optimal par la méthode de <i>L</i> -courbe . . . . .	93
A.1.1. Description générale de la méthode de <i>L</i> -courbe . . . . .	93
A.1.2. Méthode pour l'estimation de $\alpha_{opt}$ et sa vérification . . . . .	94
A.2. Procédures d'approximation par les fonctions spline . . . . .	100
A.2.1. <i>ParIntSplP1</i> . . . . .	100
A.2.2. <i>ParLmsSplP1</i> . . . . .	103
A.2.3. <i>ParSmoSplP1</i> . . . . .	104
A.2.4. <i>ValSplP1</i> . . . . .	107
A.2.5. <i>ParIntSplC1</i> . . . . .	108
A.2.6. <i>ParLmsSplC1</i> . . . . .	109
A.2.7. <i>ParSmoSplC1</i> . . . . .	111
A.2.8. <i>ValSplC1</i> . . . . .	115

<i>A.2.9. ParSmoSplβ1</i> . . . . .	117
<b>BIBLIOGRAPHIE</b> . . . . .	120

## LISTE DES SYMBOLES ET ABRÉVIATIONS

$x$	le mesurande scalaire, $x \in [X_{\min}, X_{\max}]$ ;	
$X_{\max}$	la valeur maximale de $x$ ;	
$X_{\min}$	la valeur minimale de $x$ ;	
$\hat{x}$	une valeur exacte de $x$ ;	
$\tilde{x}$	une valeur entachée par l'erreur de $x$ dont la valeur exacte est $\hat{x}$ ;	
$\hat{x}$	une valeur estimée de $x$ dont la valeur exacte est $\hat{x}$ ;	
$x_i^{cal}$	une valeur de $x$ utilisée pour l'étalonnage, $i = 1, \dots, N_x^{cal}$ ;	
$x_i^{val}$	une valeur de $x$ utilisée pour l'évaluation de résultats d'étalonnage, $i = 1, \dots, N_x^{val}$ ;	
$\nu$	la grandeur d'influence;	
$V_{\max}$	la valeur maximale de $\nu$ ;	
$V_{\min}$	la valeur minimale de $\nu$ ;	
$\hat{\nu}$	une valeur exacte de $\nu$ ;	
$\tilde{\nu}$	une valeur entachée par l'erreur de $\nu$ dont la valeur exacte est $\hat{\nu}$ ;	
$\hat{\nu}$	une valeur estimée de $\nu$ dont la valeur exacte est $\hat{\nu}$ ;	
$\nu_i^{cal}$	une valeur de $\nu$ utilisée pour l'étalonnage, $i = 1, \dots, N_\nu^{cal}$ ;	
$\nu_i^{val}$	une valeur de $\nu$ utilisée pour l'évaluation de résultats d'étalonnage, $i = 1, \dots, N_\nu^{val}$ ;	
$y$	le résultat brut de mesure;	
$\hat{y}$	une valeur exacte de $y$ ;	
$\tilde{y}$	une valeur entachée par l'erreur de $y$ dont la valeur exacte est $\hat{y}$ ;	
$y_{i,j}^{cal}$	une valeur de $y$ utilisée pour l'étalonnage, $i = 1, \dots, N_x^{cal}$ , $j = 1, \dots, N_\nu^{cal}$ ;	
$y_{i,j}^{val}$	une valeur de $y$ utilisée pour l'évaluation de résultats d'étalonnage, $i = 1, \dots, N_x^{val}$ , $j = 1, \dots, N_\nu^{val}$ ;	
$N_x$	le nombre des valeurs de $x$ ;	
$N_\nu$	le nombre des valeurs de $\nu$ ;	
$\hat{D}^{cal} = \{ \hat{x}_i^{cal}, \{ \hat{\nu}_j^{cal}, \hat{y}_{i,j}^{cal} \mid j = 1, \dots, N_\nu^{cal} \} \mid i = 1, \dots, N_x^{cal} \}$		les données exactes pour l'étalonnage;
$\tilde{D}_r^{cal} = \{ \tilde{x}_{i,r}^{cal}, \{ \tilde{\nu}_{j,r}^{cal}, \tilde{y}_{i,j,r}^{cal} \mid j = 1, \dots, N_\nu^{cal} \} \mid i = 1, \dots, N_x^{cal} \}$		les données entachées par les erreurs pour l'étalonnage, pour $r = 1, \dots, R$ ;
$\hat{D}^{val} = \{ \hat{x}_i^{val}, \{ \hat{\nu}_j^{val}, \hat{y}_{i,j}^{val} \mid j = 1, \dots, N_\nu^{val} \} \mid i = 1, \dots, N_x^{val} \}$		les données pour l'évaluation de résultats de l'étalonnage;

$\begin{aligned} \hat{x}^{cal} &= [\hat{x}_1^{cal}, \dots, \hat{x}_{N_x}^{cal}] \\ \hat{v}^{cal} &= [\hat{v}_1^{cal}, \dots, \hat{v}_{N_x}^{cal}] \\ \hat{Y}^{cal} &= \begin{bmatrix} y_{1,1}^{cal} & y_{1,2}^{cal} & \dots & y_{1,N_x^{val}}^{cal} \\ y_{2,1}^{cal} & y_{2,2}^{cal} & \dots & y_{2,N_x^{val}}^{cal} \\ \dots & \dots & \dots & \dots \\ y_{N_x^{val},1}^{cal} & y_{N_x^{val},2}^{cal} & \dots & y_{N_x^{val},N_x^{val}}^{cal} \end{bmatrix} \end{aligned}$	} les données exactes pour l'étalonnage;
$\begin{aligned} \tilde{x}^{cal} &= [\tilde{x}_1^{cal}, \dots, \tilde{x}_{N_x}^{cal}] \\ \tilde{v}^{cal} &= [\tilde{v}_1^{cal}, \dots, \tilde{v}_{N_x}^{cal}] \\ \tilde{Y}^{cal} &= \begin{bmatrix} y_{1,1}^{cal} & y_{1,2}^{cal} & \dots & y_{1,N_x^{val}}^{cal} \\ y_{2,1}^{cal} & y_{2,2}^{cal} & \dots & y_{2,N_x^{val}}^{cal} \\ \dots & \dots & \dots & \dots \\ y_{N_x^{val},1}^{cal} & y_{N_x^{val},2}^{cal} & \dots & y_{N_x^{val},N_x^{val}}^{cal} \end{bmatrix} \end{aligned}$	} les données entachées par les erreurs pour l'étalonnage;
$\begin{aligned} \hat{x}^{val} &= [\hat{x}_1^{val}, \dots, \hat{x}_{N_x}^{val}] \\ \hat{v}^{val} &= [\hat{v}_1^{val}, \dots, \hat{v}_{N_x}^{val}] \\ \hat{Y}^{val} &= \begin{bmatrix} y_{1,1}^{val} & y_{1,2}^{val} & \dots & y_{1,N_x^{val}}^{val} \\ y_{2,1}^{val} & y_{2,2}^{val} & \dots & y_{2,N_x^{val}}^{val} \\ \dots & \dots & \dots & \dots \\ y_{N_x^{val},1}^{val} & y_{N_x^{val},2}^{val} & \dots & y_{N_x^{val},N_x^{val}}^{val} \end{bmatrix} \end{aligned}$	} les données pour l'évaluation de résultats de l'étalonnage;
$\hat{x}^{val} = [\hat{x}_1^{val}, \dots, \hat{x}_{N_x^{val}}^{val}]^T$	} le résultat de reconstitution de $\hat{x}^{val}$ ;
$\Delta \hat{x} = \sup \{  \hat{x}_i^{val} - \hat{x}_i^{cal}  \mid i = 1, \dots, N_x^{val} \}$	} l'erreur absolue maximale de reconstitution de mesurande $x$ ;
$\Delta x_i^{cal} = \tilde{x}_i^{cal} - \hat{x}_i^{cal}$	} l'erreur absolue de $\tilde{x}_i^{cal}$ ;
$\Delta v_i^{cal} = \tilde{v}_i^{cal} - \hat{v}_i^{cal}$	} l'erreur absolue de $\tilde{v}_i^{cal}$ ;
$\Delta y_{ij}^{cal} = \tilde{y}_{ij}^{cal} - y_{ij}^{cal}$	} l'erreur absolue de $\tilde{y}_{ij}^{cal}$ ;
$\overline{\Delta v}^{cal}$	} la limite d'erreur absolue de $\tilde{v}^{cal}$ ;
$\overline{\Delta x}^{cal}$	} la limite d'erreur absolue de $\tilde{x}^{cal}$ ;

$\bar{\Delta}y^{cal}$	la limite d'erreur absolue de $y^{cal}$ ;
$y = G(x)$	la caractéristique statique du système de mesure à une variable;
$y = G(x, v)$	la caractéristique statique du système de mesure à deux variables;
$x = F(y)$	la caractéristique statique du système de mesure inverse à une variable;
$x = F(y, v)$	la caractéristique statique du système de mesure inverse à deux variables;
$y = \hat{G}(x; p)$	la fonction approximée à une variable: la fonction spline;
$x = \hat{F}(y; p)$	la fonction approximée inverse à une variable: la fonction spline inverse;

$x = [x_1   \dots   x_N]^T$	} les coordonnées des noeuds d'une fonction spline à une variable;
$y = [y_1   \dots   y_N]^T$	
$x = [x_1   \dots   x_{N_x}]$	} les coordonnées des noeuds d'une fonction spline à deux variables;
$v = [v_1   \dots   v_{N_v}]$	
$Y = \begin{bmatrix} y_{1,1} & \dots & y_{1,N_x} \\ \dots & \dots & \dots \\ y_{N_v,1} & \dots & y_{N_v,N_x} \end{bmatrix}$	
$p = [p_{i,0}, p_{i,1}, p_{i,2}   i = 1, \dots, N-1]$	le vecteur des coefficients de la fonction spline parabolique;
$p = [p_{i,0}, p_{i,1}, p_{i,2}, p_{i,3}   i = 1, \dots, N-1]$	le vecteur des coefficients de la fonction spline cubique;
$N$	le nombre des noeuds qui peuvent être utilisés dans la fonction spline;
$p$	le vecteur des paramètres d'une fonction spline $\hat{G}(x; p)$ ;
$\hat{p}_\alpha$	un estimé de $p$ correspondant à $\alpha$ ;
$s$	le paramètre de lissage, $s \in (0, 1]$ ;
$\alpha = \ln(1 - s)$	la représentation logarithmique du paramètre de lissage, $\alpha \in (-\infty, 0]$ ;
$J_\alpha[p] = (1 - e^\alpha) \left\  \{y_i^{cal} - \hat{G}(x_i^{cal}; p_i)\} \right\ _2^2 + e^\alpha \left\  \hat{G}(x_i^{cal}; p_i) \right\ _2^2$	le critère utilisé pour l'estimation des paramètres d'une fonction spline de lissage;

$f_1(\alpha) = \ln \sum_{i=1}^{N_{val}} \left[ y_i^{cal} - \hat{G}(x_i^{cal}; \hat{p}_{i,\alpha}) \right]^2$	} la description paramétrique d'une $L$ -courbe $f_2 = L(f_1)$ ;
$f_2(\alpha) = \ln \int_{x_1^{cal}}^{x_n^{cal}} \left  \frac{d^2}{dx^2} \hat{G}(x^{cal}; \hat{p}_{i,\alpha}) \right ^2 dx$	
$\tau \in [0 \sim 1]$	la variable indépendante dans la définition paramétrique d'une fonction <i>Beta</i> -spline;
$\mathbf{b}_\beta$	le vecteur des fonctions de base d'une fonction <i>Beta</i> -spline;
$\beta_1, \beta_2$	les paramètres de profil d'une fonction <i>Beta</i> -spline;
$\mathbf{W}_\beta(\beta_1, \beta_2)$	la matrice des fonctions pondérées d'une fonction <i>Beta</i> -spline;
$f(\cdot)$	une fonction;
$f'(\cdot)$	la première dérivée de $f(\cdot)$ ;
$f^{-1}(\cdot)$	l'inverse de $f(\cdot)$ ;
$\sigma_x^2$	la variance de $x$ ;
$\Delta \bar{x} = \bar{x} - \hat{x}$	l'erreur absolue de $\bar{x}$ ;
$\Delta \hat{x} = \hat{x} - \bar{x}$	l'erreur absolue de $\hat{x}$ ;
$\bar{\Delta} \bar{x} = \sup \{  \Delta \bar{x}  \mid \bar{x} \in [X_{min}, X_{max}] \}$	l'erreur absolue maximale de $\bar{x}$ ;
$\bar{\Delta} \hat{x}_i = \sup \{  \hat{x}_i^{val} - \bar{x}_i^{val}  \mid i = 1, \dots, N_x^{val} \}$	l'erreur absolue maximale de $\hat{x}$ ;
$\delta_2[\hat{x}^{val}] = \frac{\  \hat{x}^{val} - \bar{x}^{val} \ _2}{\  \hat{x}^{val} \ _2}$	l'erreur quadratique moyenne relative de l'estimé $\hat{x}^{val}$ (RMSE);
$\delta_\infty[\hat{x}^{val}] = \frac{\  \hat{x}^{val} - \bar{x}^{val} \ _\infty}{\  \hat{x}^{val} \ _\infty}$	l'erreur maximale relative de l'estimé $\hat{x}^{val}$ (MRE);
$p$	la pression;
$MPa$	Mégapascal, unité de mesure de la pression;
$U$	la tension;
$V$	Volt, unité de mesure de la tension;
$T$	la température;
$^\circ C$	degré Celsius, unité de mesure de la température;
$ParIntSplPI$	la procédure d'estimation des paramètres de la fonction spline parabolique d'interpolation à une variable;
$ParLmsSplPI$	la procédure d'estimation des paramètres de la fonction spline parabolique de moindres carrés à une variable;
$ParSmoSplPI$	la procédure d'estimation des paramètres de la fonction spline parabolique de lissage à une variable;
$ValSplPI$	la procédure d'approximation de valeur de la fonction

<i>ParIntSplC1</i>	spline parabolique à une variable; la procédure d'estimation des paramètres de la fonction spline cubique d'interpolation à une variable;
<i>ParLmsSplC1</i>	la procédure d'estimation des paramètres de la fonction spline cubique de moindres carrés à une variable;
<i>ParSmoSplC1</i>	la procédure d'estimation des paramètres de la fonction spline cubique de lissage à une variable;
<i>ValSplC1</i>	la procédure d'approximation de valeur de la fonction spline cubique à une variable;
<i>ParSmoSplβ1</i>	la procédure d'estimation des paramètres de la fonction <i>Beta</i> -spline de lissage à une variable;
<i>IntSplP1</i>	la fonction spline parabolique d'interpolation à une variable;
<i>LmsSplP1</i>	la fonction spline parabolique de moindres carrés à une variable;
<i>SmoSplP1</i>	la fonction spline parabolique de lissage à une variable;
<i>IntSplC1</i>	la fonction spline cubique d'interpolation à une variable;
<i>LmsSplC1</i>	la fonction spline cubique de moindres carrés à une variable;
<i>SmoSplC1</i>	la fonction spline cubique de lissage à une variable;
<i>SmoSplβ1</i>	la fonction <i>Beta</i> -spline de lissage à une variable.

# CHAPITRE 1.

## INTRODUCTION

La mesure est un procédé de conversion des quantités physiques du monde réel et d'identification des systèmes physiques utilisant la relation entre l'entrée et la sortie.

Les modèles des systèmes de mesure peuvent être soit non linéaires ou linéaires, soit statiques ou dynamiques:

- linéaire: le transopérateur du système de mesure est la relation linéaire;
- non linéaire: le transopérateur du système de mesure est la relation non linéaire;
- mesurage statique: le mesurage d'une grandeur dont on peut considérer que la valeur reste constante pour la durée du mesurage;
- mesurage dynamique: la détermination de la valeur instantanée d'une grandeur et, le cas échéant, de sa variation dans le temps.

Ce projet porte sur l'étalonnage et la reconstitution dans les systèmes de mesure non linéaires et statiques par les méthodes des fonctions spline. L'étalonnage et la reconstitution sont les problèmes fondamentaux du traitement des signaux. Alors, le choix d'une méthode détermine en quelque sorte la qualité des systèmes de mesure. La qualité se traduit par la synthèse de la vitesse de réponse, de l'influence d'erreur, du nombre de points d'échantillon et de l'erreur entre le signal original et le signal comme le résultat.

Le système de mesure est le système qui complète la conversion, le transfert et la reconstitution des signaux, c'est-à-dire, il convertit une quantité physique inconnue en une autre quantité physique qui peut être déterminée. Tous les signaux de mesure peuvent être caractérisés par leur nature physique et par les caractéristiques des modèles mathématiques assumés pour les analyses. La structure du système de mesure est montrée à la Fig.1-1, dans laquelle,

- $x$  est le mesurande: un signal à mesurer qui caractérise l'objet mesuré;
- $x$  est une valeur exacte du mesurande;
- $\hat{x}$  est une valeur perturbée du mesurande dont la valeur exacte est  $x$ ;



$\hat{x}$  est une valeur estimée du mesurande dont la valeur exacte est  $x$  et le résultat de reconstitution de  $x$ ;

$y$  est un résultat brut de mesure et la conversion de la nature physique du mesurande accomplie par le capteur;

$\tilde{y}$  est la valeur entachée par l'erreur de  $y$ ;

$\hat{x}$  est l'estimé du mesurande et le résultat de la reconstitution accomplie par l'instrument;

$v$  est la grandeur de l'influence;

$G$  est la fonction de transfert du système de mesure et le modèle mathématique du système.

Dans ce projet,  $G$  est exprimé en  $y = G(x, v)$ ;

$F$  est la caractéristique du système de mesure inverse. Dans ce projet,  $F$  est exprimé en  $x = F(y, v)$ .

La signification du système de mesure demeure très vaste. Il se rattache au traitement de la parole, des images, la conversion des signaux non électriques, etc.

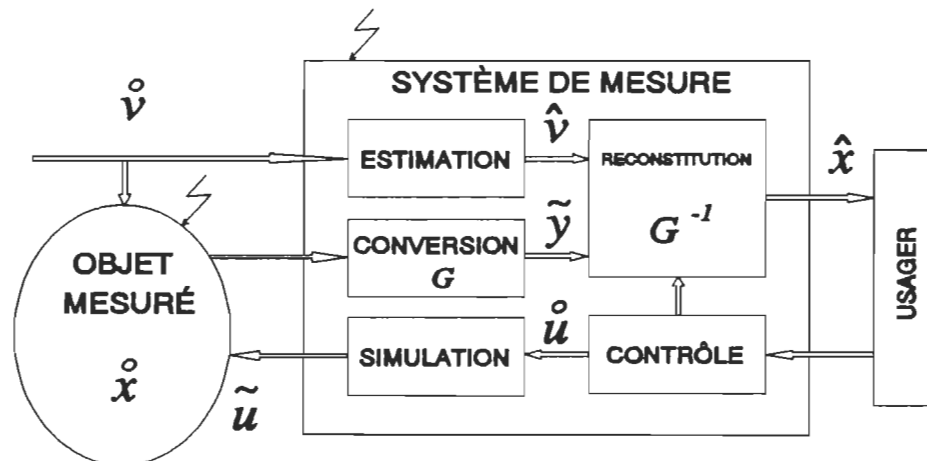


Fig.1-1. La structure du système de mesure.

Le traitement de l'information d'un objet mesuré dans un système de mesure se réalise en deux phases: conversion et reconstitution. La conversion dans un système de mesure électrique consiste en une série de transformations de signaux reçus d'un objet de mesure en signaux électriques, préférablement numériques. L'identification du modèle mathématique de la conversion est l'étalonnage du système. La reconstitution du mesurande constitue un problème fondamental dans la métrologie. Elle consiste à estimer le mesurande  $x$ , qui n'est pas mesurable directement, à partir des résultats du mesurage de  $y$ , qui est lié avec le premier de façon causative.

Les méthodes de reconstitution sont généralement basées sur certaines suppositions concernant

le modèle mathématique de la relation entre les signaux, l'information *a priori* sur le signal reconstitué ainsi que l'erreur qui entache les résultats de la conversion  $y$ . Pour effectuer la reconstitution, le système de mesure doit être étalonné. L'étalonnage consiste à identifier le modèle de la relation entre la grandeur mesurée et les résultats de la conversion et à estimer les paramètres de ce modèle. La procédure d'étalonnage doit tenir compte de l'existence des facteurs d'influence qui agissent sur la conversion.

## 1.1. Objectifs de la recherche

Les méthodes d'étalonnage et de reconstitution de signaux analytiques donnent de très bons résultats dans le cas où nous disposons suffisamment d'informations sur le modèle mathématique du système de mesure et sur les erreurs de perturbation. Cependant il est parfois impossible d'avoir ces informations et donc ces méthodes ne sont pas toujours applicables pour résoudre n'importe quel problème de reconstitution, c'est le cas des systèmes de mesure fortement non linéaires ou si l'erreur de perturbation est multiplicative. Compte tenu des problèmes inhérents aux méthodes de reconstitution analytiques, nous envisageons une méthode non analytique, basée sur les fonctions spline et leurs performances de capacité d'approximation et d'estimation.

Toutes les méthodes d'étalonnage impliquent les solutions de deux problèmes:

- choix de la structure du modèle  $y = G(x, v)$  ou l'inverse  $x = F(y, v)$ ;
- estimation des paramètres  $p$  dans le modèle.

Le premier problème est de nature heuristique, tandis que le second peut être résolu avec des algorithmes standards d'estimation de paramètres.

L'étude comparative de plusieurs algorithmes d'étalonnage est basée sur les méthodes de l'approximation de fonctions spline avec des degrés différents (parabolique et cubique), et les façons d'approximation (interpolation, moindres carrés et lissage), ou les choix des paramètres (cubique de lissage et *Beta-spline*). Le mode de fonctionnement des fonctions spline est basé sur l'utilisation de modèles des polynômes.

Nous proposons d'étudier les possibilités d'application des fonctions spline aux problèmes d'étalonnage et de reconstitution de mesurandes statiques dans un système de mesure. L'objectif de ce projet est donc de faire le lien entre deux domaines de recherche distincts: les systèmes de mesure et les fonctions spline.

Les méthodes des fonctions spline devront être appliquées dans un cas d'étalonnage et de reconstitution de mesurandes pour les systèmes de mesure avec des capteurs dont les caractéristiques statiques sont non linéaires et sensibles aux grandeurs d'influence.

L'objectif principal de ce projet est d'étudier les possibilités d'application des fonctions spline à une variable aux systèmes de mesure avec deux grandeurs d'entrée:  $x$  et  $v$ .

La méthode d'étalonnage et de reconstitution de mesurandes à l'aide des fonctions spline se base sur les performances des fonctions spline en reconnaissance de formes. L'opération d'étalonnage du système correspond à l'application des fonctions spline en deux directions, cette opération fait appel à la généralisation des fonctions spline; la reconstitution de mesurandes correspond à l'inverse des fonctions spline, cette opération fait appel à l'interpolation des fonctions spline. Cette capacité est définie comme l'aptitude à donner une réponse, bonne ou satisfaisante, à un signal qui n'a pas été appris, c'est donc la possibilité de généraliser ce qui a été appris en faisant face à une nouvelle situation. En exploitant cette capacité des fonctions spline, un mesurande doit pouvoir être reconstruit à partir de la sortie et de l'influence.

Les fonctions spline sont des outils très efficaces pour l'étalonnage et la reconstitution. L'utilisation des méthodes, leur exactitude, rapidité, complexité et adaptabilité font l'objet de discussion dans ce projet. Des exemples seront présentés pour expliquer l'utilisation de ses méthodes et une comparaison de leurs résultats sera étudiée.

## **1.2. Problématique de la recherche**

Du point de vue mathématique, le système de mesure est le regroupement des données selon certaines règles. Quand nous analysons le système de mesure, nous devons traiter tout d'abord les données. Plus les données sont nombreuses, plus l'information sur l'objet mesuré est riche, plus le système de mesure est exact. Toutefois, à cause de la limite de la vitesse et de la mémoire, nous devons tenir compte du nombre de données et de l'algorithme à utiliser.

L'utilisation des technologies électroniques et des ordinateurs permet aux systèmes de mesure d'être plus rapides, plus efficaces et plus pratiques. Dans le design d'un système de mesure moderne, il faut répondre aux questions suivants en particulier:

- Comment construire un modèle mathématique d'un système de mesure complexe?
- Comment assurer la qualité de données d'étalonnage afin d'obtenir l'exactitude demandée?
- Comment reconstituer le mesurande à partir des résultats brut de mesure?

La complexité d'un modèle mathématique ne rapporte pas nécessairement le meilleur résultat. Dans la recherche des solutions à l'étalonnage et la reconstitution d'un système de mesure, il importe que le modèle mathématique soit le plus simple possible pour inverser plus facilement la fonction de transfert, que les données discrètes soient moins nombreuses dans la mesure du possible pour sauver la mémoire du système de mesure et augmenter la vitesse de calcul.

Les méthodes de reconstitution de mesurandes sont fortement dépendantes du modèle mathématique de la relation entre  $x$ ,  $v$  et  $y$  et de l'information *a priori* sur le système de mesure. La supposition simplificatrice que le système de mesure peut être modélisé de façon linéaire n'est pas toujours justifiée. Ce qui fait que plusieurs méthodes de reconstitution ne peuvent s'appliquer. Compte tenu des problèmes inhérents aux méthodes de reconstitution analytiques, nous envisageons une méthode numérique, indépendante du modèle mathématique de la relation entre  $x$ ,  $v$  et  $y$ , basée sur les fonctions spline et leurs performances en approximation.

### 1.3. Méthodologie de la recherche

La recherche proposée sera concentrée sur les études de l'applicabilité des fonctions spline: parabolique et cubique, interpolation, moindres carrés, lissage et *Beta-spline* aux trois systèmes de mesure:

- avec un capteur ultrasonique pour la mesure de la concentration d'alcool dans les solutions d'eau (CAS1);
- avec un capteur à fibre-optique pour la mesure des hautes pressions (CAS2);
- avec un capteur thermique à enzyme pour la mesure de la concentration d'urée (CAS3).

Les caractéristiques statiques de ces capteurs sont monotones et dépendent de la température (CAS1 et CAS2) et de la valeur de pH (CAS3). L'étalonnage statique consiste dans ces cas en l'approximation des caractéristiques statiques inverses qui sont des fonctions de deux variables. Les algorithmes d'étalonnage doivent être comparés selon deux principaux critères: l'exactitude de mesure et la complexité de calcul. Un algorithme choisi (le plus exact parmi les algorithmes dont la complexité est acceptable ou le plus simple parmi les algorithmes dont l'exactitude est acceptable) doit être optimisé du point de vue de son implantation dans les systèmes de mesure, c'est-à-dire que nous devons faire un compromis entre la complexité et l'exactitude.

Nous verrons d'abord en détail ce qu'est la reconstitution et la raison de sa nécessité. Nous expliquerons ensuite l'utilité des fonctions spline pour la mesure. Nous suivrons les étapes lors de l'implantation sur MATLAB et langage C. Certains des résultats obtenus seront présentés et comparés. Nous pourrons enfin tirer les conclusions sur la faisabilité, l'exactitude, la rapidité, la sensibilité à l'erreur, l'utilité et la fiabilité des algorithmes étudiés.

Ce projet consiste en l'étude et le développement d'un algorithme pour l'estimation des paramètres d'une fonction spline approximant le mesurande. Cet algorithme est basé sur les procédures d'interpolation, de moindres carrés et de lissage correspondant au modèle mathématique de la relation entre le mesurande et le résultat brut de mesure.

L'efficacité des algorithmes sera évaluée en utilisant des données synthétiques et réelles. De plus, les résultats seront comparés aux résultats obtenus par d'autres méthodes. La validité des résultats s'étend également à d'autres cas de mesure où les modèles mathématiques de la relation  $x$ ,  $v$  et  $y$  sont semblables.

## 1.4 Structure du mémoire

Le présent projet se divise en sept chapitres distincts qui permettent de définir le problème à résoudre, de présenter les fondements de la nouvelle méthode proposée et finalement de vérifier la validité de cette dernière dans les applications.

- Chapitre 2:** définir les systèmes de mesure, expliquer les problèmes d'étalonnage statique des systèmes de mesure et de reconstitution de mesurandes, discuter des sources des erreurs et leurs effets sur le système.
- Chapitre 3:** définir et caractériser les fonctions spline différentes (*IntSplP1*, *IntSplC1*, *LmsSplP1*, *LmsSplC1*, *SmoSplP1*, *SmoSplC1* et *SmoSplβ1*).
- Chapitre 4:** expliquer les méthodes d'étalonnage et de reconstitution basées sur les fonctions spline, et appliquer la méthode d'étalonnage et de reconstitution de mesurandes dans le cas des systèmes de mesure. Quatre procédures (*PCR1*, *PCR2*, *PCR3* et *PCR4*) de l'étalonnage et de la reconstitution sont montrées.
- Chapitre 5:** définir la méthodologie d'étude des algorithmes d'étalonnage, montrer trois systèmes de mesure utilisés dans ce projet (*CAS1*, *CAS2* et *CAS3*), et expliquer l'évaluation de l'étalonnage statique des systèmes de mesure et le traitement postérieur des résultats.
- Chapitre 6:** montrer les résultats de l'étude des algorithmes de l'étalonnage et de la reconstitution de mesurandes dans le cas des systèmes de mesure étudiés avec des données synthétiques et réelles. Une comparaison entre eux est proposée afin d'évaluer la qualité des méthodes mises au point dans ce projet.
- Chapitre 7:** faire la conclusion qui est une synthèse du projet réalisé et des résultats obtenus.

## **CHAPITRE 2.**

# **SYSTÈMES DE MESURE**

Nous allons définir en quoi consiste la tâche à réaliser, c'est-à-dire l'étalonnage d'un système de mesure et la reconstitution de mesurandes. Les notions de base seront introduites sous forme de définitions, soit la *mesure*, le *système de mesure*, la *conversion*, l'*étalonnage* et la *reconstitution de mesurandes*.

Le problème d'étalonnage et de reconstitution de mesurandes sera défini du point de vue analytique dans ce chapitre. Nous avons vu que la principale difficulté pour résoudre ces opérations est due à l'erreur de conversion inévitable. De plus les méthodes numériques de reconstitution de signaux sont dépendantes du modèle mathématique de la relation entre les signaux et l'information liée au système de mesure, tels que la fonction de transfert et les erreurs qui le perturbent. La résolution est souvent sensible aux perturbations qui agissent sur le système de mesure. Les méthodes proposées dans ce projet sont dépendantes des modèles mathématiques. L'algorithme d'optimisation permet d'utiliser plusieurs points discrets pour répondre à toutes les solutions dans la gamme de mesure.

### **2.1. Notions de base**

#### **2.1.1. Description de la mesure**

De manière très générale, la mesure peut être définie comme suit:

*La mesure est un procédé opérationnel d'application des nombres aux éléments d'une certaine classe des aspects ou des caractéristiques de l'univers suivant des règles bien définies.*

Les règles sont conçues de façon à ce que le nombre appliqué à l'entité - la décrive. Une relation entre les nombres appliqués aux différents éléments de la classe correspond à une relation

empirique entre les éléments auxquels ils sont appliqués. Mathématiquement cela se traduit par la relation représentée à la Fig.2-1.

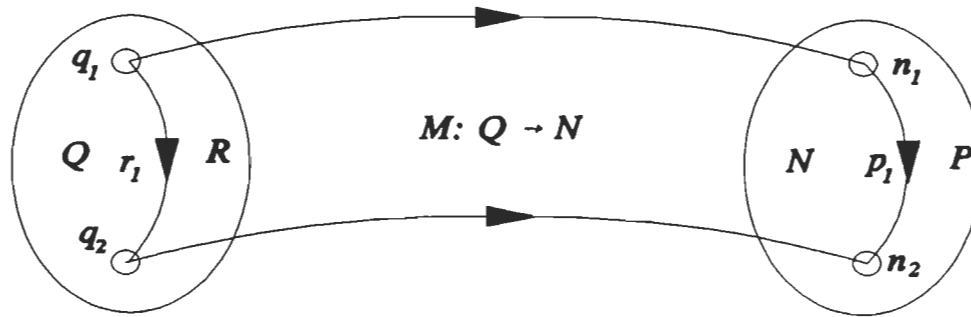


Fig.2-1 Fonction mathématique de mesure.

où,

- $Q$  est une classe bien définie des entités extra-mathématiques:  $q_1, \dots, q_n$ ;
- $R$  est un ensemble de relations empiriques:  $r_1, \dots, r_n$ ;
- $N$  est un ensemble de nombres, en général un sous-ensemble de l'ensemble des nombres réels:  $n_1, \dots, n_n$ ;
- $P$  est un ensemble de relations numériques définies dans l'ensemble des nombres:  $p_1, \dots, p_n$ ;
- $M$  est une transformation du domaine  $Q$  vers  $N$ , un homomorphisme d'un système relationnel empirique  $(Q, R)$  dans un système numérique relationnel  $(N, P)$ .

Par la transformation  $M$ , le procédé opérationnel est bien défini:

- $n_i = M(q_i)$  devient l'image de  $q_i \in Q$ ;
- $n_i$  est appelé la *mesure* de  $q_i$ ;
- $q_i$  est la *valeur mesurée*;
- $Q$  est la classe des valeurs mesurées.

### 2.1.2. Description du système de mesure

La définition précédente nous permet de concevoir toutes les mesures pouvant exister et de mieux comprendre la complexité qui résulte souvent de l'action de mesurer une grandeur physique. Une opération de mesure nécessite, généralement, que l'information qu'elle délivre soit transmise à distance du point où elle est saisie, protégée contre l'altération par des

phénomènes parasites, amplifiée, avant d'être traitée et exploitée par différents moyens. Ces fonctions sont assurées par des dispositifs ou éléments qui constituent un système dit *système de mesure*. La modélisation des systèmes de mesure a pour but de répondre à la question fondamentale: *de quelle façon la représentation d'une grandeur fournie par un instrument ressemble-t-elle à la grandeur elle-même?*. Pour répondre à cette question, les systèmes de mesure sont décomposés en une chaîne d'éléments, chacun décrit par un modèle physique adéquat et mis sous forme d'une caractéristique de transfert.

Un système de mesure peut être représenté par le schéma de la Fig.1-1, celui-ci nous permet de définir les deux opérations fondamentales associées à tout système de mesure: la *conversion* et la *reconstitution*. Soit la variable  $x$  sur le schéma une caractéristique empirique d'un objet de mesure, dont la détermination est le but de la mesure, soit  $\mathcal{X}$  le résultat obtenu en sortie de l'appareil de mesure. La transformation d'information de mesure  $x \rightarrow \mathcal{X}$  peut être décomposée en deux parties, la conversion et la reconstitution:

- la *conversion* consiste à transmettre l'information de mesure dans le domaine de phénomènes physiques qui sont faciles à interpréter ( $x \rightarrow y$ ),
- la *reconstitution* consiste à interpréter le résultat de conversion  $y$  ( $y \rightarrow \mathcal{X}$ ).

La *conversion* dans un système de mesure électrique consiste en une série de transformations de signaux reçus d'un objet de mesure, en signaux électriques standards, préférablement numériques. Ces derniers sont remis au traitement afin d'obtenir des estimés des grandeurs à mesurer; cette dernière opération est appelée *reconstitution de mesurandes*.

Les performances d'un système de mesure sont décrites par les caractéristiques métrologiques suivantes:

- *sensibilité*: quotient de l'accroissement de la réponse d'un système de mesure par l'accroissement correspondant du signal d'entrée,
- *mobilité*: l'aptitude d'un système de mesure à répondre aux petites variations de la valeur du signal d'entrée,
- *finesse*: l'aptitude d'un système de mesure à ne pas modifier la valeur de la grandeur mesurée,
- *justesse*: l'aptitude d'un système de mesure à donner des indications, qui en moyenne, correspondent à la valeur vraie ou à la valeur conventionnellement vraie de la grandeur mesurée,
- *fidélité*: l'aptitude d'un système de mesure à donner la même indication pour une même valeur de la grandeur mesurée,
- *rapidité*: un système est dit rapide s'il est capable de suivre l'évolution de la grandeur mesurée,
- *erreur (absolue) de mesure*: le résultat d'un mesurage moins valeur vraie de la grandeur mesurée.



## 2.2. Sources des erreurs de mesure

Dans la procédure de mesurage, les perturbations, les erreurs, la restriction des bits du processeur, la quantité de calcul paraissent toujours et causent des erreurs dans le résultat final. Dans tout système de mesure, il existe des sources d'erreurs. Ces sources peuvent provenir des erreurs ambiantes, du modèle mathématique utilisé pour les calculs, des capteurs de mesurandes, du convertisseur A/N, du processeur numérique utilisé, de l'imprécision des éléments employés pour traiter les signaux, de l'instabilité des caractéristiques des capteurs et autres.

Aucun instrument de mesure n'est parfait. Lors d'une mesure, le signal à mesurer (le mesurande) est déformé par l'instrument (le système de mesure). De plus, l'erreur présente dans tout système physique s'ajoute aux résultats de mesure. Ces deux phénomènes (la déformation par l'instrument et l'erreur additive) peuvent causer de grandes erreurs de mesure. Ces erreurs peuvent être réduites en utilisant les méthodes appropriées de reconstitution de mesurande. Ces dernières résident dans l'approximation de mesurande basée sur le résultat brut de mesure. Alors, des méthodes plus robustes doivent être utilisées pour la reconstitution de mesurandes.

Nous avons une méthode simple et logique de classification des erreurs pour le modèle mathématique du système de mesure  $x = \hat{F}(y, \nu; \hat{p})$  :

1. l'erreur relative de la valeur estimée de mesurande  $\delta[x]$  résulte des erreurs:
  - de la fonction approximée  $\hat{F}$ ;
  - du résultat brut de mesure entachée par l'erreur  $y$ ;
  - de la grandeur estimée d'influence  $\nu$ ;
  - des paramètres de la fonction approximée  $\hat{p}$ ;
2. l'erreur relative de la fonction approximée inverse  $\delta[\hat{F}]$  résulte du choix de la structure du modèle mathématique de la relation  $x$ ,  $\nu$  et  $y$ .
3. l'erreur relative du résultat brut de mesure entachée par l'erreur  $\delta[y]$  résulte de la qualité de conversion:
  - les paramètres métrologiques du matériel utilisé;
  - les perturbations de l'objet de mesure causées par le système de mesure;
  - les perturbations externes;
4. l'erreur de la grandeur estimée d'influence  $\delta[\nu]$  résulte de la qualité d'estimation:
  - les perturbations sur la grandeur d'influence;
  - les perturbations externes;
5. l'erreur des paramètres de la fonction approximée  $\delta[\hat{p}]$  résulte de la qualité d'estimation:

- le choix de la structure du modèle mathématique;
- la méthode d'estimation de la fonction approximée.

Quant aux erreurs considérées dans ce projet, nous accordons une attention particulière aux erreurs dans la procédure de mesure et la procédure de calcul. Nous pouvons prévoir l'amplitude d'erreur dans la procédure de mesure. Nous devons choisir une méthode pour minimiser l'erreur.

Pour ajouter les erreurs aux données exactes comme les données réelles entachées par les erreurs, les données pour l'étalonnage sont sujettes aux erreurs de mesure suivantes:

$$\left| y_{i,j}^{cal} - \hat{y}_{i,j}^{cal} \right| \leq \bar{\Delta} y^{cal}, \quad \left| x_i^{cal} - \hat{x}_i^{cal} \right| \leq \bar{\Delta} x^{cal}, \quad \left| v_{i,j}^{cal} - \hat{v}_{i,j}^{cal} \right| \leq \bar{\Delta} v^{cal} \quad (2-1)$$

où  $\bar{\Delta} y^{cal}$ ,  $\bar{\Delta} x^{cal}$  et  $\bar{\Delta} v^{cal}$  sont les limites des erreurs absolues des  $y$ ,  $x$  et  $v$ , respectivement.

## 2.3. Étalonnage statique de systèmes de mesure

### 2.3.1. Étalonnage statique

L'étalonnage est l'opération qui permet d'identifier la relation empirique reliant les résultats fournis par un capteur et le mesurande. Dans le cas d'un étalonnage statique, le temps n'a pas d'action prépondérante sur le fonctionnement du système, par opposition à un étalonnage dynamique. En nous référant à la Fig.1-1, résoudre un problème d'étalonnage statique revient à construire un modèle mathématique du bloc de conversion du système de mesure.

En situation pratique, l'*étalonnage statique* est le relevé des performances d'un système de mesure dans les conditions spécifiées suivantes: toutes les grandeurs d'entrée, désirées, interférentes, modifiantes, sont maintenues constantes à l'exception de l'une d'entre elles. Nous faisons varier cette dernière par paliers et la grandeur de sortie est relevée lorsque tout effet transitoire a disparu. L'énoncé d'une caractéristique de transfert statique n'a donc de valeur que lorsqu'il est accompagné des conditions spécifiées de l'étalonnage statique correspondant. En particulier, les valeurs des grandeurs maintenues constantes doivent être données.

### 2.3.2. Méthodes d'étalonnage statique

En considérant le système proposé à la Fig.2-2 représentant le bloc de conversion, l'étalonnage statique est le processus par lequel nous allons approximer la fonction reliant les  $y_i$  aux  $x_i$ . À partir d'un modèle mathématique simple nous allons présenter quelques méthodes d'étalonnage statique. Deux classes distinctes sont considérées, les méthodes dites *paramétriques*, et les méthodes dites *non paramétriques*.

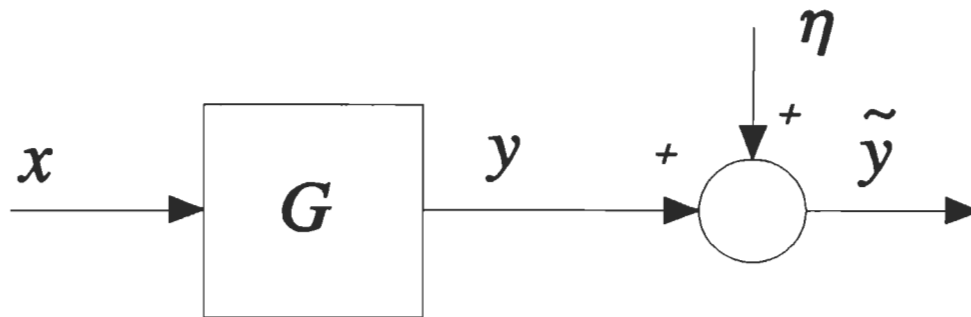


Fig.2-2 Représentation de la partie conversion du système de mesure.

Nous observons le comportement de la sortie  $y$  pour une entrée  $x = [x_1, x_2, \dots, x_n]^T$ , soit à chaque entrée  $x_i$  correspond une sortie  $y_i$  sur une plage de  $n$  observations. Le modèle d'étalonnage est le suivant:

$$y_i = G(x_i) + \eta_i \quad \text{pour } i = 1, \dots, n \quad (2-2)$$

où  $\eta_i$  est une erreur de moyenne zéro et de variance  $\sigma_\eta^2$ .

Un modèle d'étalonnage est dit *paramétrique* si la forme de la fonction  $G$  est connue et qu'un nombre fini de paramètres est à déterminer, par exemple:

$$y = \sum_{i=1}^n p_i x_i \quad (2-3)$$

où les  $p_i$  sont les paramètres à déterminer.

Les modèles paramétriques ne sont pas uniques et les modèles sont donc très flexibles et s'ajustent très bien aux données. Cependant les modèles paramétriques demandent un grand niveau de spécifications pour être estimés, ce niveau est diminué dans le cas de modèles d'étalonnage non paramétriques.

Un modèle d'étalonnage est dit *non paramétrique* si la fonction  $G$  est représentée par un élément d'un espace de fonctions choisi. Ce choix est en général motivé par les propriétés de lissage que la fonction  $G$  est supposée avoir. Les données permettent alors de déterminer l'élément de l'espace de fonctions  $G$ .

## 2.4. Reconstitution de mesurandes

Le mesure est une expérimentation pour l'identification d'un mesurande. Le problème de la reconstitution se pose à l'interprétation du résultat de conversion  $y$  pour accomplir le traitement de  $y$  à  $x$  qui est le mesurande estimé. Le problème de reconstitution peut être classifié en deux types:

- la reconstitution statique et nonlinéaire (SR);
- la reconstitution dynamique et linéaire (DR).

Dans ce projet, nous considérons le problème de reconstitution statique et nonlinéaire. Le problème SR est posé avec le modèle mathématique suivant:

$$y = G(x, v; p) \quad (2-4)$$

où,  $G$  est une fonction du système de mesure. Alors, la reconstitution de mesurandes peut être exprimée par:

$$\hat{x} = \arg_x \{y = G(x, v; p)\} = F(y, v; \hat{p}) \quad (2-5)$$

où,  $F$  est une fonction inverse du système de mesure.

La reconstitution n'est en fait qu'un problème inverse particulier, et les difficultés rencontrées sont liées à la résolution d'une équation intégrale de première espèce, compliquée par la présence de l'erreur de mesure, ceci est représenté à la Fig.2-2. En effet, le signal accessible  $y$  est entaché d'erreurs venant de perturbations inévitables. Ce qui fait que ce problème très simple et fort répandu offre le désagrément d'être mal posé. Ceci ne signifie pas qu'il soit mal formulé, mais simplement que sa résolution est très sensible aux inévitables erreurs dues à la modélisation. Nous obtenons l'équation numérique suivante:

$$\hat{y} = \sum_{i=0}^k p_i x^i + \eta \quad (2-6)$$

où  $\eta$  est une erreur additionnelle causant, même à de faibles valeurs, de grandes erreurs de reconstitution. Ainsi, une information *a priori* apportée sur la solution désirée fournissant une solution acceptable physiquement est appelée régularisation d'un problème mal posé.

La résolution de (2-5) devient un problème de *reconstitution* dans le cas où l'on cherche à reconstituer le signal d'entrée  $x$  et un problème d'*identification* ou d'*étalonnage* dans le cas où nous cherchons à identifier les paramètres du système  $\hat{G}$ .

Les méthodes de reconstitution sont généralement basées sur certaines suppositions concernant le modèle mathématique de la relation entre les signaux et l'information *a priori* accessible et l'erreur qui entache les résultats de conversion  $\hat{y}$ . Il existe une diversité de méthodes pour résoudre le problème de la reconstitution ce qui démontre bien la difficulté présente. En fait, il

n'existe pas de solution universelle. Le fait de ne pas avoir de solution unique pour résoudre (2-7) nous oblige à faire un choix parmi les diverses possibilités de résolution et pour chaque problème de reconstitution, il faut choisir ou développer une méthode adaptée à l'idée que l'on se fait *a priori* du signal à restaurer.

Dans le cadre de notre projet, nous allons nous intéresser à une méthode d'approximation basée sur les fonctions spline. Pour pouvoir évaluer les résultats d'étalonnage et de reconstitution de mesurandes obtenus avec ces méthodes, différentes méthodes d'estimation des paramètres seront utilisées. Nous dirons simplement que l'utilisation des fonctions d'approximation telles que les fonctions spline consiste à établir une approximation d'une fonction inconnue en se basant sur des données de mesure du système à l'étude. Ces données sont sujettes à des erreurs et parfois ne donnent pas une approximation unique. Une fonction dite *spline* est une fonction formée de morceaux de polynômes qui se raccordent ainsi que certaines de leurs dérivées aux points de jonction.

Les méthodes numériques de reconstitution de signaux sont dépendantes du modèle mathématique de la relation entre les signaux et l'information liée au système de mesure, tels que la fonction de transfert et des erreurs qui le perturbent. La résolution est souvent sensible aux perturbations qui agissent sur le système de mesure.

Le problème de reconstitution, posé d'une façon abstraite, est bien conforme aux nombreuses situations pratiques dans les différents domaines de la technique de mesure et de commande. Des méthodes numériques de reconstitution sont profitablement utilisées pour:

- l'interprétation des données sismiques,
- l'amélioration de la résolution de l'analyse spectrométrique et chromatographique,
- la correction dynamique de capteurs,
- la mesure de la thermocinétique de réactions chimiques,
- l'accélération de mesurage quasi-statique,
- l'égalisation de canaux de télécommunication,
- le diagnostic médical basé sur les techniques de tomographie, etc.

Elles sont aussi utilisables dans le domaine de la robotique comme des moyens d'étalonnage cinétique et d'estimation de paramètres dynamiques de manipulateurs. Les deux principales raisons de l'intérêt actuel pour l'étude des méthodes de reconstitution de signaux sont:

- L'amélioration de l'exactitude des résultats fournis par un appareil sans pour autant atteindre les limites de résolution et ainsi travailler avec un appareil à moindre coût.
- La technologie actuelle nous permet la réalisation de processeurs en traitement du signal pour ainsi atteindre des vitesses de reconstitution raisonnables.

Compte tenu des problèmes inhérents aux méthodes de reconstitution analytiques, nous envisageons une méthode non analytique, indépendante du modèle mathématique du système de

mesure, basée sur les fonctions spline et leurs performances en reconnaissance de formes et classification. L'idée de base est la suivante: une fonction spline peut être considérée comme une *fonction de transfert* accessible par ses entrées et ses sorties. Les signaux d'entrée sont introduits dans la *fonction*, ils sont traités, et finalement un résultat nous est fourni par la fonction au niveau de sa sortie. Avec les fonctions spline, nous disposons donc d'un système de traitement de signaux, la méthode développée dans ce projet repose sur ce principe de base.

Nous allons présenter les systèmes de mesure qui seront utilisés dans la suite du projet pour valider l'applicabilité des fonctions spline à l'étalonnage d'un système de mesure et à la reconstitution de mesurandes. Les systèmes de mesure peuvent être décomposés en trois blocs: les deux premiers servent à convertir le mesurande en un signal numérique, c'est la partie de la conversion du système, traitable par le troisième bloc qui sert à reconstituer le mesurande d'entrée. Le premier bloc est composé d'un capteur sensible au mesurande qu'il subit et dont la caractéristique d'entrée/sortie est non linéaire et monotone. Il reste toutefois un autre problème important qui est l'influence sur la caractéristique d'entrée/sortie du capteur, il faudra donc tenir compte de l'influence pour étalonner le système de mesure et reconstituer les mesurandes.

## CHAPITRE 3.

# MÉTHODES NUMÉRIQUES D'APPROXIMATION PAR LES FONCTIONS SPLINE À UNE VARIABLE

La documentation traitant ce sujet nous donne la description suivante [24-§9.5], [25] et [26]:

*Dans les premiers temps du dessin, par exemple le dessin de bateaux au dix-neuvième siècle, le problème était celui de tracer (à la main) une courbe lisse à travers un ensemble donné de points. Une manière d'obtenir une solution était la suivante: placez des poids de métal (appelés "canards") aux points donnés, puis passez un faisceau de bois mince et élastique (appelé une "spline") entre les canards. La courbe résultante est toujours très lisse et généralement très esthétique. Alors, le "spline" physique est en fait une pièce ou bande de bois ou de métal ou d'autres matériaux longue, étroite, et relativement mince utilisée pour dessiner les courbes.*

*Le même principe est appliqué aujourd'hui, lorsqu'il n'y a pas de programme de dessin approprié disponible, ou pour vérification manuelle du résultat informatique. Le faisceau de plastique ou de bois maintient une position qui minimise son énergie élastique de tension. Le modèle mathématique du faisceau est une courbe. Une fonction, qui a spécifié les valeurs à un nombre fini de points et constitué par les segments de la fonction polynomiale jointe lissement sur les points, pourrait être utilisée pour l'approximation et l'interpolation des fonctions. Donc, une courbe continue constituée pour passer au travers d'un groupe de points et avoir la première dérivé continue et la deuxième dérivé continue.*

Le spline manifeste ses propriétés importantes et ses applications sont introduites dans l'analyse numérique. Le spline mathématique est continu, il a la première et la seconde dérivés continues. Les fonctions spline ont été longtemps utilisées dans les domaines tels que le graphisme et le dessin assisté par ordinateur. Nous définissons la fonction spline comme étant une courbe continue construite de façon à passer par une série de points tout en ayant sa première et

deuxième dérivés continues. La Fig.3-1 montre un exemple possible.

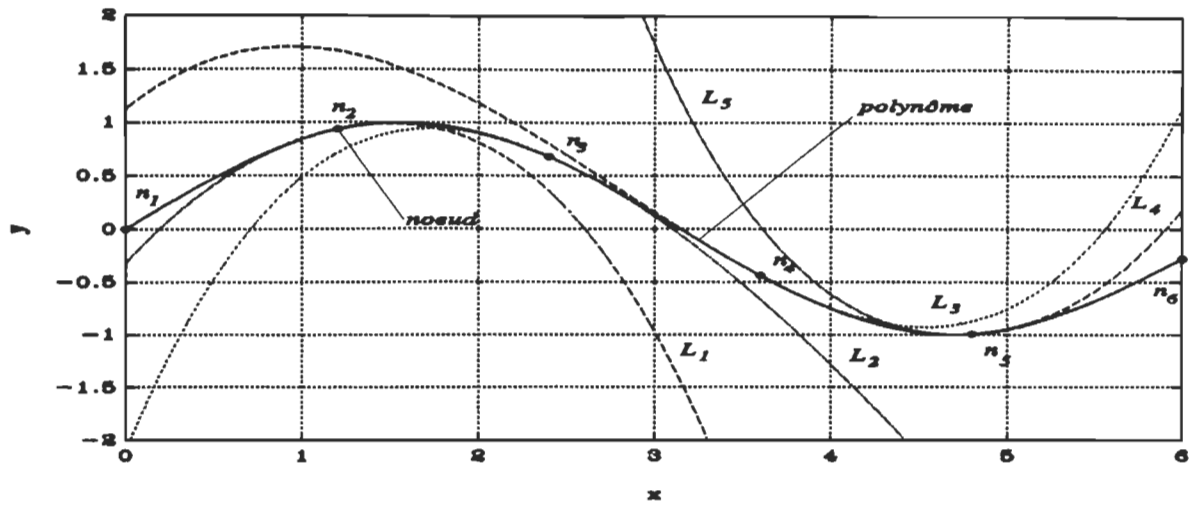


Fig.3-1. Les polynômes et les noeuds.  $n_i$  est le noeud.  $L_i$  est le polynôme spline.

Il existe plusieurs fonctions spline différentes dépendant des méthodologies d'estimation des paramètres dans les fonctions spline.

### 3.1. Fonctions spline d'interpolation

La méthode d'interpolation est efficace et flexible. Elle est utilisée dans les domaines des analyses de données et de traitement de signaux. Les données discrètes appelées les noeuds et le polynôme entre deux noeuds  $i$  et  $i+1$  constituent la base d'interpolation. Les données discrètes sont

$$\{ x_i, y_i \mid i = 1, \dots, N \} \quad (3-1)$$

Le polynôme entre deux noeuds  $i$  et  $i+1$  est décrit par

$$y = \hat{G}(x; p_i) = p_{0,i} + p_{1,i}x + p_{2,i}x^2 + \dots + p_{m,i}x^m \quad \text{pour } i = 1, \dots, N-1 \quad (3-2)$$

pour simuler la caractéristique du système de mesure composée par les données discrètes. Chaque polynôme entre deux noeuds  $i$  et  $i+1$  est différent. La combinaison des polynômes et des noeuds est montrée sur la Fig.3-1.  $n_i$  ( $i = 1, \dots, N$ ) est le noeud.  $L_i$  ( $i = 1, \dots, N-1$ ) est le polynôme spline qui correspond à la fonction de polynôme  $\hat{G}(x; p_i)$  comme (3-2). Dans ce cas, il y a six noeuds et cinq polynômes spline qui forment une caractéristique du système.

Ainsi, les polynômes spline décrivent la relation entre  $x$  et  $y$ . Ils sont aussi les fonctions de transfert entre le signal mesuré  $x$  et le résultat brut de mesure  $y$ . La conversion des signaux et la reconstitution des signaux sont basées sur les polynômes spline.



Nous avons choisi le polynôme comme la base de la construction de spline pour la raison suivante: n'importe quelle fonction à  $x=x_0$  peut être exprimée approximativement par le polynôme selon les séries de Taylor:

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + \dots + \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n + \dots \quad (3-3)$$

L'exactitude d'approximation dépend du degré  $n$ . Les termes rémanents d'ordre supérieur à  $n$  sont considérés nulles. Bien sûr, l'exactitude d'approximation dépend aussi de la condition d'extrémité et du choix des paramètres. De plus, la valeur du polynôme peut être calculée très facilement. L'inverse du polynôme n'est pas difficile.

Maintenant, le point focal du problème est le choix du polynôme: ses paramètres et son degré. La définition d'interpolation entend que

- la fonction d'interpolation passe à travers chaque noeud;
- la fonction d'interpolation soit lisse partout.

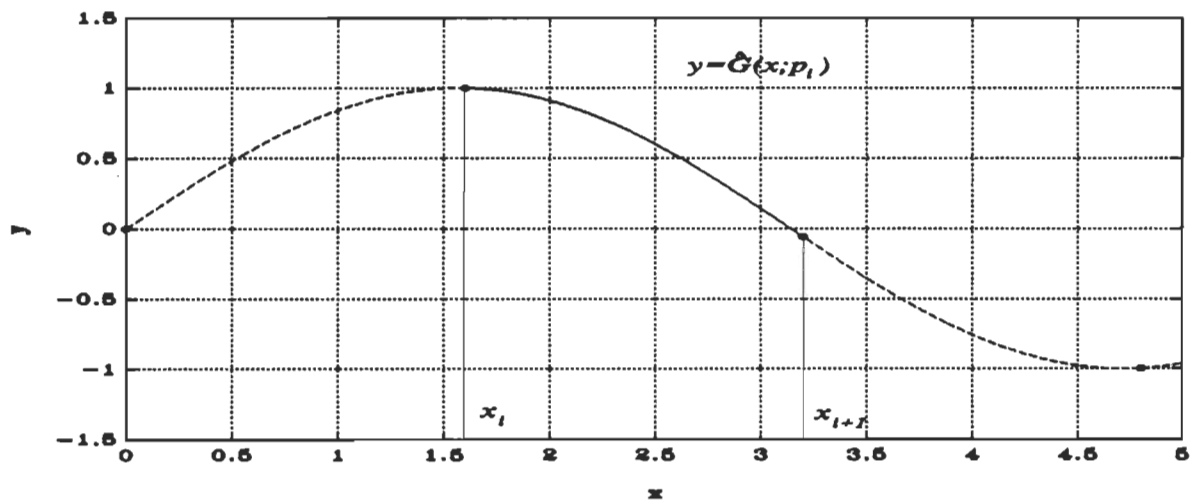


Fig.3-2. Le sous-intervalle et les noeuds  $x \in (x_i, x_{i+1})$ .

Nous posons la fonction de la caractéristique du système à être interpolée:

$$y_i = G(x_i) \quad \text{pour } i = 1, \dots, N \quad (3-4)$$

et la fonction d'interpolation à une variable:

$$y = \hat{G}(x; p_i) \quad \text{pour } i = 1, \dots, N-1 \quad (3-5)$$

sur le sous-intervalle  $(x_i, x_{i+1})$ . Le sous-intervalle  $x \in (x_i, x_{i+1})$  est montré sur la Fig.3-2. Sur le sous-intervalle  $(x_i, x_{i+1})$ , il faut assurer la continuité de la fonction d'interpolation, de sa première (la fonction spline parabolique d'interpolation) et de sa seconde dérivé (la fonction spline parabolique d'interpolation et la fonction spline cubique d'interpolation) pour approximer au maximum la fonction réelle. Les conditions sont montrées sur la Fig.3-3.

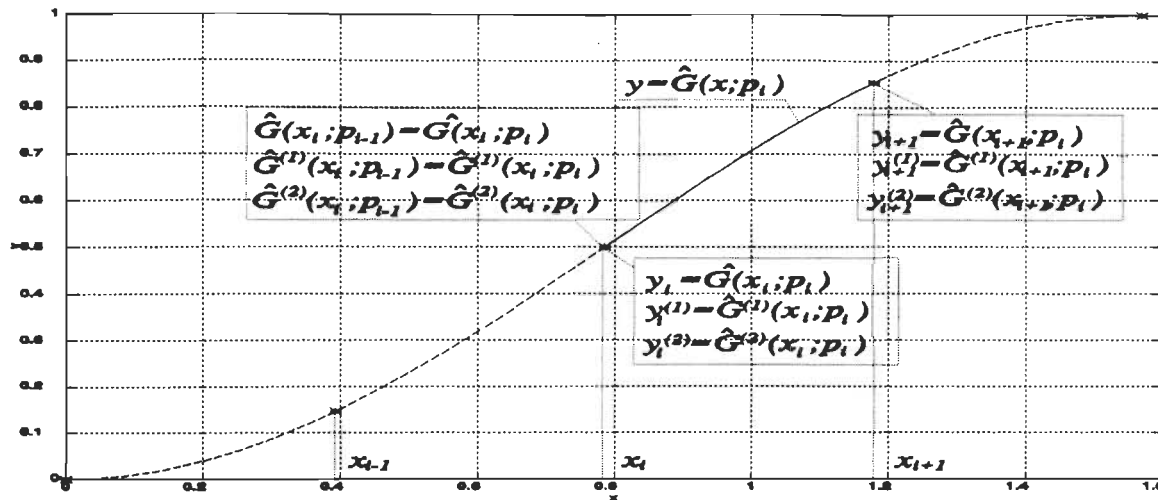


Fig.3-3. Les conditions d'interpolation sur le subintervalle et les noeuds  $x \in (x_i, x_{i+1})$ .

### 3.1.1. Fonction spline parabolique d'interpolation

La fonction spline parabolique d'interpolation *IntSplPI* est exprimée par un polynôme d'ordre 2. Sa formule sur le sous-intervalle  $(x_i, x_{i+1})$  est

$$y = \hat{G}(x; p_i) = p_{0,i} + p_{1,i}(x - x_i) + p_{2,i}(x - x_i)^2 \quad \text{pour } i = 1, \dots, N-1 \quad (3-6)$$

et sa première dérivée est

$$y^{(1)} = \hat{G}^{(1)}(x; p_i) = p_{1,i} + 2p_{2,i}(x - x_i) \quad \text{pour } i = 1, \dots, N-1 \quad (3-7)$$

et sa deuxième dérivée est

$$y^{(2)} = \hat{G}^{(2)}(x; p_i) = 2p_{2,i} \quad \text{pour } i = 1, \dots, N-1 \quad (3-8)$$

où,  $x_i$  est le noeud.  $p_{0,i}$ ,  $p_{1,i}$ ,  $p_{2,i}$  sont les coefficients de la fonction d'interpolation qui doivent être déterminés à chaque sous intervalle selon certaines conditions.

Il faut assurer la continuité de la fonction d'interpolation et de sa première dérivée pour approximer au maximum la fonction réelle.

(1) la condition d'interpolation:

$$y_i = \hat{G}(x; p_i) \quad \text{quand } x = x_i \quad (3-9)$$

et

$$y_{i+1} = \hat{G}(x; p_i) \quad \text{quand } x = x_{i+1} \quad (3-10)$$

Cette condition aboutit à

$$y_i = \hat{G}(x_i; p_i) = p_{0,i} \quad (3-11)$$

et

$$y_{i+1} = \hat{G}(x_{i+1}; \mathbf{p}_i) = p_{0,i} + p_{1,i}(x_{i+1} - x_i) + p_{2,i}(x_{i+1} - x_i)^2 \quad (3-12)$$

(2) la condition de continuité à  $x = x_i$ :

$$\hat{G}(x_i; \mathbf{p}_{i-1}) = \hat{G}(x_i; \mathbf{p}_i) \quad (3-13)$$

i.e.,

$$y_i = \hat{G}(x_i; \mathbf{p}_i) = p_{0,i} \quad (3-14)$$

et

$$y_i = \hat{G}(x_i; \mathbf{p}_{i-1}) = p_{0,i-1} + p_{1,i-1}(x_i - x_{i-1}) + p_{2,i-1}(x_i - x_{i-1})^2 \quad (3-15)$$

Cette condition aboutit à

$$p_{0,i} = p_{0,i-1} + p_{1,i-1}(x_i - x_{i-1}) + p_{2,i-1}(x_i - x_{i-1})^2 \quad (3-16)$$

(3) la condition de continuité de la première dérivée à  $x = x_i$ :

$$\hat{G}^{(1)}(x_i; \mathbf{p}_{i-1}) = \hat{G}^{(1)}(x_i; \mathbf{p}_i) \quad (3-17)$$

i.e.,

$$y_i^{(1)} = \hat{G}^{(1)}(x_i; \mathbf{p}_i) = p_{1,i} \quad (3-18)$$

et

$$y_i^{(1)} = \hat{G}^{(1)}(x_i; \mathbf{p}_{i-1}) = p_{1,i-1} + 2p_{2,i-1}(x_i - x_{i-1}) \quad (3-19)$$

Cette condition aboutit à

$$p_{1,i} = p_{1,i-1} + 2p_{2,i-1}(x_i - x_{i-1}) \quad (3-20)$$

Alors, à partir de (3-12), (3-14) et (3-20), pour le sous intervalle  $i = 2, \dots, N-1$ , nous avons la forme de matrice:

$$\begin{bmatrix} p_{1,i-1} + 2p_{2,i-1}(x_i - x_{i-1}) \\ y_i \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & (x_{i+1} - x_i) & (x_{i+1} - x_i)^2 \end{bmatrix} \begin{bmatrix} p_{0,i} \\ p_{1,i} \\ p_{2,i} \end{bmatrix} \quad (3-21)$$

Pour  $i = 1$ , nous avons besoin d'une condition aux limites supplémentaire. La condition limite est choisie d'une façon variée.

(1) la condition aux limites avec la première dérivée spécifiée à  $i = 1$ :

$$\frac{dy}{dx} \Big|_{x=x_1} = \hat{G}^{(1)}(x_1; \mathbf{p}_1) = 0 \quad (3-22)$$

Cette condition aboutit à

$$p_{1,1} = 0 \quad (3-23)$$

$$y_1 = p_{0,1} \quad (3-24)$$

$$y_2 = p_{0,1} + p_{1,1}(x_2 - x_1) + p_{2,1}(x_2 - x_1)^2 \quad (3-25)$$

et la forme de matrice:

$$\begin{bmatrix} 0 \\ y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & (x_2 - x_1) & (x_2 - x_1)^2 \end{bmatrix} \begin{bmatrix} p_{0,1} \\ p_{1,1} \\ p_{2,1} \end{bmatrix} \quad (3-26)$$

Le résultat obtenu pour l'approximation d'un signal sinusoidal, sous la condition aux limites, est montré sur la Fig.3-4.

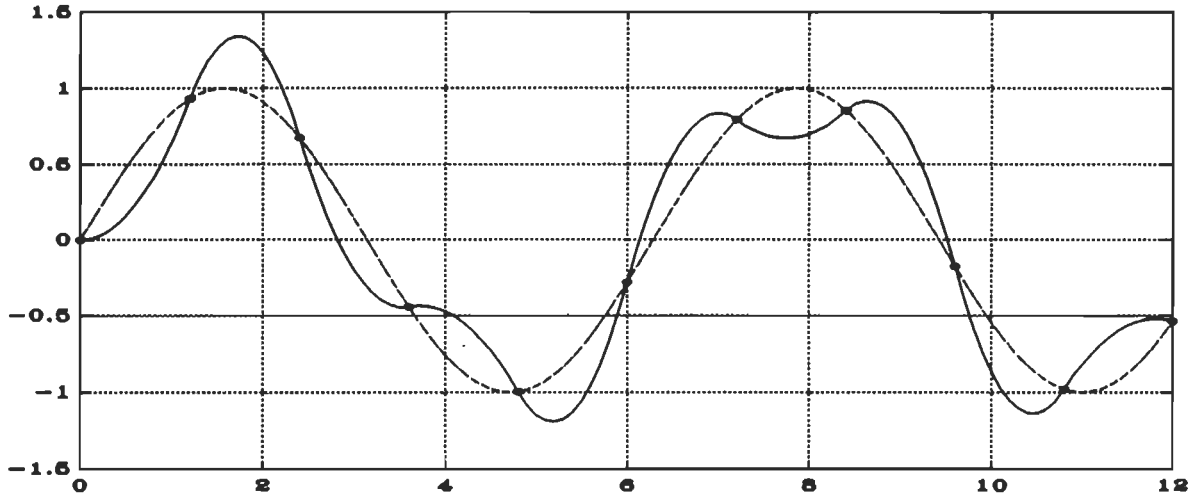


Fig.3-4. La fonction spline parabolique d'interpolation avec la condition d'extrémité (1):  $dy/dx=0$  à  $x=x_1$ .

(2) la condition aux limites "naturelle" (la deuxième dérivée spécifiée à  $i = 1$ ):

$$\frac{d^2y}{dx^2} \Big|_{x=x_1} = \hat{G}^{(2)}(x_1; p_1) = 0 \quad (3-27)$$

Cette condition aboutit à

$$2p_{2,1} = 0 \quad (3-28)$$

$$y_1 = p_{0,1} \quad (3-29)$$

$$y_2 = p_{0,1} + p_{1,1}(x_2 - x_1) + p_{2,1}(x_2 - x_1)^2 \quad (3-30)$$

et la forme de matrice:

$$\begin{bmatrix} 0 \\ y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 2 \\ 1 & 0 & 0 \\ 1 & (x_2 - x_1) & (x_2 - x_1)^2 \end{bmatrix} \begin{bmatrix} p_{0,1} \\ p_{1,1} \\ p_{2,1} \end{bmatrix} \quad (3-31)$$

Le résultat obtenu pour l'approximation d'un signal sinusoidal, sous la condition aux limites, est montré sur la Fig.3-5.

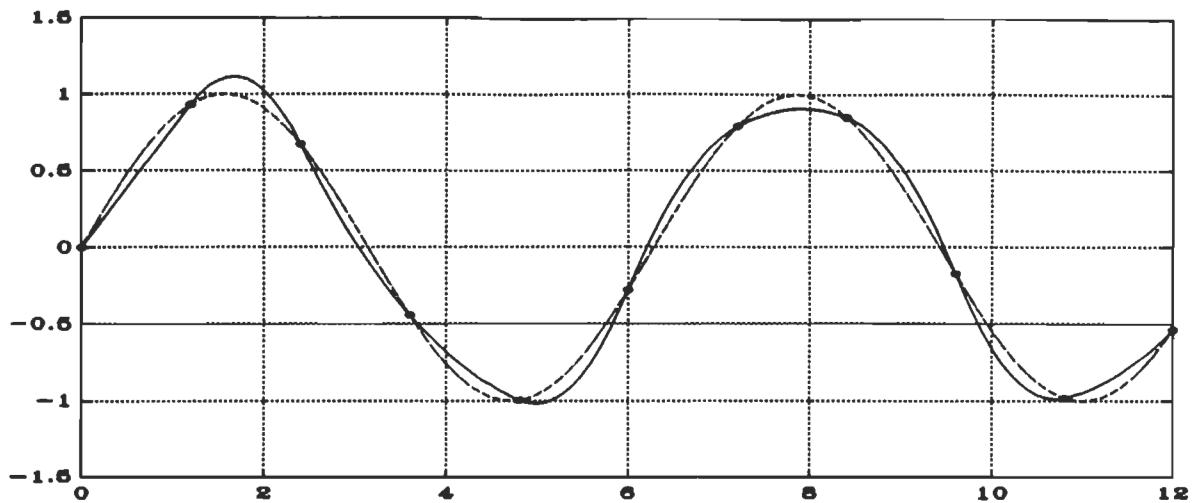


Fig.3-5. La fonction spline parabolique d'interpolation avec la condition d'extrémité "naturelle" (2):  $d^2y/dx^2=0$  à  $x=x_1$ .

(3) la condition aux limites avec le calcul de la première dérivée à  $i=1$ :

Nous utilisons les quatre noeuds pour déterminer la valeur  $\frac{dy}{dx} \Big|_{i=1}$  selon la formule d'interpolation de Lagrange:

$$y_1^{(1)} = y_1 \frac{\Delta x_{13} \Delta x_{14} + \Delta x_{12} \Delta x_{14} + \Delta x_{12} \Delta x_{13}}{\Delta x_{12} \Delta x_{13} \Delta x_{14}} + y_2 \frac{\Delta x_{13} \Delta x_{14}}{\Delta x_{21} \Delta x_{23} \Delta x_{24}} + y_3 \frac{\Delta x_{12} \Delta x_{14}}{\Delta x_{31} \Delta x_{32} \Delta x_{34}} + y_4 \frac{\Delta x_{12} \Delta x_{13}}{\Delta x_{41} \Delta x_{42} \Delta x_{43}} \quad (3-32)$$

où  $\Delta x_{ij} = x_i - x_j$ , et nous aboutissons à la matrice:

$$\begin{bmatrix} y_1^{(1)} \\ y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & (x_2 - x_1) & (x_2 - x_1)^2 \end{bmatrix} \begin{bmatrix} p_{0,1} \\ p_{1,1} \\ p_{2,1} \end{bmatrix} \quad (3-33)$$

Le résultat obtenu pour l'approximation d'un signal sinusoidal, sous la condition aux limites, est montré sur la Fig.3-6. Cette condition aux limites est utilisée dans ce projet.

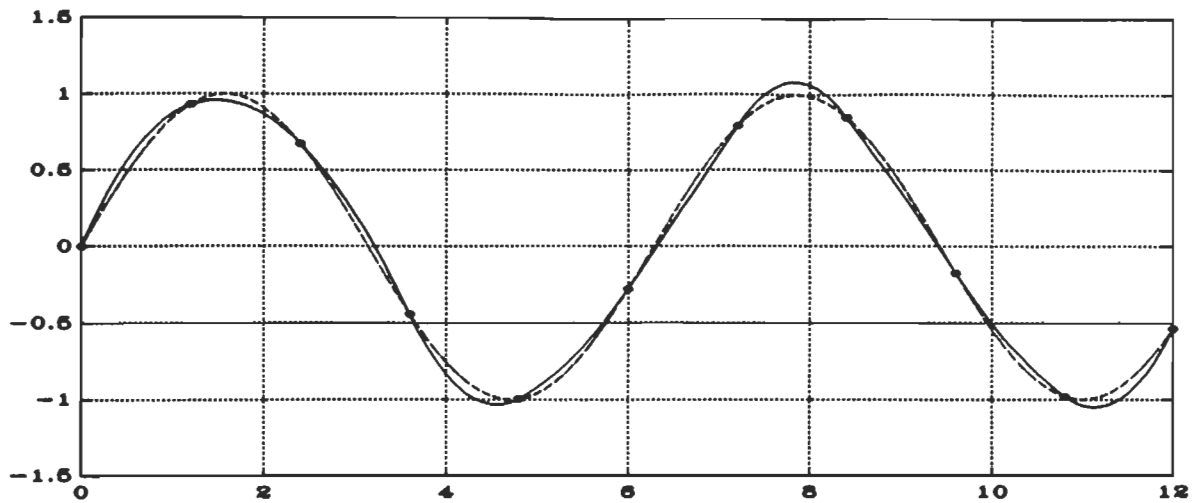


Fig.3-6. La fonction spline parabolique d'interpolation avec la condition d'extrémité (3): le calcul de la première dérivée à  $x=x_i$ .

### 3.1.2. Fonction spline cubique d'interpolation

La fonction spline cubique d'interpolation  $IntSplCI$  sur le sous-intervalle  $(x_i, x_{i+1})$  est

$$y = \hat{G}(x; \mathbf{p}_i) = p_{0,i} + p_{1,i}(x - x_i) + p_{2,i}(x - x_i)^2 + p_{3,i}(x - x_i)^3 \quad \text{pour } i = 1, \dots, N-1 \quad (3-34)$$

et sa première dérivée est

$$y^{(1)} = \hat{G}^{(1)}(x; \mathbf{p}_i) = p_{1,i} + 2p_{2,i}(x - x_i) + 3p_{3,i}(x - x_i)^2 \quad \text{pour } i = 1, \dots, N-1 \quad (3-35)$$

et sa deuxième dérivée est

$$y^{(2)} = \hat{G}^{(2)}(x; \mathbf{p}_i) = 2p_{2,i} + 6p_{3,i}(x - x_i) \quad \text{pour } i = 1, \dots, N-1 \quad (3-36)$$

$p_{0,i}$ ,  $p_{1,i}$ ,  $p_{2,i}$ ,  $p_{3,i}$  sont les coefficients de la fonction d'interpolation qui doivent être déterminés pour chaque sous intervalle. La fonction est exprimée par le polynôme d'ordre 3.

Les coefficients  $p_{0,i}$ ,  $p_{1,i}$ ,  $p_{2,i}$ ,  $p_{3,i}$  peuvent être trouvés avec les conditions d'interpolation et les conditions de continuité pour la première et la deuxième dérivées sur tous les noeuds. Encore une fois, nous avons besoin de deux conditions supplémentaires qui peuvent être choisies par une variété de moyens.

Il faut assurer la continuité de la fonction d'interpolation, de sa première et de sa seconde dérivée pour approximer au maximum la fonction réelle.

(1) la condition d'interpolation:

$$y_i = \hat{G}(x; \mathbf{p}_i) \quad \text{quand } x = x_i \quad (3-37)$$

et

$$y_{i+1} = \hat{G}(x; \mathbf{p}_i) \quad \text{quand } x = x_{i+1} \quad (3-38)$$

Cette condition aboutit à

$$y_i = \hat{G}(x_i; \mathbf{p}_i) = p_{0,i} \quad (3-39)$$

et

$$y_{i+1} = \hat{G}(x_{i+1}; \mathbf{p}_i) = p_{0,i} + p_{1,i}(x_{i+1} - x_i) + p_{2,i}(x_{i+1} - x_i)^2 + p_{3,i}(x_{i+1} - x_i)^3 \quad (3-40)$$

(2) la condition d'interpolation de la première dérivée à  $x = x_{i+1}$  :

$$y_{i+1}^{(1)} = \hat{G}^{(1)}(x_{i+1}; \mathbf{p}_i) = p_{1,i} + 2p_{2,i}(x_{i+1} - x_i) + 3p_{3,i}(x_{i+1} - x_i)^2 \quad (3-41)$$

(3) la condition de continuité à  $x = x_i$  :

$$\hat{G}(x_i; \mathbf{p}_{i-1}) = \hat{G}(x_i; \mathbf{p}_i) \quad (3-42)$$

i.e.,

$$y_i = \hat{G}(x_i; \mathbf{p}_i) = p_{0,i} \quad (3-43)$$

et

$$y_i = \hat{G}(x_i; \mathbf{p}_{i-1}) = p_{0,i-1} + p_{1,i-1}(x_i - x_{i-1}) + p_{2,i-1}(x_i - x_{i-1})^2 + p_{3,i-1}(x_i - x_{i-1})^3 \quad (3-44)$$

Cette condition aboutit à

$$p_{0,i} = p_{0,i-1} + p_{1,i-1}(x_i - x_{i-1}) + p_{2,i-1}(x_i - x_{i-1})^2 + p_{3,i-1}(x_i - x_{i-1})^3 \quad (3-45)$$

(4) la condition de continuité de la première dérivée à  $x = x_i$  :

$$\hat{G}^{(1)}(x_i; \mathbf{p}_{i-1}) = \hat{G}^{(1)}(x_i; \mathbf{p}_i) \quad (3-46)$$

i.e.,

$$y_i^{(1)} = \hat{G}^{(1)}(x_i; \mathbf{p}_i) = p_{1,i} \quad (3-47)$$

et

$$y_i^{(1)} = \hat{G}^{(1)}(x_i; \mathbf{p}_{i-1}) = p_{1,i-1} + 2p_{2,i-1}(x_i - x_{i-1}) + 3p_{3,i-1}(x_i - x_{i-1})^2 \quad (3-48)$$

Cette condition aboutit à

$$p_{1,i} = p_{1,i-1} + 2p_{2,i-1}(x_i - x_{i-1}) + 3p_{3,i-1}(x_i - x_{i-1})^2 \quad (3-49)$$

(5) la condition de continuité de la deuxième dérivée à  $x = x_i$  :

$$\hat{G}^{(2)}(x_i; \mathbf{p}_{i-1}) = \hat{G}^{(2)}(x_i; \mathbf{p}_i) \quad (3-50)$$

i.e.,

$$y_i^{(2)} = \hat{G}^{(2)}(x_i; \mathbf{p}_i) = 2p_{2,i} \quad (3-51)$$

et

$$y_i^{(2)} = \hat{G}^{(2)}(x_i; \mathbf{p}_{i-1}) = 2p_{2,i-1} + 6p_{3,i-1}(x_i - x_{i-1}) \quad (3-52)$$

Cette condition aboutit à

$$2p_{2,i} = 2p_{2,i-1} + 6p_{3,i-1}(x_i - x_{i-1}) \quad (3-53)$$

Alors, pour le sous intervalle  $i=2, \dots, N-1$ , nous avons les équations des paramètres par rapport à la première dérivée:

À partir de (3-39), nous avons

$$p_{0,i} = y_i \tag{3-54}$$

À partir de (3-47), nous avons

$$p_{1,i} = y_i^{(1)} \tag{3-55}$$

À partir de (3-40) et (3-41), nous avons

$$p_{2,i} = \frac{3(y_{i+1} - y_i) - 2y_i^{(1)}(x_{i+1} - x_i) - y_{i+1}^{(1)}(x_{i+1} - x_i)}{(x_{i+1} - x_i)^2} \tag{3-56}$$

$$p_{3,i} = \frac{-2(y_{i+1} - y_i) + y_i^{(1)}(x_{i+1} - x_i) + y_{i+1}^{(1)}(x_{i+1} - x_i)}{(x_{i+1} - x_i)^3} \tag{3-57}$$

et en utilisant (3-53) avec ses dernières relations, on aura

$$3(y_{i+1} - y_{i-1}) = y_{i-1}^{(1)}(x_{i+1} - x_i) + 4y_i^{(1)}(x_{i+1} - x_i) + y_{i+1}^{(1)}(x_{i+1} - x_i) \tag{3-58}$$

Pour  $i=1$  et  $i=N$ , nous avons besoin d'une condition aux limites supplémentaire pour déterminer les paramètres. La condition aux limites est choisie d'une façon variée.

(1) la condition aux limites "naturelle" (ce choix requiert que la deuxième dérivée des points d'extrémité soit nulle):

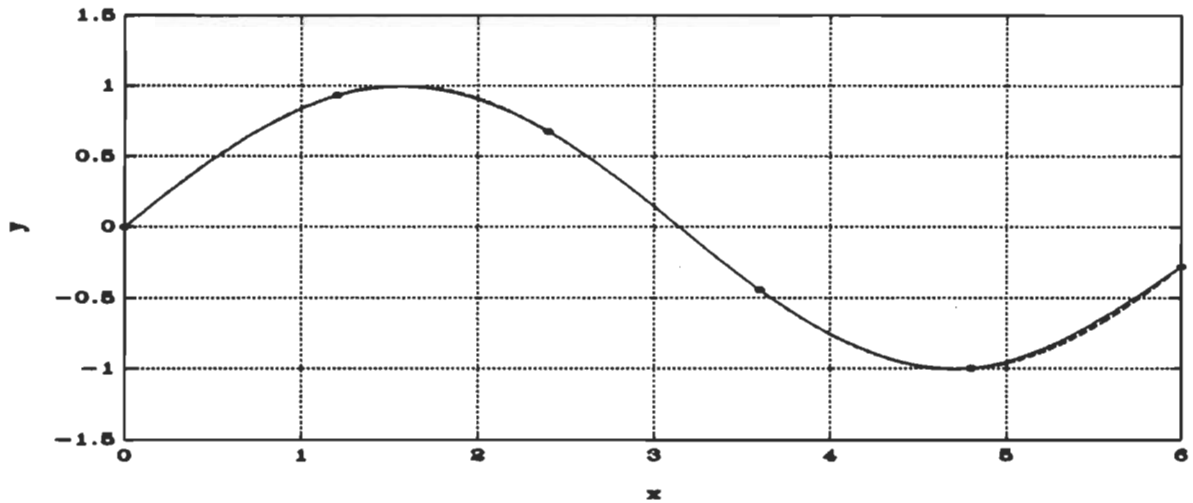


Fig.3-7. La fonction spline cubique d'interpolation avec la condition d'extrémité "naturelle" (1):  $d^2y/dx^2=0$  à  $x=x_1$  et  $x=x_N$ .

$$\frac{d^2y}{dx^2} = \hat{G}^{(2)}(x; p_i) = 2p_{2,i} + 6p_{3,i}(x - x_i) = 0 \quad \text{pour } x = x_1, x = x_N \tag{3-59}$$

i.e.,



$$y_1^{(2)} = 2p_{2,1} = 0 \quad (3-60)$$

En remplaçant cette équation dans (3-56) avec  $i=1$ , on aura

$$3(y_2 - y_1) = 2y_1^{(1)}(x_2 - x_1) + y_2^{(1)}(x_2 - x_1) \quad (3-61)$$

et

$$y_N^{(2)} = 2p_{2,N-1} = 0 \quad (3-62)$$

En remplaçant cette équation dans (3-57) avec  $i=N$ , on aura

$$3(y_N - y_{N-1}) = 2y_N^{(1)}(x_N - x_{N-1}) + y_{N-1}^{(1)}(x_N - x_{N-1}) \quad (3-63)$$

C'est une équation linéaire par rapport au vecteur de valeurs de la première dérivée:

$$\begin{bmatrix} 2(x_2 - x_1) & (x_2 - x_1) & 0 & \dots & \dots & \dots \\ (x_3 - x_2) & 4(x_3 - x_2) & (x_3 - x_2) & 0 & \dots & \dots \\ 0 & (x_4 - x_3) & 4(x_4 - x_3) & (x_4 - x_3) & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & 0 & (x_N - x_{N-1}) & 4(x_N - x_{N-1}) & (x_N - x_{N-1}) \\ \dots & \dots & \dots & 0 & (x_N - x_{N-1}) & 2(x_N - x_{N-1}) \end{bmatrix} \begin{bmatrix} y_1^{(1)} \\ y_2^{(1)} \\ y_3^{(1)} \\ \dots \\ y_{N-1}^{(1)} \\ y_N^{(1)} \end{bmatrix} = \begin{bmatrix} 3(y_2 - y_1) \\ 3(y_3 - y_1) \\ 3(y_4 - y_2) \\ \dots \\ 3(y_N - y_{N-2}) \\ 3(y_N - y_{N-1}) \end{bmatrix} \quad (3-64)$$

Les résultats de  $y_i^{(1)}$  sont utilisés dans les équations des paramètres de  $p_i$  dans (3-54) ~ (3-57). Le résultat obtenu sous la condition aux limites est montré sur la Fig.3-7. La condition aux limites "naturelle" est utilisée dans ce projet.

(2) la condition aux limites avec le calcul de la première dérivé à  $i=1$  et  $i=N$  (les quatre noeuds sont utilisés pour déterminer la valeur  $\frac{dy}{dx}$  selon la formule d'interpolation de Lagrange):

$$y_1^{(1)} = y_1 \frac{\Delta x_{13} \Delta x_{14} + \Delta x_{12} \Delta x_{14} + \Delta x_{12} \Delta x_{13}}{\Delta x_{12} \Delta x_{13} \Delta x_{14}} + y_2 \frac{\Delta x_{13} \Delta x_{14}}{\Delta x_{21} \Delta x_{23} \Delta x_{24}} + y_3 \frac{\Delta x_{12} \Delta x_{14}}{\Delta x_{31} \Delta x_{32} \Delta x_{34}} + y_4 \frac{\Delta x_{12} \Delta x_{13}}{\Delta x_{41} \Delta x_{42} \Delta x_{43}} \quad (3-65)$$

où  $\Delta x_{ij} = x_i - x_j$ . Le même calcul est fait pour  $y_N^{(1)}$ . Les résultats de  $y_i^{(1)}$  sont utilisés dans les équations des paramètres de  $p_i$ . Le résultat obtenu sous la condition aux limites est montré sur la Fig.3-8.

### 3.2. Fonctions spline d'approximation de moindres carrés

Nous définissons que  $\{x_i, y_i\}$  est le point d'étalonnage et que  $\{\hat{x}_i, \hat{y}_i\}$  est le point

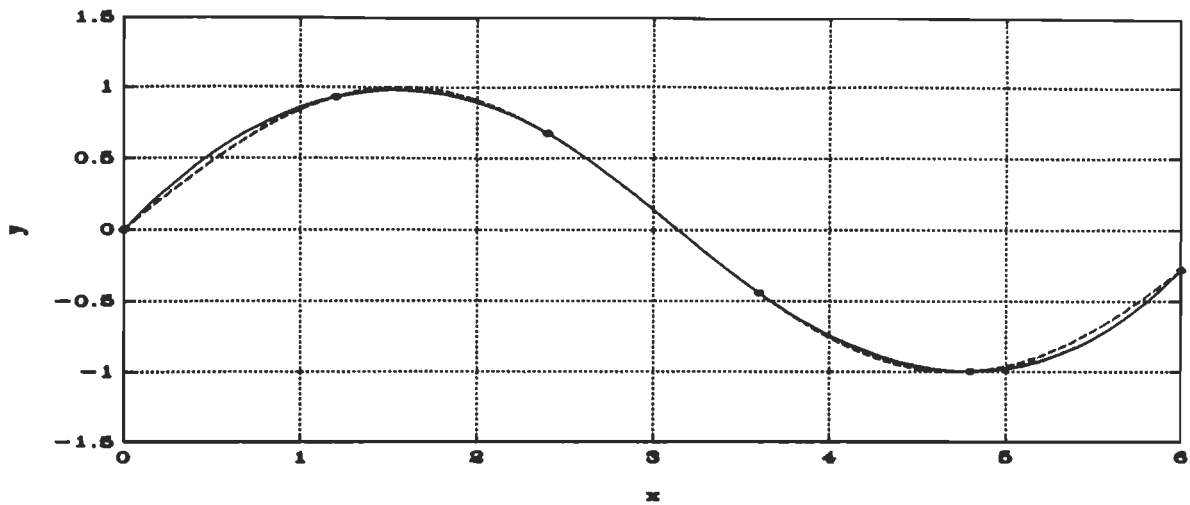


Fig.3-8. La fonction spline cubique d'interpolation avec la condition d'extrémité (2): le calcul de la première dérivé à  $x=x_1$  et  $x=x_N$ .

d'approximation. Ces définitions sont montrées sur la Fig.3-9. En général,  $y_i = \hat{y}_i$  est dans la fonction spline d'interpolation,  $y_i \neq \hat{y}_i$  est dans la fonction spline de moindres carrés ou de lissage.

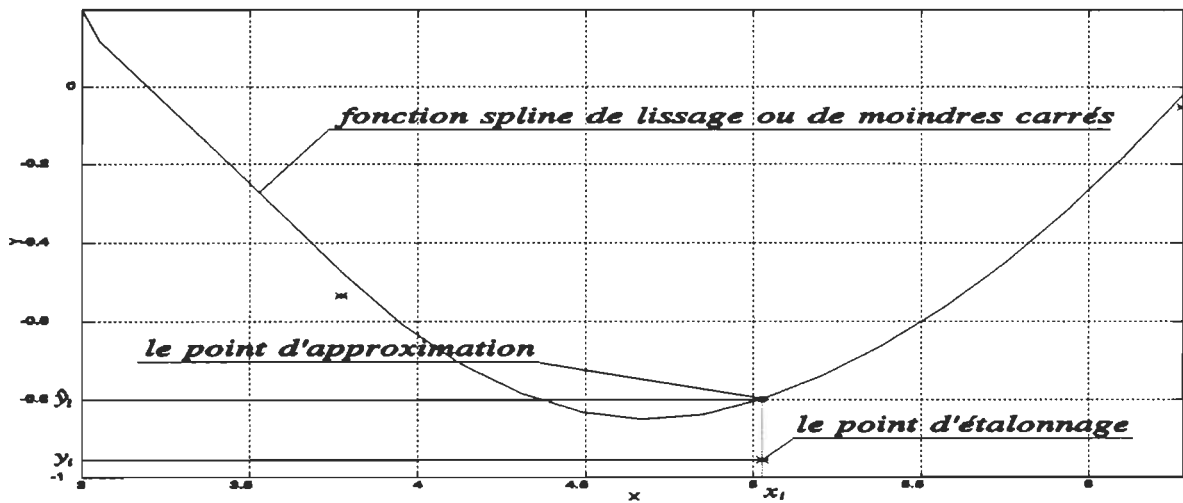


Fig.3-9. La courbe de la fonction spline de lissage ou de moindres carrés, les données d'approximation ( $\cdot$ ) et les données d'étalonnage (\*).

### 3.2.1. Fonction spline parabolique d'approximation de moindres carrés

La routine *ParLsmSplPl* calcule les paramètres:

$$\mathbf{p}_i = [p_{0,i}, p_{1,i}, p_{2,i}] \quad (3-66)$$

dans la fonction spline parabolique de moindres carrés

$$y = \hat{G}(x; \mathbf{p}_i) = p_{0,i} + p_{1,i}(x - x_i) + p_{2,i}(x - x_i)^2 \quad (3-67)$$

La première dérivée est donnée par la relation

$$y^{(1)} = \hat{G}^{(1)}(x; \mathbf{p}_i) = p_{1,i} + 2p_{2,i}(x - x_i) \quad (3-68)$$

et la deuxième dérivée par la relation

$$y^{(2)} = \hat{G}^{(2)}(x; \mathbf{p}_i) = 2p_{2,i} \quad (3-69)$$

pour  $x \in (x_i, x_{i+1})$ ,  $i = 1, \dots, N-1$ . La fonction spline parabolique de moindres carrés exige seulement la condition de continuité de la fonction et celle de la première dérivée.

(1) la condition de continuité à  $x = x_i$ :

$$\hat{G}(x_i; \mathbf{p}_{i-1}) = \hat{G}(x_i; \mathbf{p}_i) \quad (3-70)$$

i.e.,

$$\hat{y}_i = \hat{G}(x_i; \mathbf{p}_i) = p_{0,i} \quad (3-71)$$

et

$$\hat{y}_i = \hat{G}(x_i; \mathbf{p}_{i-1}) = p_{0,i-1} + p_{1,i-1}(x_i - x_{i-1}) + p_{2,i-1}(x_i - x_{i-1})^2 \quad (3-72)$$

Cette condition aboutit à

$$p_{0,i} = p_{0,i-1} + p_{1,i-1}(x_i - x_{i-1}) + p_{2,i-1}(x_i - x_{i-1})^2 \quad (3-73)$$

(2) la condition de continuité de la première dérivée à  $x = x_i$ :

$$\hat{G}^{(1)}(x_i; \mathbf{p}_{i-1}) = \hat{G}^{(1)}(x_i; \mathbf{p}_i) \quad (3-74)$$

i.e.,

$$y_i^{(1)} = \hat{G}^{(1)}(x_i; \mathbf{p}_i) = p_{1,i} \quad (3-75)$$

et

$$y_i^{(1)} = \hat{G}^{(1)}(x_i; \mathbf{p}_{i-1}) = p_{1,i-1} + 2p_{2,i-1}(x_i - x_{i-1}) \quad (3-76)$$

Cette condition aboutit à

$$p_{1,i} = p_{1,i-1} + 2p_{2,i-1}(x_i - x_{i-1}) \quad (3-77)$$

(3) la condition de continuité à  $x = x_{i+1}$ :

$$\hat{G}(x_{i+1}; \mathbf{p}_i) = \hat{G}(x_{i+1}; \mathbf{p}_{i+1}) \quad (3-78)$$

i.e.,

$$\hat{y}_{i+1} = \hat{G}(x_{i+1}; \mathbf{p}_{i+1}) = p_{0,i+1} \quad (3-79)$$

et

$$\hat{y}_{i+1} = \hat{G}(x_{i+1}; \mathbf{p}_i) = p_{0,i} + p_{1,i}(x_{i+1} - x_i) + p_{2,i}(x_{i+1} - x_i)^2 \quad (3-80)$$

Cette condition aboutit à

$$p_{0,i+1} = p_{0,i} + p_{1,i}(x_{i+1} - x_i) + p_{2,i}(x_{i+1} - x_i)^2 \quad (3-81)$$

(4) la condition de continuité de la première dérivée à  $x = x_{i+1}$ :

$$\hat{G}^{(1)}(x_{i+1}; \mathbf{p}_i) = \hat{G}^{(1)}(x_{i+1}; \mathbf{p}_{i+1}) \quad (3-82)$$

i.e.,

$$y_{i+1}^{(1)} = \hat{G}^{(1)}(x_{i+1}; \mathbf{p}_{i+1}) = p_{1,i+1} \quad (3-83)$$

et

$$y_{i+1}^{(1)} = \hat{G}^{(1)}(x_{i+1}; \mathbf{p}_i) = p_{1,i} + 2p_{2,i}(x_{i+1} - x_i) \quad (3-84)$$

Cette condition aboutit à

$$p_{1,i+1} = p_{1,i} + 2p_{2,i}(x_{i+1} - x_i) \quad (3-85)$$

(5) la condition de moindres carrés consiste à minimiser la fonction:

$$J[\mathbf{p}] = \sum_{i=1}^N [\hat{G}(x_i; \mathbf{p}_i) - y_i]^2 \quad (3-86)$$

Donc, les paramètres  $\mathbf{p}$  sont choisis par l'équation suivante:

$$\hat{\mathbf{p}} = \arg_p \inf \{J[\mathbf{p}]\} \quad (3-87)$$

(6) la condition aux limites "naturelle" est choisie comme la suivante:

$$\frac{d^2}{dx^2} \hat{G}(x; \mathbf{p}_i) \Big|_{x=x_i} = 0 \quad \text{pour } i=1 \quad (3-88)$$

Alors, à partir de (3-81), (3-85), (3-86) et (3-88) pour les sous intervalles  $i = 1, \dots, N-1$ , nous pouvons trouver la solution suivante pour la fonction spline de moindres carrés:

pour  $i = 1$ :

$$2p_{2,1} = 0 \quad (3-89)$$

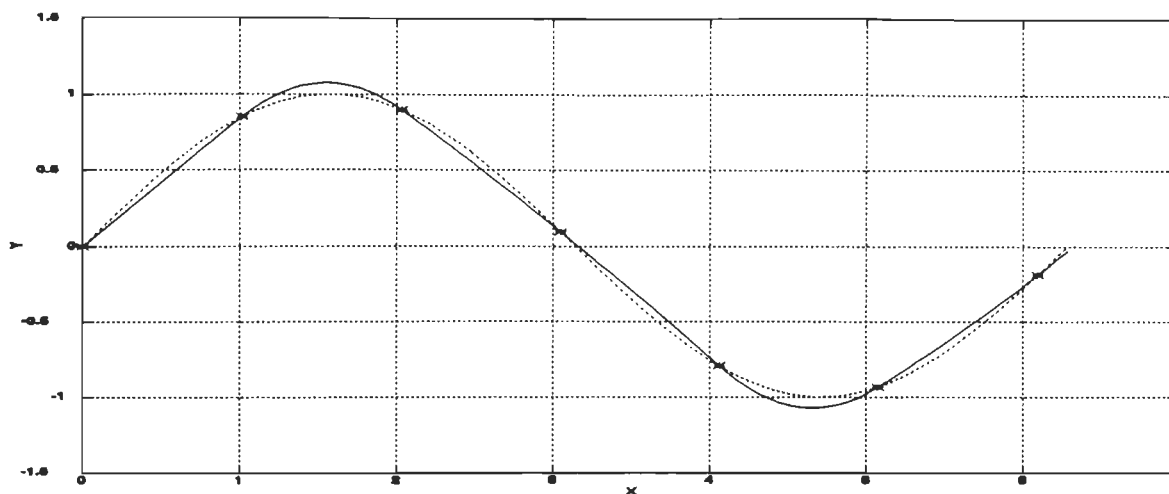
$$p_{1,1} = \frac{p_{0,2} - p_{0,1}}{(x_2 - x_1)} \quad (3-90)$$

pour  $i \neq 1$

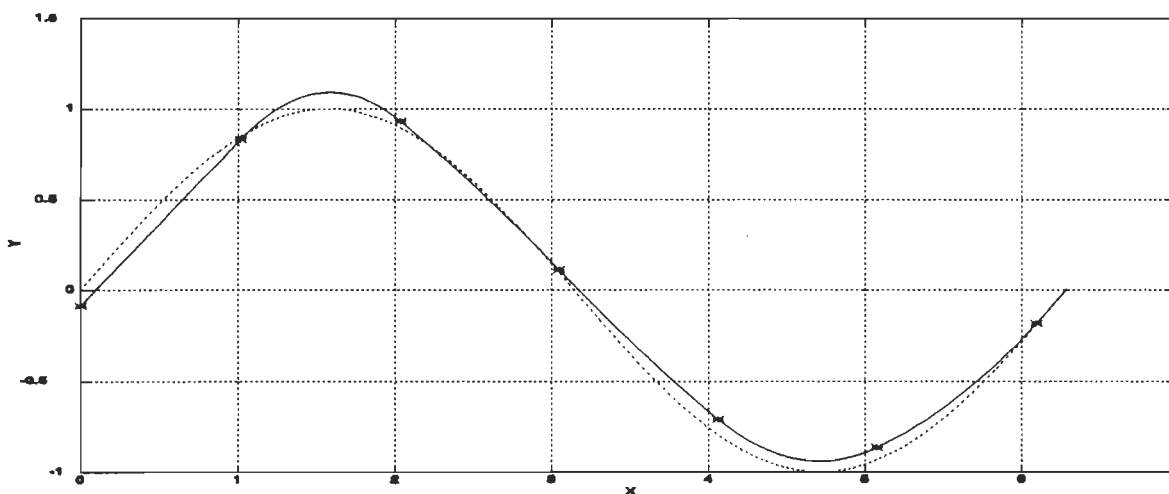
$$p_{1,i} = p_{1,i-1} + 2p_{2,i-1}(x_i - x_{i-1}) \quad (3-91)$$

$$p_{2,i} = \frac{p_{0,i+1} - p_{0,i} - p_{1,i}(x_{i+1} - x_i)}{(x_{i+1} - x_i)^2} \quad (3-92)$$

Les Fig.3-10 et Fig.3-11 montrent la fonction spline parabolique de moindres carrés pour l'approximation d'un signal sinusoidal dans les cas de données sans erreur et de données entachées par les erreurs.



*Fig.3-10. Les résultats de la fonction spline parabolique de moindres carrés dans les cas de données sans erreurs.*



*Fig.3-11. Les résultats de la fonction spline parabolique de moindres carrés dans les cas de données entachées par les erreurs.*

### 3.2.2. Fonction spline cubique d'approximation de moindres carrés

La routine *ParLsmSplC1* calcule les paramètres:

$$\mathbf{p}_i = [p_{0,i}, p_{1,i}, p_{2,i}, p_{3,i}] \quad (3-93)$$

dans la fonction spline cubique de moindres carrés

$$y = \hat{G}(x; \mathbf{p}_i) = p_{0,i} + p_{1,i}(x - x_i) + p_{2,i}(x - x_i)^2 + p_{3,i}(x - x_i)^3 \quad (3-94)$$

La première dérivée est donnée par la relation

$$y^{(1)} = \hat{G}^{(1)}(x; \mathbf{p}_i) = p_{1,i} + 2p_{2,i}(x - x_i) + 3p_{3,i}(x - x_i)^2 \quad (3-95)$$

et la deuxième dérivée par la relation

$$y^{(2)} = \hat{G}^{(2)}(x; \mathbf{p}_i) = 2p_{2,i} + 6p_{3,i}(x - x_i) \quad (3-96)$$

pour  $x \in (x_i, x_{i+1})$ ,  $i = 1, \dots, N-1$ , avec les conditions de continuité des première et deuxième dérivées pour approximer au maximum la fonction réelle.

(1) la condition de continuité à  $x = x_i$ :

$$\hat{G}(x_i; \mathbf{p}_{i-1}) = \hat{G}(x_i; \mathbf{p}_i) \quad (3-97)$$

i.e.,

$$\hat{y}_i = \hat{G}(x_i; \mathbf{p}_i) = p_{0,i} \quad (3-98)$$

et

$$\hat{y}_i = \hat{G}(x_i; \mathbf{p}_{i-1}) = p_{0,i-1} + p_{1,i-1}(x_i - x_{i-1}) + p_{2,i-1}(x_i - x_{i-1})^2 + p_{3,i-1}(x_i - x_{i-1})^3 \quad (3-99)$$

Cette condition aboutit à

$$p_{0,i} = p_{0,i-1} + p_{1,i-1}(x_i - x_{i-1}) + p_{2,i-1}(x_i - x_{i-1})^2 + p_{3,i-1}(x_i - x_{i-1})^3 \quad (3-100)$$

(2) la condition de continuité de la première dérivée à  $x = x_i$ :

$$\hat{G}^{(1)}(x_i; \mathbf{p}_{i-1}) = \hat{G}^{(1)}(x_i; \mathbf{p}_i) \quad (3-101)$$

i.e.,

$$y_i^{(1)} = \hat{G}^{(1)}(x_i; \mathbf{p}_i) = p_{1,i} \quad (3-102)$$

et

$$y_i^{(1)} = \hat{G}^{(1)}(x_i; \mathbf{p}_{i-1}) = p_{1,i-1} + 2p_{2,i-1}(x_i - x_{i-1}) + 3p_{3,i-1}(x_i - x_{i-1})^2 \quad (3-103)$$

Cette condition aboutit à

$$p_{1,i} = p_{1,i-1} + 2p_{2,i-1}(x_i - x_{i-1}) + 3p_{3,i-1}(x_i - x_{i-1})^2 \quad (3-104)$$

(3) la condition de continuité de la deuxième dérivée à  $x = x_i$ :

$$\hat{G}^{(2)}(x_i; \mathbf{p}_{i-1}) = \hat{G}^{(2)}(x_i; \mathbf{p}_i) \quad (3-105)$$

i.e.,

$$y_i^{(2)} = \hat{G}^{(2)}(x_i; \mathbf{p}_i) = 2p_{2,i} \quad (3-106)$$

et

$$y_i^{(2)} = \hat{G}^{(2)}(x_i; \mathbf{p}_{i-1}) = 2p_{2,i-1} + 6p_{3,i-1}(x_i - x_{i-1}) \quad (3-107)$$

Cette condition aboutit à

$$2p_{2,i} = 2p_{2,i-1} + 6p_{3,i-1}(x_i - x_{i-1}) \quad (3-108)$$

(4) la condition aux limites "naturelle" (ce choix requiert que la deuxième dérivée des points d'extrémité soit nulle):

$$\frac{d^2y}{dx^2} = \hat{G}^{(2)}(x; \mathbf{p}_i) = 2p_{2,i} + 6p_{3,i}(x - x_i) = 0 \text{ pour } x = x_1, x = x_N \quad (3-109)$$

i.e.,

$$y_1^{(2)} = 2p_{2,1} = 0 \quad (3-110)$$

Cette condition aboutit à

$$3(p_{0,2} - p_{0,1}) = 2y_1^{(1)}(x_2 - x_1) + y_2^{(1)}(x_2 - x_1) \quad (3-111)$$

et

$$y_N^{(2)} = 2p_{2,N-1} = 0 \quad (3-112)$$

Cette condition aboutit à

$$3(p_{0,N} - p_{0,N-1}) = 2y_N^{(1)}(x_N - x_{N-1}) + y_{N-1}^{(1)}(x_N - x_{N-1}) \quad (3-113)$$

(5) la condition de moindres carrés consiste à minimiser le critère:

$$J[\mathbf{p}] = \sum_{i=1}^N [\hat{G}(x_i; \mathbf{p}_i) - y_i]^2 \quad (3-114)$$

et les paramètres  $\mathbf{p}$  seront estimés en résolvant l'équation suivante:

$$\hat{\mathbf{p}} = \arg_p \inf \{J[\mathbf{p}]\} \quad (3-115)$$

Alors, nous avons les équations des paramètres par rapport à la première dérivée:

À partir de (3-115), nous avons

$$\{p_{0,i}\} = \arg_p \inf \left\{ \sum_{i=1}^N [p_{0,i} - y_i]^2 \right\} \quad \text{pour } i = 1, \dots, N \quad (3-116)$$

À partir de (3-102), nous avons

$$p_{1,i} = y_i^{(1)} \quad \text{pour } i = 2, \dots, N-1 \quad (3-117)$$

À partir de (3-99) et (3-103), nous avons

$$p_{2,i} = \frac{3(p_{0,i+1} - p_{0,i}) - 2y_i^{(1)}(x_{i+1} - x_i) - y_{i+1}^{(1)}(x_{i+1} - x_i)}{(x_{i+1} - x_i)^2} \quad \text{pour } i = 2, \dots, N-1 \quad (3-118)$$

$$p_{3,i} = \frac{-2(p_{0,i+1} - p_{0,i}) + y_i^{(1)}(x_{i+1} - x_i) + y_{i+1}^{(1)}(x_{i+1} - x_i)}{(x_{i+1} - x_i)^3} \quad \text{pour } i = 2, \dots, N-1 \quad (3-119)$$

et

$$3(p_{0,i+1} - p_{0,i-1}) = y_{i-1}^{(1)}(x_{i+1} - x_i) + 4y_i^{(1)}(x_{i+1} - x_i) + y_{i+1}^{(1)}(x_{i+1} - x_i) \quad \text{pour } i = 2, \dots, N-1 \quad (3-120)$$

C'est une équation linéaire par rapport au vecteur de valeurs de la première dérivée:

$$\begin{bmatrix}
 2(x_2-x_1) & (x_2-x_1) & 0 & \dots & \dots & \dots \\
 (x_3-x_2) & 4(x_3-x_2) & (x_3-x_2) & 0 & \dots & \dots \\
 0 & (x_4-x_3) & 4(x_4-x_3) & (x_4-x_3) & 0 & \dots \\
 \dots & \dots & \dots & \dots & \dots & \dots \\
 \dots & \dots & 0 & (x_N-x_{N-1}) & 4(x_N-x_{N-1}) & (x_N-x_{N-1}) \\
 \dots & \dots & \dots & 0 & (x_N-x_{N-1}) & 2(x_N-x_{N-1})
 \end{bmatrix}
 \begin{bmatrix}
 y_1^{(1)} \\
 y_2^{(1)} \\
 y_3^{(1)} \\
 \dots \\
 y_{N-1}^{(1)} \\
 y_N^{(1)}
 \end{bmatrix}
 =
 \begin{bmatrix}
 3(p_{0,2}-p_{0,1}) \\
 3(p_{0,3}-p_{0,1}) \\
 3(p_{0,4}-p_{0,2}) \\
 \dots \\
 3(p_{0,N}-p_{0,N-2}) \\
 3(p_{0,N}-p_{0,N-1})
 \end{bmatrix}
 \quad (3-121)$$

Les résultats de  $y_i^{(1)}$  sont utilisés dans les équations des paramètres de  $p_i$  dans (3-116) ~ (3-119). La condition aux limites "naturelle" est utilisée dans ce projet.

Les Fig.3-12 et Fig.3-13 montrent les résultats de reconstitution d'un signal sinusoidal par la fonction spline cubique de moindres carrés pour les cas de données sans erreur et de données entachées par les erreurs.

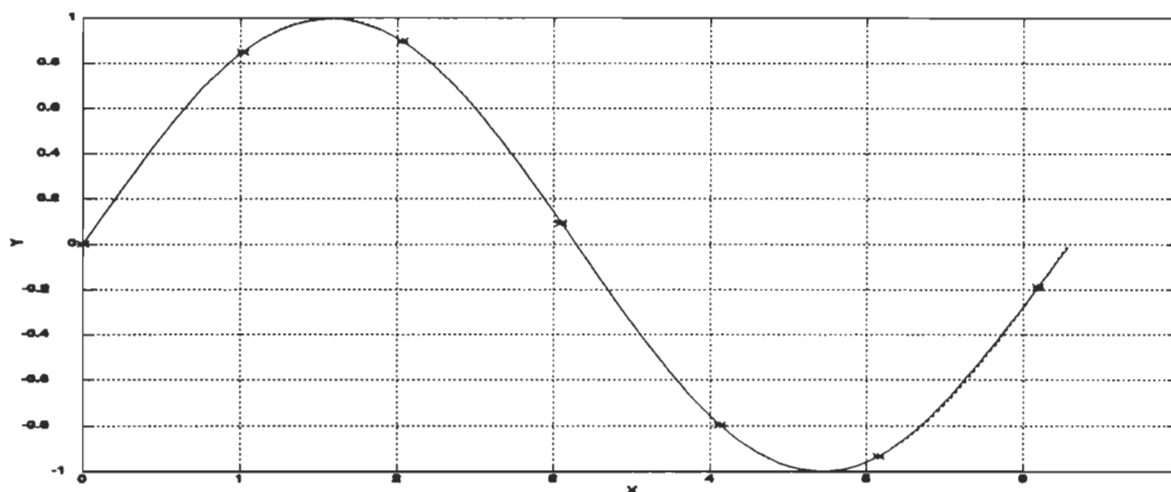


Fig.3-12. Les résultats de la fonction spline cubique de moindres carrés dans les cas de données sans erreurs.



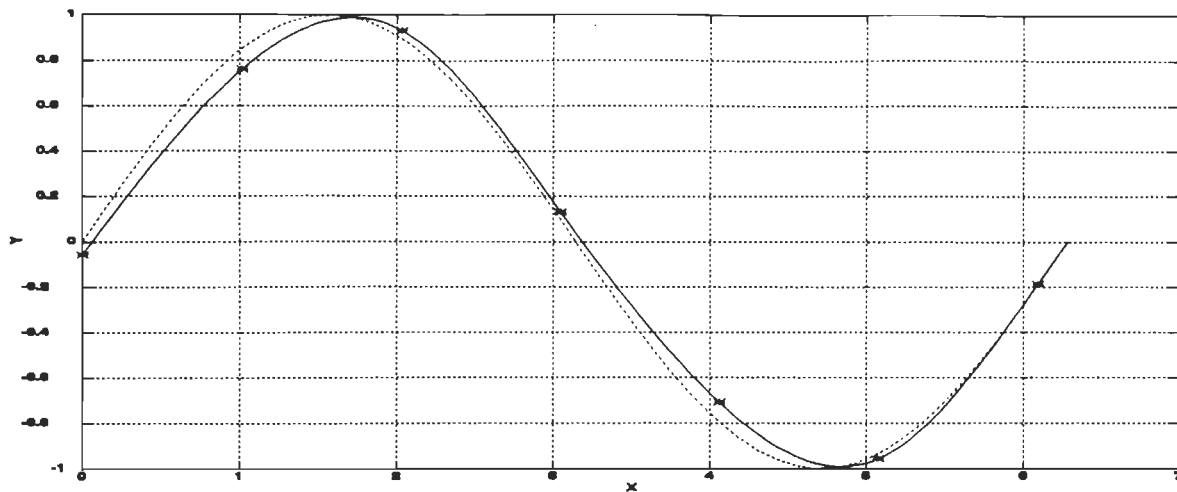


Fig.3-13. Les résultats de la fonction spline cubique de moindres carrés dans les cas de données entachées par les erreurs.

### 3.3. Fonctions spline de lissage

La méthode d'interpolation est utilisée dans le cas où les données estimées sont suffisamment précises. Autrement, la fonction spline de lissage peut nous donner une autre possibilité d'approximation dans le cas de données entachées par les erreurs. Les Fig.3-14 et Fig.3-15 montrent les différences entre la fonction spline cubique de lissage et la fonction spline cubique d'interpolation pour les cas de données sans erreur et avec erreur. Dans le cas de données sans erreur, les deux fonctions spline montrent les mêmes profils et les mêmes exactitudes. Dans le cas de données entachées par les erreurs, la fonction spline de lissage a un meilleur profil avec les mêmes exactitudes que la fonction spline cubique d'interpolation.

#### 3.3.1. Fonction spline parabolique de lissage [4]

La routine *ParSmoSplPI* calcule les paramètres

$$p_i = [p_{0,i}, p_{1,i}, p_{2,i}] \quad (3-122)$$

dans la fonction spline parabolique de lissage

$$y = \hat{G}(x; p_i) = p_{0,i} + p_{1,i}(x - x_i) + p_{2,i}(x - x_i)^2 \quad (3-123)$$

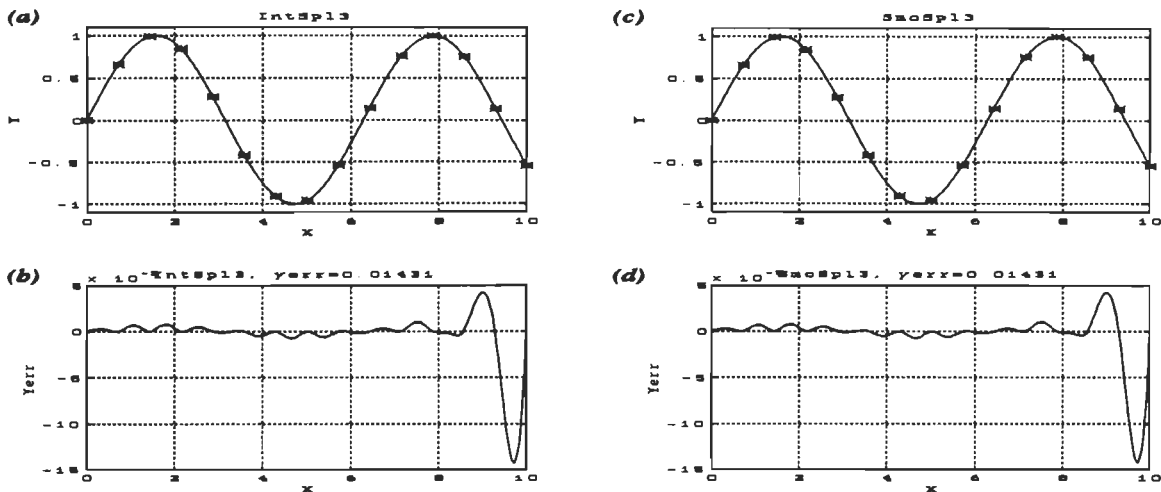
La première dérivée est donnée par la relation

$$y^{(1)} = \hat{G}^{(1)}(x; p_i) = p_{1,i} + 2p_{2,i}(x - x_i) \quad (3-124)$$

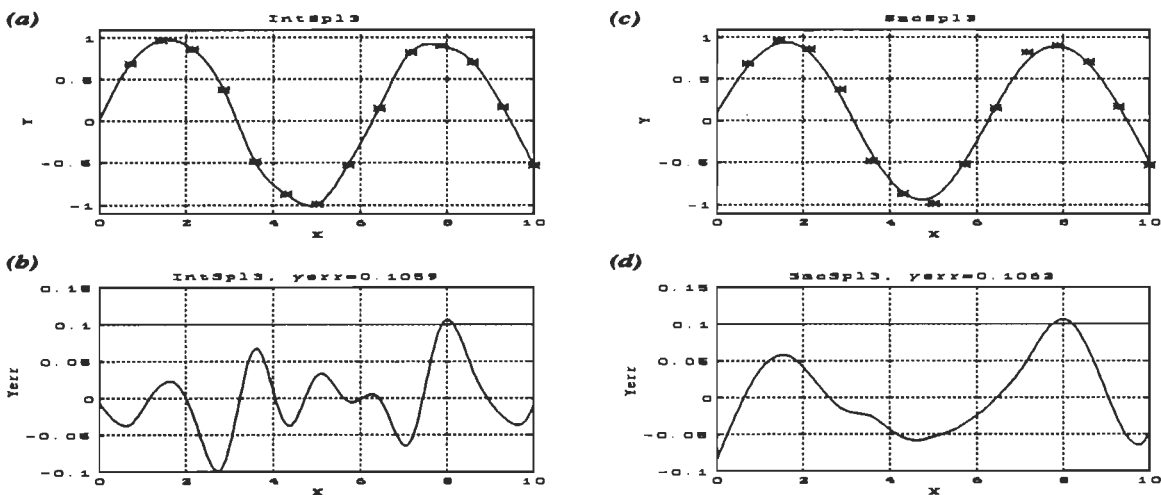
et la deuxième dérivée par la relation

$$y^{(2)} = \hat{G}^{(2)}(x; p_i) = 2p_{2,i} \tag{3-125}$$

pour  $i = 1, \dots, N-1$ . Il faut assurer la continuité de la fonction spline parabolique de lissage et de sa première dérivée pour approximer au maximum la fonction réelle.



**Fig.3-14.** Pour le cas de données sans erreurs: (a) la fonction spline cubique d'interpolation; et (b) ses erreurs; (c) la fonction spline cubique de lissage; et (d) ses erreurs.



**Fig.3-15.** Pour le cas de données entachées par les erreurs: (a) la fonction spline cubique d'interpolation; et (b) ses erreurs; (c) la fonction spline cubique de lissage; et (d) ses erreurs.

(1) la condition de continuité à  $x = x_i$ :

$$\hat{G}(x_i; \mathbf{p}_{i-1}) = \hat{G}(x_i; \mathbf{p}_i) \quad (3-126)$$

i.e.,

$$\hat{y}_i = \hat{G}(x_i; \mathbf{p}_i) = p_{0,i} \quad (3-127)$$

et

$$\hat{y}_i = \hat{G}(x_i; \mathbf{p}_{i-1}) = p_{0,i-1} + p_{1,i-1}(x_i - x_{i-1}) + p_{2,i-1}(x_i - x_{i-1})^2 \quad (3-128)$$

Cette condition aboutit à

$$p_{0,i} = p_{0,i-1} + p_{1,i-1}(x_i - x_{i-1}) + p_{2,i-1}(x_i - x_{i-1})^2 \quad (3-129)$$

(2) la condition de continuité de la première dérivée à  $x = x_i$ :

$$\hat{G}^{(1)}(x_i; \mathbf{p}_{i-1}) = \hat{G}^{(1)}(x_i; \mathbf{p}_i) \quad (3-130)$$

i.e.,

$$y_i^{(1)} = \hat{G}^{(1)}(x_i; \mathbf{p}_i) = p_{1,i} \quad (3-131)$$

et

$$y_i^{(1)} = \hat{G}^{(1)}(x_i; \mathbf{p}_{i-1}) = p_{1,i-1} + 2p_{2,i-1}(x_i - x_{i-1}) \quad (3-132)$$

Cette condition aboutit à

$$p_{1,i} = p_{1,i-1} + 2p_{2,i-1}(x_i - x_{i-1}) \quad (3-133)$$

Alors, pour les sous intervalles  $i = 2, \dots, N-1$ , nous avons la forme matricielle:

$$\begin{bmatrix} p_{1,i-1} + 2p_{2,i-1}(x_i - x_{i-1}) \\ y_i \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & (x_{i+1} - x_i) & (x_{i+1} - x_i)^2 \end{bmatrix} \begin{bmatrix} p_{0,i} \\ p_{1,i} \\ p_{2,i} \end{bmatrix} \quad (3-134)$$

(3) la condition de lissage:

$$\sum_{i=1}^N [\hat{G}(x_i; \mathbf{p}_i) - y_i] = 0 \quad (3-135)$$

$$\sum_{i=1}^N [\hat{G}(x_i; \mathbf{p}_i) - y_i]^2 = s \hat{\sigma}_y^2 \quad (3-136)$$

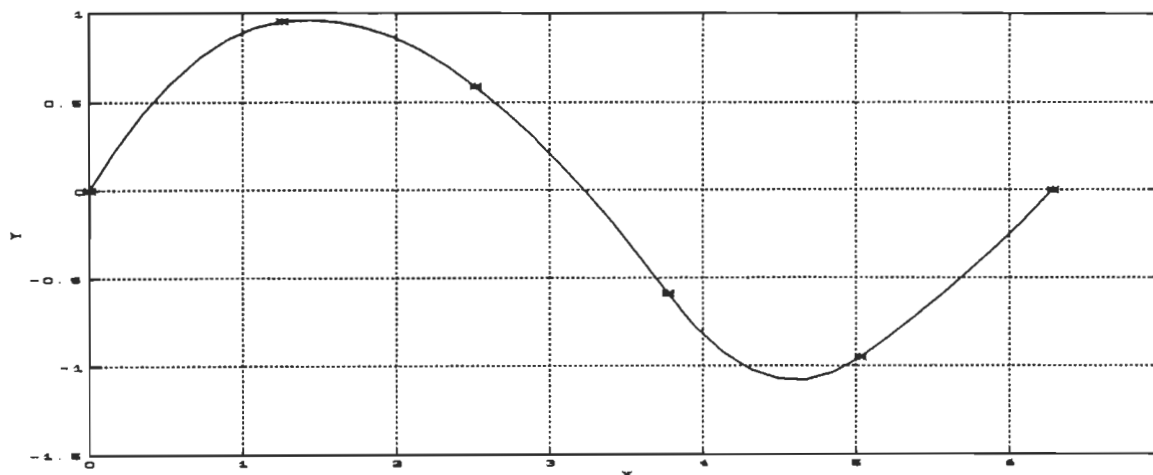
où  $\hat{\sigma}_y^2$  est l'estimation *a priori* de la variance des erreurs dans les données  $\{y_i\}$ , et  $s$  est le paramètre de lissage (typiquement  $s = N$  ou  $s = N-1$ ). Les paramètres libres ( $N+1$ ) sont déterminés par la minimisation du critère:

$$J[\mathbf{p}] = \frac{1}{8} \int_{x_1}^{x_N} \left| \frac{d^2}{dx^2} \hat{G}(x; \mathbf{p}) \right|^2 dx \quad (3-137)$$

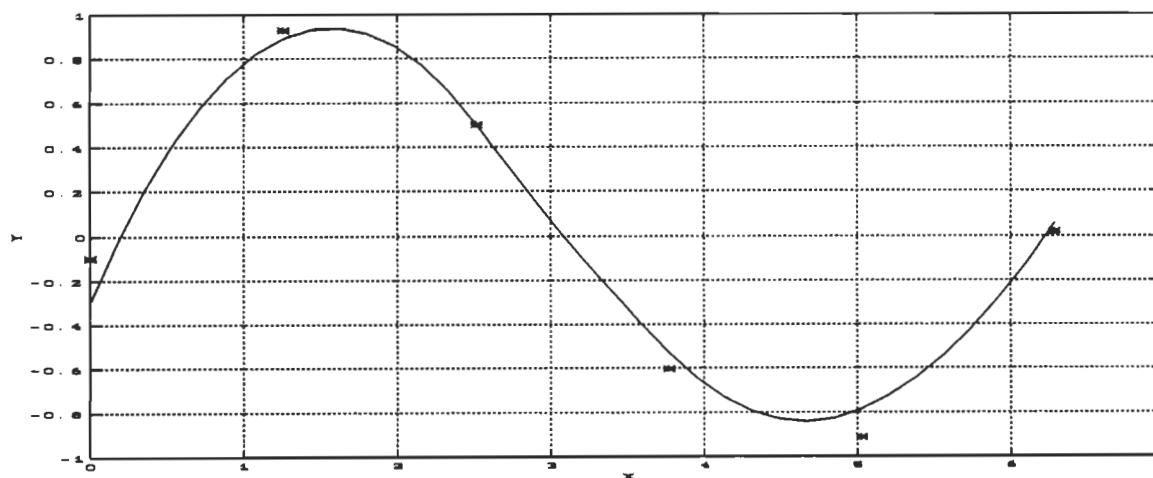
i.e., les paramètres  $\mathbf{p}$  sont choisis par l'équation suivante:

$$\hat{\mathbf{p}} = \arg_p \inf \{J[\mathbf{p}]\} \quad (3-138)$$

Les Fig.3-16 et Fig.3-17 montrent les résultats de reconstitution d'un signal sinusoidal par la fonction spline parabolique de lissage pour les cas de données sans erreur et avec entachées par erreur.



**Fig.3-16.** Les résultats de la fonction spline parabolique de lissage dans les cas de données sans erreurs.



**Fig.3-17.** Les résultats de la fonction spline parabolique de lissage dans les cas de données entachées par les erreurs.

### 3.3.2. Fonction spline cubique de lissage

La routine *ParSmoSplCI* calcule les paramètres

$$\mathbf{p}_i = [p_{0,i}, p_{1,i}, p_{2,i}, p_{3,i}] \quad \text{pour } i = 1, \dots, N-1 \quad (3-139)$$

dans la fonction spline cubique de lissage

$$y = \hat{G}(x; \mathbf{p}_i) = p_{0,i} + p_{1,i}(x - x_i) + p_{2,i}(x - x_i)^2 + p_{3,i}(x - x_i)^3 \quad \text{pour } i = 1, \dots, N-1 \quad (3-140)$$

La première dérivée est donnée par la relation

$$y^{(1)} = \hat{G}^{(1)}(x; \mathbf{p}_i) = p_{1,i} + 2p_{2,i}(x - x_i) + 3p_{3,i}(x - x_i)^2 \quad \text{pour } i = 1, \dots, N-1 \quad (3-141)$$

et la deuxième dérivée par la relation

$$y^{(2)} = \hat{G}^{(2)}(x; \mathbf{p}_i) = 2p_{2,i} + 6p_{3,i}(x - x_i) \quad \text{pour } i = 1, \dots, N-1 \quad (3-142)$$

Il faut assurer la continuité de la fonction de lissage, de sa première et de sa seconde dérivée pour approximer au maximum la fonction réelle.

(1) la condition de continuité à  $x = x_i$ :

$$\hat{G}(x_i; \mathbf{p}_{i-1}) = \hat{G}(x_i; \mathbf{p}_i) \quad (3-143)$$

i.e.,

$$\hat{y}_i = \hat{G}(x_i; \mathbf{p}_i) = p_{0,i} \quad (3-144)$$

et

$$\hat{y}_i = \hat{G}(x_i; \mathbf{p}_{i-1}) = p_{0,i-1} + p_{1,i-1}(x_i - x_{i-1}) + p_{2,i-1}(x_i - x_{i-1})^2 + p_{3,i-1}(x_i - x_{i-1})^3 \quad (3-145)$$

Cette condition aboutit à

$$p_{0,i} = p_{0,i-1} + p_{1,i-1}(x_i - x_{i-1}) + p_{2,i-1}(x_i - x_{i-1})^2 + p_{3,i-1}(x_i - x_{i-1})^3 \quad (3-146)$$

(2) la condition de continuité de la première dérivée à  $x = x_i$ :

$$\hat{G}^{(1)}(x_i; \mathbf{p}_{i-1}) = \hat{G}^{(1)}(x_i; \mathbf{p}_i) \quad (3-147)$$

i.e.,

$$y_i^{(1)} = \hat{G}^{(1)}(x_i; \mathbf{p}_i) = p_{1,i} \quad (3-148)$$

et

$$y_i^{(1)} = \hat{G}^{(1)}(x_i; \mathbf{p}_{i-1}) = p_{1,i-1} + 2p_{2,i-1}(x_i - x_{i-1}) + 3p_{3,i-1}(x_i - x_{i-1})^2 \quad (3-149)$$

Cette condition aboutit à

$$p_{1,i} = p_{1,i-1} + 2p_{2,i-1}(x_i - x_{i-1}) + 3p_{3,i-1}(x_i - x_{i-1})^2 \quad (3-150)$$

(3) la condition de continuité de la deuxième dérivée à  $x = x_i$ :

$$\hat{G}^{(2)}(x_i; \mathbf{p}_{i-1}) = \hat{G}^{(2)}(x_i; \mathbf{p}_i) \quad (3-151)$$

i.e.,

$$y_i^{(2)} = \hat{G}^{(2)}(x_i; \mathbf{p}_i) = 2p_{2,i} \quad (3-152)$$

et

$$y_i^{(2)} = \hat{G}^{(2)}(x_i; \mathbf{p}_{i-1}) = 2p_{2,i-1} + 6p_{3,i-1}(x_i - x_{i-1}) \quad (3-153)$$

Cette condition aboutit à

$$2p_{2,i} = 2p_{2,i-1} + 6p_{3,i-1}(x_i - x_{i-1}) \quad (3-154)$$

(4) la condition de lissage:

Les paramètres libres ( $N+1$ ) sont déterminés par la minimisation du critère [5]:

$$J = s \sum_{i=1}^N [\hat{G}(x_i; \mathbf{p}_i) - y_i]^2 + (1-s) \int_{x_1}^{x_n} \left| \frac{d^2}{dx^2} \hat{G}(x; \mathbf{p}) \right|^2 dx \quad (3-155)$$

où  $s \in [0, 1]$  est un paramètre de lissage. Pour les données de l'étalonnage sans erreur  $s = 1$ , pour les données entachées par les erreurs,  $s$  est optimisé par l'utilisation de la méthode de *L-courbe* [9] (cf. Annexe A.1. pour les détails). Les paramètres  $\mathbf{p}$  sont choisis par l'équation suivante:

$$\hat{\mathbf{p}} = \arg_p \inf \{J[\mathbf{p}]\} \quad (3-156)$$

L'équation (3-155) établit une sorte de compromis entre le désir de rester près des données d'étalonnage et le désir d'obtenir une fonction de lissage. C'est-à-dire que la courbe spline approche le plus possible tous les noeuds au lieu de les traverser pour former une courbe plus lisse. Il y a un peu de différence entre la courbe spline et les noeuds. Mais, la courbe spline permet d'avoir un lissage satisfaisant. Le choix du paramètre  $s$  dépend des deux buts incompatibles.

(5) la condition aux limites:

Deux paramètres libres  $p_{1,1}$  et  $p_{1,N}$  sont déterminés selon la condition aux limites "naturelle":

$$\frac{d^2}{dx^2} \hat{G}(x; \mathbf{p}) \Big|_{x=x_i} = 0 \quad \text{pour } i=1 \text{ et } N \quad (3-157)$$

Selon la condition aux limites "naturelle" et la première dérivée continue, nous avons

$$p_{0,i} = \hat{y}_i \quad (3-158)$$

$$p_{2,i} = y_i^{(2)}/2 \quad (3-159)$$

pour  $i = 1, \dots, N$ . Ainsi, nous obtenons le système linéaire suivant

$$p_{2,i} = 0 \quad (3-160)$$

$$p_{2,i-1} \Delta x_{i-1} + 2p_{2,i} (\Delta x_{i-1} + \Delta x_i) + p_{2,i+1} \Delta x_i = 3 \left( \frac{\Delta p_{0,i}}{\Delta x_i} - \frac{\Delta p_{0,i-1}}{\Delta x_{i-1}} \right) \quad \text{pour } i = 2, \dots, N-1 \quad (3-161)$$

$$p_{2,N} = 0 \quad (3-162)$$

où  $\Delta x_i = x_{i+1} - x_i$ ,  $\Delta p_{0,i} = p_{0,i+1} - p_{0,i}$  pour  $i = 1, \dots, N-2$ .  $\mathbf{x} = \{x_i\}$  pour  $i = 1, \dots, N-1$ ,  $\mathbf{p}_0 = \{p_{0,i}\}$  pour  $i = 1, \dots, N-2$ . Nous définissons plusieurs équations suivantes:

$$y^{(1)} = \frac{\Delta y}{\Delta x} \quad \text{pour } i = 1, \dots, N-1 \quad (3-163)$$

où  $\Delta y_i = y_{i+1} - y_i$  pour  $i = 1, \dots, N-1$ .

$$\Delta y_i^{(1)} = y_{i+1}^{(1)} - y_i^{(1)} \quad \text{pour } i = 1, \dots, N-2 \quad (3-164)$$

Par combinaison des conditions de la fonction spline cubique de lissage (1), (2) et (3), nous pouvons obtenir les matrices et les vecteurs auxiliaires  $(N-2 \times N-2)$  suivantes:

(1) la matrice auxiliaire  $(N-2 \times N-2)$ :

$$R = \begin{bmatrix} 2(\Delta x_1 + \Delta x_2) & \Delta x_2 & 0 & \dots & \dots \\ \Delta x_2 & 2(\Delta x_2 + \Delta x_3) & \Delta x_3 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \Delta x_{i-1} & 2(\Delta x_{i-1} + \Delta x_i) & \Delta x_i & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & 0 & \Delta x_{N-2} & 2(\Delta x_{N-2} + \Delta x_{N-1}) \end{bmatrix} \quad (3-165)$$

pour  $i = 2, \dots, N-1$ .

(2) la matrice auxiliaire  $(N-2 \times N)$ :

$$Q = \begin{bmatrix} 1/\Delta x_1 & -1/\Delta x_1 - 1/\Delta x_2 & 1/\Delta x_2 & 0 & \dots & \dots \\ 0 & 1/\Delta x_2 & -1/\Delta x_2 - 1/\Delta x_3 & 1/\Delta x_3 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & 1/\Delta x_{i-1} & -1/\Delta x_{i-1} - 1/\Delta x_i & 1/\Delta x_i & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & 0 & 1/\Delta x_{N-2} & -1/\Delta x_{N-2} - 1/\Delta x_{N-1} & 1/\Delta x_{N-1} \end{bmatrix} \quad (3-166)$$

pour  $i = 2, \dots, N-1$ .

(3) le vecteur auxiliaire  $(N-2 \times 1)$ :

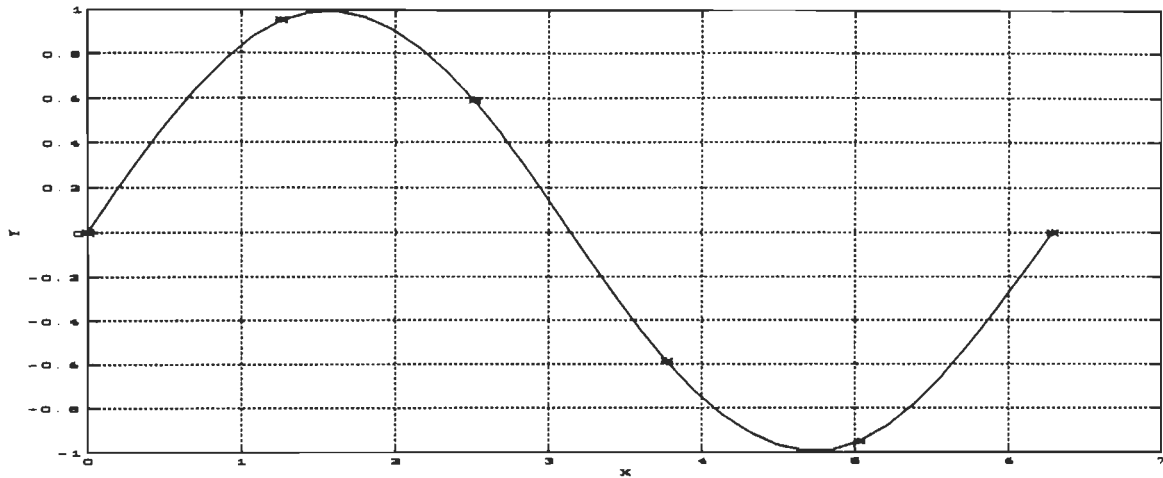
$$u = [6(1-s)QQ^T + sR]^{-1} \times Qy \quad (3-167)$$

(4) le vecteur auxiliaire  $(N \times 1)$ :

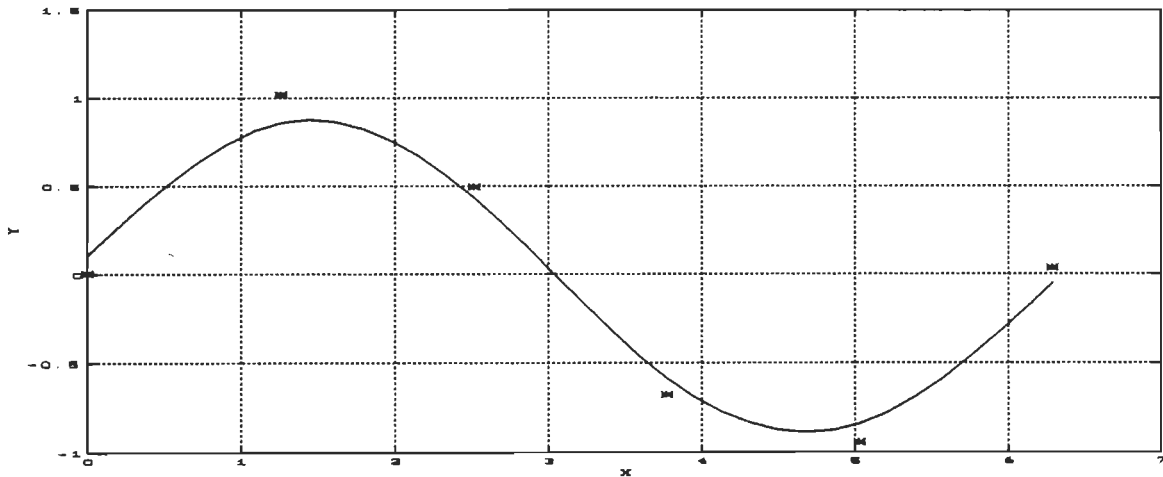
$$v = y - 6(1-s)Q^T u \quad (3-168)$$

(5) le vecteur auxiliaire  $(N \times 1)$ :

$$w = \begin{bmatrix} 0 \\ su \\ 0 \end{bmatrix} \quad (3-169)$$



**Fig.3-18.** Les résultats de la fonction spline cubique de lissage dans le cas de données sans erreurs.



**Fig.3-19.** Les résultats de la fonction spline cubique de lissage dans le cas de données entachées par les erreurs.

Selon les matrices et les vecteurs, nous avons les paramètres  $(1 \times N-1)$  suivants de la fonction spline cubique de lissage:



$$p_3 = \left[ \frac{\Delta w}{\Delta x} \right]^T = \left\{ \frac{w_{i+1} - w_i}{x_{i+1} - x_i} \right\}^T \quad \text{pour } i = 1, \dots, N-1 \quad (3-170)$$

$$p_2 = \left[ \begin{array}{c} 0 \\ 3su \end{array} \right]^T \quad (3-171)$$

$$p_1 = \left\{ \frac{\Delta v}{\Delta x} - \Delta x \left[ \begin{array}{c} 0 \\ 2su \end{array} \right] + \left[ \begin{array}{c} su \\ 0 \end{array} \right] \right\}^T \quad (3-172)$$

$$p_0 = \{v_i\} \quad \text{pour } i = 1, \dots, N-1 \quad (3-173)$$

où,  $p_0 = \{p_{0,i}\}$ ,  $p_1 = \{p_{1,i}\}$ ,  $p_2 = \{p_{2,i}\}$ ,  $p_3 = \{p_{3,i}\}$ , pour  $i = 1, \dots, N-1$ .

Les Fig.3-18 et Fig.3-19 montrent les résultats de reconstitution d'un signal sinusoidal par la fonction spline cubique de lissage pour les cas de données sans erreur et avec erreur.

### 3.3.3. Fonction Beta-spline cubique

La courbe de spline formée par la fonction *Beta-spline SmoSplβ1* ne passe pas à travers les noeuds, elle est tirée proche d'eux. Il y a plusieurs méthodes pour former les différentes fonctions *Beta-spline*. Nous choisissons la fonction *Beta-spline* avec le profil uniforme. L'équation générale pour la fonction *Beta-spline* avec la représentation paramétrique est la suivante [9-§14], [3-§7]:

$$\hat{x}_i(\tau) = \sum_{r=0}^3 x_{i+r} b_r(\beta_1, \beta_2, \tau) = x b_\beta(\beta_1, \beta_2, \tau) = p_{0,i} + p_{1,i}\tau + p_{2,i}\tau^2 + p_{3,i}\tau^3$$

$$\hat{y}_i(\tau) = \sum_{r=0}^3 y_{i+r} b_r(\beta_1, \beta_2, \tau) = y b_\beta(\beta_1, \beta_2, \tau) = q_{0,i} + q_{1,i}\tau + q_{2,i}\tau^2 + q_{3,i}\tau^3 \quad (3-174)$$

où,  $p_i = [p_{0,i}, p_{1,i}, p_{2,i}, p_{3,i}]$  et  $q_i = [q_{0,i}, q_{1,i}, q_{2,i}, q_{3,i}]$  sont les paramètres de la fonction *Beta-spline*, pour  $i = 1, \dots, N-1$ .  $\{\hat{x}_i, \hat{y}_i \mid i = 1, \dots, N-1\}$  est le point à estimer au  $i^e$  segment de la fonction *Beta-spline*.  $\{x_i, y_i \mid i = 1, \dots, N-1\}$  est le noeud pour l'étalonnage.  $b_\beta(\beta_1, \beta_2, \tau)$  est le vecteur des fonctions de base.  $\tau$  est le paramètre dans le sous intervalle  $(x_i, x_{i+1})$ ,  $\tau \in [0 \sim 1]$ .

Il suit de cette équation que les paramètres de la fonction *Beta-spline* correspondant au segment entre les  $i^e$  et  $i-1^e$  points sont complètement déterminés par:

$$\{x_{i-2}, y_{i-2}\}, \{x_{i-1}, y_{i-1}\}, \{x_i, y_i\}, \{x_{i+1}, y_{i+1}\} \quad (3-175)$$

3.3.3.1. *La continuité géométrique et les paramètres de profil [3-§6] (y est utilisé comme exemple, x a la même formule que y):*

## (1) la continuité du vecteur tangent unité

Le vecteur tangent unité à la courbe paramétrique a la même direction et le même sens que le vecteur paramétrique de la première dérivée, mais il est normalisé. Supposant une première dérivée non nulle, il est exprimé par:

$$\frac{y^{(1)}(\tau)}{|y^{(1)}(\tau)|} \quad (3-176)$$

Pour les deux courbes  $y_{i-1}(\tau)$  et  $y_i(\tau)$ , nous considérons le joint  $\frac{y_{i-1}(\tau)}{y_i(\tau)}$ . La continuité du vecteur de tangente unité est réalisée si

$$\frac{y_{i-1}^{(1)}(1)}{|y_{i-1}^{(1)}(1)|} = \frac{y_i^{(1)}(0)}{|y_i^{(1)}(0)|} \quad (3-177)$$

$$y_{i-1}^{(1)}(1) \frac{|y_i^{(1)}(0)|}{|y_{i-1}^{(1)}(1)|} = y_i^{(1)}(0) \quad (3-178)$$

ou

$$y_i^{(1)}(0) = \beta_1 y_{i-1}^{(1)}(1) \quad (3-179)$$

## (2) La continuité du vecteur de courbure

Le vecteur de courbure est défini par:

$$\frac{(y^{(1)}(\tau) \times y^{(2)}(\tau)) \times y^{(1)}(\tau)}{|y^{(1)}(\tau)|^4} \quad (3-180)$$

Le problème est de déterminer la condition appropriée à  $y_i^{(2)}(0)$  pour maintenir la continuité du vecteur de courbure sur la contrainte au vecteur de la première dérivé qui est donnée par (3-179):

$$\frac{(\beta_1 y_{i-1}^{(1)}(1) \times y_i^{(2)}(0)) \times \beta_1 y_{i-1}^{(1)}(1)}{|\beta_1 y_{i-1}^{(1)}(1)|^4} = \frac{(y_{i-1}^{(1)}(1) \times y_{i-1}^{(2)}(1)) \times y_{i-1}^{(1)}(1)}{|y_{i-1}^{(1)}(1)|^4} \quad (3-181)$$

Il est facile de trouver que la condition suffisante est

$$y_i^{(2)}(0) = \beta_1^2 y_{i-1}^{(2)}(1) \quad (3-182)$$

En général,  $y_i^{(2)}(0)$  peut aussi inclure un terme supplémentaire qui est un multiple de  $y_{i-1}^{(1)}(1)$

$$y_i^{(2)}(0) = \beta_1^2 y_{i-1}^{(2)}(1) + \beta_2 y_{i-1}^{(1)}(1) \quad (3-183)$$

Cette condition garantit la continuité du vecteur de courbure.

## (3) Les paramètres de profil

Les contraintes de continuité géométrique peuvent être appliquées au joint des segments  $i^e$  et  $i-1^e$

pour produire les conditions suivantes:

$$\begin{aligned} y_i(0) &= y_{i-1}(1); \\ y_i^{(1)}(0) &= \beta_1 y_{i-1}^{(1)}(1); \\ y_i^{(2)}(0) &= \beta_1^2 y_{i-1}^{(2)}(1) + \beta_2 y_{i-1}^{(1)}(1). \end{aligned} \quad (3-184)$$

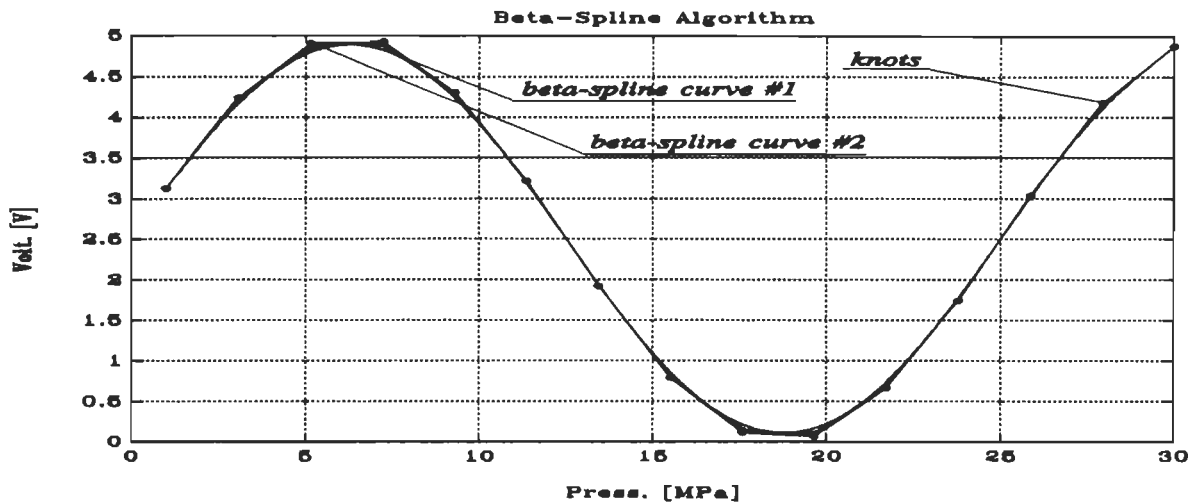


Fig.3-20. Les résultats de la fonction Beta-spline avec les différents paramètres profilés, i.e. #1:  $\beta_1=1, \beta_2=0$ ; #2:  $\beta_1=1, \beta_2=10$ .

Les  $\beta_1, \beta_2$  sont appelés les paramètres de profil parce qu'ils contrôlent le profil Beta-spline. En augmentant  $\beta_1$ , nous augmentons la "vélocité", autrement dit, nous traversons un segment immédiatement après le point de jonction, en tenant compte de la "vélocité" juste avant le point de jonction, ainsi servant au biais du segment. Quand  $\beta_1 > 1$ , le segment du point  $(i+1)^{e}$  approche plus de la direction de la tangente que le segment du point  $i^e$ . Quand la valeur du  $\beta_1$  se situe entre 0 et 1, la situation est inversée.  $\beta_1$  mesure l'influence relative de la tangente à côté du points de jonction. Alors, il est appelé le *biais* au point de jonction. Le paramètre  $\beta_2$  sert à contrôler la tension sur le spline. Le changement de la valeur du  $\beta_2$  déplace le point de jonction entre  $\{x_{i-1}, y_{i-1}\}$  et  $\{x_i, y_i\}$  le long du vecteur qui passe à travers le noeud  $\{x_{i-2}, y_{i-2}\}$ , et il se produit simultanément pour tous les points de jonction sur le spline avec un profil uniforme. Par exemple, l'augmentation de  $\beta_2$  déplacent chaque point de jonction vers son noeud correspondant. Quand  $\beta_2$  augmente, le sommet de la fonction de base s'approche de un et aux "queues" de la fonction de base, i.e. sur les intervalles de son support à l'extrême gauche et à l'extrême droite, elle approche zéro. En général, le spline s'approche aux données exactes quand  $\beta_2$  tend vers l'infini; les points de jonction entre les segments convergent vers les noeuds. Le paramètre  $\beta_2$  donne l'outil mathématique de tension sur Beta-spline. Les résultats de

reconstitution d'une sinusoïde par la fonction *Beta*-spline avec différents paramètres profilés sont montrés sur la Fig.3-20.

### 3.3.3.2. La procédure *Beta*-spline pour l'approximation

Le point au  $i^e$  segment est une moyenne pondérée des quatre vecteurs de noeud  $\{x_{i+r}, y_{i+r}\}$ , ( $r = -2, -1, 0, 1$ ). Les facteurs de pondération  $b_r(\beta_1, \beta_2, \tau)$ , ( $r = -2, -1, 0, 1$ ), sont des fonctions scalaires qui sont évaluées à plusieurs valeurs sur le domaine du paramètre  $u$ , et de chaque paramètre profilé  $\beta_1, \beta_2$ . Chaque segment *Beta*-spline avec  $\beta_1 > 0, \beta_2 \geq 0$  a une représentation unique comme une combinaison linéaire de ces fonctions, et les deux paramètres  $\beta_1$  et  $\beta_2$  sont linéairement indépendants. Chaque fonction est un polynôme cubique en  $\tau$  dont les coefficients sont des fonctions elles-mêmes des  $\beta_1, \beta_2$  [20-§4]:

$$\begin{aligned} b_r(\beta_1, \beta_2, \tau) &= \sum_{g=0}^3 w_{g,r}(\beta_1, \beta_2) \tau^g \\ &= w_{0,r}(\beta_1, \beta_2) + w_{1,r}(\beta_1, \beta_2) \tau + w_{2,r}(\beta_1, \beta_2) \tau^2 + w_{3,r}(\beta_1, \beta_2) \tau^3 \end{aligned} \quad (3-185)$$

pour  $0 \leq \tau < 1$  et  $r = -2, -1, 0, 1$ . Les coefficients  $w_{g,r}(\beta_1, \beta_2) \tau^g$ ,  $g = 0, 1, 2, 3$  et  $r = -2, -1, 0, 1$ , doivent être déterminés pour satisfaire les contraintes de continuité géométrique données par (3-184).

Selon (3-185), nous avons

$$b_\beta(\beta_1, \beta_2, \tau) = \begin{bmatrix} b_{-2}(\beta_1, \beta_2, \tau) \\ b_{-1}(\beta_1, \beta_2, \tau) \\ b_0(\beta_1, \beta_2, \tau) \\ b_1(\beta_1, \beta_2, \tau) \end{bmatrix} = \begin{bmatrix} \sum_{g=0}^3 w_{g,-2}(\beta_1, \beta_2) \tau^g \\ \sum_{g=0}^3 w_{g,-1}(\beta_1, \beta_2) \tau^g \\ \sum_{g=0}^3 w_{g,0}(\beta_1, \beta_2) \tau^g \\ \sum_{g=0}^3 w_{g,1}(\beta_1, \beta_2) \tau^g \end{bmatrix} \quad (3-186)$$

En accomplissant les différenciations nécessaires pour les fonctions  $b_r$ , et les évaluant à  $\tau = 0$  et  $\tau = 1$  nous pouvons obtenir les fonctions de pondération suivantes:

$$\begin{aligned}
 W_{\beta}(\beta_1, \beta_2) &= \begin{bmatrix} w_{0,-2} & w_{0,-1} & w_{0,0} & w_{0,1} \\ w_{1,-2} & w_{1,-1} & w_{1,0} & w_{1,1} \\ w_{2,-2} & w_{2,-1} & w_{2,0} & w_{2,1} \\ w_{3,-2} & w_{3,-1} & w_{3,0} & w_{3,1} \end{bmatrix} \\
 &= \begin{bmatrix} \frac{2\beta_1^3}{\delta} & \frac{4\beta_1^2 + 4\beta_1 + \beta_2}{\delta} & \frac{2}{\delta} & 0 \\ \frac{-6\beta_1^3}{\delta} & \frac{6\beta_1^3 - 6\beta_1}{\delta} & \frac{6\beta_1}{\delta} & 0 \\ \frac{6\beta_1^3}{\delta} & \frac{-6\beta_1^3 - 6\beta_1^2 - 3\beta_2}{\delta} & \frac{6\beta_1^3 + 3\beta_2}{\delta} & 0 \\ \frac{-2\beta_1^3}{\delta} & \frac{2\beta_1^3 + 2\beta_1^2 + \beta_1 + \beta_2}{\delta} & \frac{-2\beta_1^2 - 2\beta_1 - 2\beta_2 - 2}{\delta} & \frac{2}{\delta} \end{bmatrix} \quad (3-187)
 \end{aligned}$$

où  $\delta = 2\beta_1^3 + 4\beta_1^2 + 4\beta_1 + \beta_2 + 2$ .

Pour le but de l'étalonnage, les valeurs  $\beta_1 = 1$ ,  $\beta_2 = 0$  sont adoptées parce qu'elles permettent la continuité de la première et la deuxième dérivé paramétrique de la fonction *Beta-spline* et produisent un *Beta-spline* avec le profil uniforme. Les valeurs des  $\beta_1$  et  $\beta_2$  affectent la courbe entière de la fonction *Beta-spline*.

La définition complète du segment de spline par une équation ouverte *Beta-spline* requiert la spécification de la condition aux limites pour une courbe ou une surface. Nous adoptons la technique de double noeud qui définit un segment additionnel sur chaque extrémité par la répétition des noeuds d'extrémité dans l'équation *Beta-spline*, i.e.,  $x_{-1} = x_0$ ,  $y_{-1} = y_0$ ,  $x_N = x_{N-1}$ ,  $y_N = y_{N-1}$ . Le nombre actuel de noeuds est  $N$ , l'index de noeuds est  $[0 \sim N-1]$ .

L'approximation *Beta-spline* réside dans le calcul de la matrice  $W_{\beta}(\beta_1, \beta_2)$  à l'avance et ensuite de chaque point d'échantillon  $\{x_i(\tau), y_i(\tau)\}$ ,  $\tau \in [0 \sim 1]$ :

- calcul de valeurs de  $b_{\beta}(\beta_1, \beta_2, \tau)$  selon (3-186);
- détermination des coordonnées de  $\{x_i(\tau), y_i(\tau)\}$  dans (3-174).

Les résultats exemplaires d'approximation d'un signal sinusoidal par *Beta-spline* sont montrés aux Fig.3-21 et Fig.3-22 avec  $\beta_1 = 1$  et  $\beta_2 = 0$ . Nous pouvons voir que la fonction *Beta-spline* est insensible aux données entachées par les erreurs.

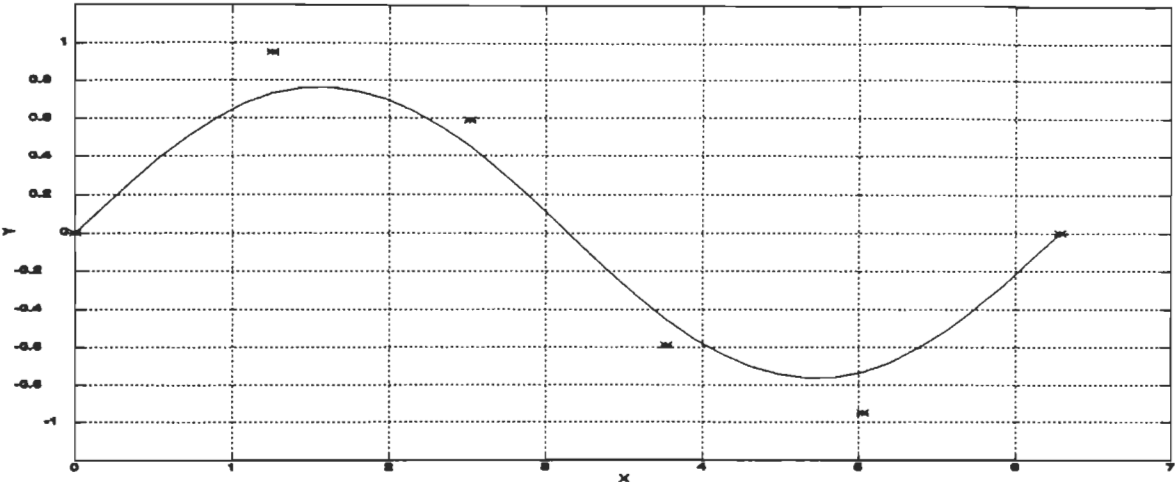


Fig.3-21. Les résultats de la fonction Beta-spline dans le cas de données sans erreurs.

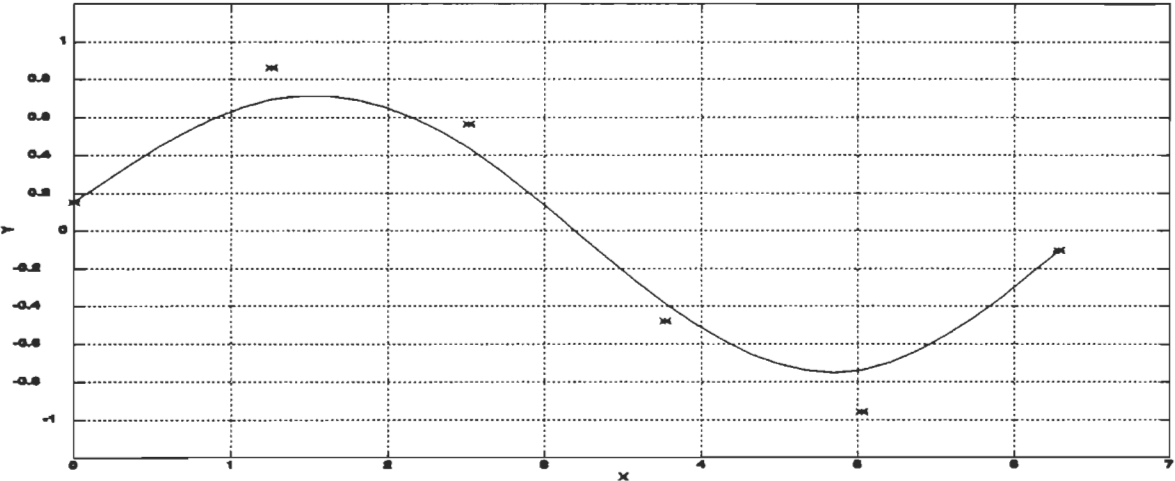


Fig.3-22. Les résultats de la fonction Beta-spline dans le cas de données entachées par les erreurs.

## CHAPITRE 4.

# MÉTHODES D'ÉTALONNAGE PROPOSÉES

L'objectif principal de l'étalonnage est de produire une représentation paramétrique finie du modèle qui peut être utilisée directement pour la reconstitution du mesurande  $x$ , i.e. pour trouver un estimé  $\hat{x}$  de la valeur exacte  $x$ , à partir de la valeur mesurée  $\hat{v}$  de  $v$  et le résultat brut de mesure  $\hat{y}$  correspondant à  $y$ . Le problème de base à résoudre est donc l'approximation des fonctions scalaires des deux variables scalaires en utilisant un ensemble de triplets des valeurs mesurées des  $x$ ,  $v$  et  $y$ . La solution au problème peut être basée sur l'application de la fonction spline à deux variables [3-§12]. Cependant, c'est en utilisant les fonctions spline à une variable que nous obtenons des solutions plus simples et suffisamment exactes; plusieurs options de cette solution seront décrites et étudiées dans le présent travail. Vu que l'exactitude de reconstitution à base des résultats d'étalonnage est une mesure appropriée de la qualité d'étalonnage, les deux opérations seront considérées ensemble.

Pour analyser la première option d'utilisation des fonctions spline à une variable dans l'étalonnage et la reconstitution, nous supposons que les données pour l'étalonnage sont organisées de la façon suivante:

$$D_1^{cal} = \left\{ \hat{x}_i^{cal}, \{ \hat{v}_j^{cal}, \hat{y}_{i,j}^{cal} \mid j=1, \dots, N_v^{cal} \} \mid i=1, \dots, N_x^{cal} \right\} \quad (4-1)$$

où les valeurs de référence  $\hat{x}_i^{cal}$  satisfont les conditions:

$$X_{\min} \cong \hat{x}_1^{cal} < \hat{x}_2^{cal} < \dots < \hat{x}_{N_x^{cal}-1}^{cal} < \hat{x}_{N_x^{cal}}^{cal} \cong X_{\max} \quad (4-2)$$

$$V_{\min} \cong \hat{v}_1^{cal} < \hat{v}_2^{cal} < \dots < \hat{v}_{N_v^{cal}-1}^{cal} < \hat{v}_{N_v^{cal}}^{cal} \cong V_{\max} \quad (4-3)$$

pour  $i=1, \dots, N_x^{cal}$ , et

$$\hat{y}_{i,j}^{cal} \cong G(\hat{x}_i^{cal}, \hat{v}_j^{cal}) \quad (4-4)$$

pour  $i=1, \dots, N_x^{cal}$ , et  $j=1, \dots, N_v^{cal}$ . Ces données sont entachées par les erreurs de mesure qui sont supposées être bornées:

$$| \tilde{y}_{i,j}^{cal} - \hat{y}_{i,j}^{cal} | \leq \bar{\Delta}y^{cal}, | \tilde{x}_i^{cal} - \hat{x}_i^{cal} | \leq \bar{\Delta}x^{cal}, | \tilde{v}_j^{cal} - \hat{v}_j^{cal} | \leq \bar{\Delta}v^{cal} \quad (4-5)$$

où  $\bar{\Delta}y^{cal}$ ,  $\bar{\Delta}x^{cal}$  et  $\bar{\Delta}v^{cal}$  sont les limites d'erreur absolue des  $\tilde{y}^{cal}$ ,  $\tilde{x}^{cal}$  et  $\tilde{v}^{cal}$ , respectivement.

### 4.1. Procédure PCR1

La première procédure d'étalonnage/reconstitution, PCR1, est définie comme suit:

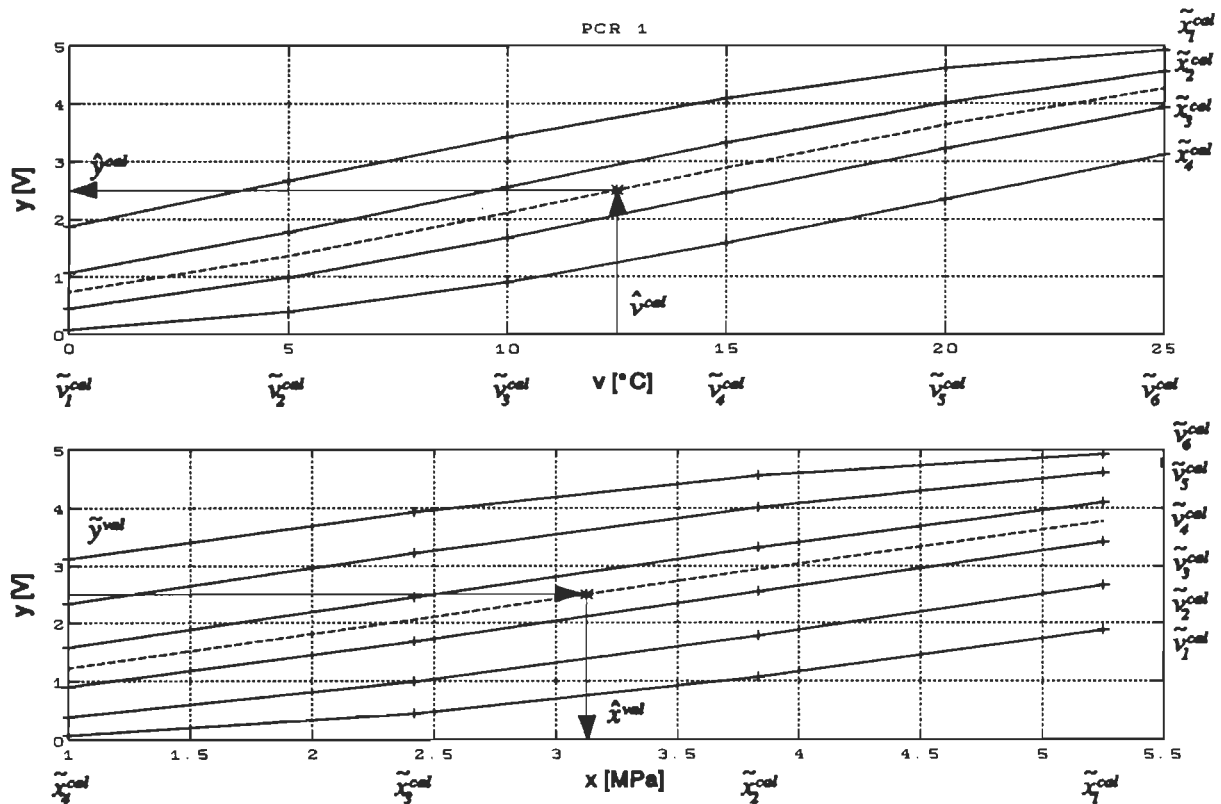


Fig.4-1. Le schéma de la procédure PCR1.

4.1.1. Pour chaque valeur de référence de la mesure  $\tilde{x}_i^{cal}$  ( $i=1, \dots, N_x^{cal}$ ), le vecteur des paramètres  $p_i$  de la fonction spline

$$y = \hat{G}(v; p_i) \quad (4-6)$$

basée sur l'approximation de la séquence des points

$$\left\{ \tilde{v}_{i,j}^{cal}, \tilde{y}_{i,j}^{cal} \mid j=1, \dots, N_v^{cal} \right\} \quad (4-7)$$

est déterminé.



4.1.2. Pour la valeur mesurée  $\hat{\nu}$  de la grandeur d'influence  $\nu$ , la séquence des valeurs

$$\{ \hat{y}_i = \hat{G}(\hat{\nu}; p_i) \mid i=1, \dots, N_x^{cal} \} \quad (4-8)$$

est calculée et utilisée pour la détermination du vecteur des paramètres  $p(\hat{\nu})$  d'autres fonctions spline

$$y = \hat{G}(x; p(\hat{\nu})) \quad (4-9)$$

basée sur l'approximation de la séquence des points

$$\{ \hat{x}_i^{cal}, \hat{y}_i^{cal} \mid i=1, \dots, N_x^{cal} \} \quad (4-10)$$

4.1.3. L'estimé du mesurande, correspondant à la valeur mesurée  $\hat{\nu}$  et au résultat brut de mesure  $\hat{y}$ , est calculé selon la formule:

$$\hat{x} = \arg_x \{ \hat{G}(x; p(\hat{\nu})) = \hat{y} \} \quad (4-11)$$

Le schéma de la procédure PCR1 est montré à la Fig.4-1.

## 4.2. Procédure PCR2

La deuxième procédure d'étalonnage/reconstitution, PCR2, est basée sur la supposition que les données pour l'étalonnage sont organisées comme suit:

$$D_2^{cal} = \{ \nu_i^{cal}, \{ \hat{x}_{i,j}^{cal}, \hat{y}_{i,j}^{cal} \mid j=1, \dots, N_x^{cal} \} \mid i=1, \dots, N_\nu^{cal} \} \quad (4-12)$$

Elles satisfont les conditions dans les equations (4-2) ~ (4-5), avec la condition additionnelle:

$$\hat{y}_{i,j}^{cal} \cong \hat{G}(\hat{x}_{i,j}^{cal}, \nu_i^{cal}) \quad (4-13)$$

Cette procédure est définie comme suit:

4.2.1. Pour chaque valeur de référence de la grandeur d'influence  $\nu_i^{cal}$  ( $i=1, \dots, N_\nu^{cal}$ ), le vecteur des paramètres  $p_i$  de la fonction spline

$$y = \hat{G}(\nu; p_i) \quad (4-14)$$

basée sur l'approximation de la séquence des points

$$\{ \hat{x}_{i,j}^{cal}, \hat{y}_{i,j}^{cal} \mid j=1, \dots, N_x^{cal} \} \quad (4-15)$$

est déterminé.

4.2.2. Pour le résultat brut de la mesure  $\hat{y}$ , la séquence des valeurs

$$\{ \hat{x}_i = \arg_x \{ \hat{G}(x; p_i) = \hat{y} \} \mid i=1, \dots, N_\nu^{cal} \} \quad (4-16)$$

est calculée et utilisée pour la détermination d'autres fonctions spline

$$x = \hat{G}(\nu; p(\hat{y})) \quad (4-17)$$

basée sur l'approximation de la séquence des points

$$\left\{ \hat{v}_i^{cal}, \hat{x}_i^{cal} \mid i=1, \dots, N_v^{cal} \right\} \quad (4-18)$$

4.2.3. L'estimé du mesurande, correspondant à la valeur mesurée  $\hat{v}$  et au résultat brut de mesure  $\hat{y}$ , est calculé comme suit:

$$\hat{x} = \hat{G}(\hat{v}; \mathbf{p}(\hat{y})) \quad (4-19)$$

### 4.3. Procédure PCR3

La troisième procédure, PCR3, est définie pour l'ensemble des données d'étalonnage  $D_1^{cal}$  comme suit:

4.3.1. Pour chaque valeur référence du mesurande  $\hat{x}_i^{cal}$  ( $i=1, \dots, N_x^{cal}$ ), le vecteur des paramètres  $\mathbf{p}_i$  de la fonction spline

$$y = \hat{G}(v; \mathbf{p}_i) \quad (4-20)$$

basée sur l'approximation de la séquence des points

$$\left\{ \hat{v}_{i,j}^{cal}, \hat{y}_{i,j}^{cal} \mid j=1, \dots, N_v^{cal} \right\} \quad (4-21)$$

est déterminé.

4.3.2. Pour chaque valeur mesurée  $\hat{v}$  de la grandeur d'influence  $v$ , la séquence des valeurs

$$\left\{ \hat{y}_i = \hat{G}(\hat{v}; \mathbf{p}_i) \mid i=1, \dots, N_x^{cal} \right\} \quad (4-22)$$

est calculée et utilisée pour la détermination du vecteur des paramètres  $\mathbf{p}(\hat{v})$  d'autres fonctions spline

$$x = \hat{G}(y; \mathbf{p}(\hat{v})) \quad (4-23)$$

basée sur l'approximation de la séquence des points

$$\left\{ \hat{y}_i, \hat{x}_i^{cal} \mid i=1, \dots, N_x^{cal} \right\} \quad (4-24)$$

4.3.3. L'estimé du mesurande est calculé comme suit:

$$\hat{x} = \hat{G}(\hat{y}; \mathbf{p}(\hat{v})) \quad (4-25)$$

Le schéma de la procédure PCR3 est montré à la Fig.4-2.

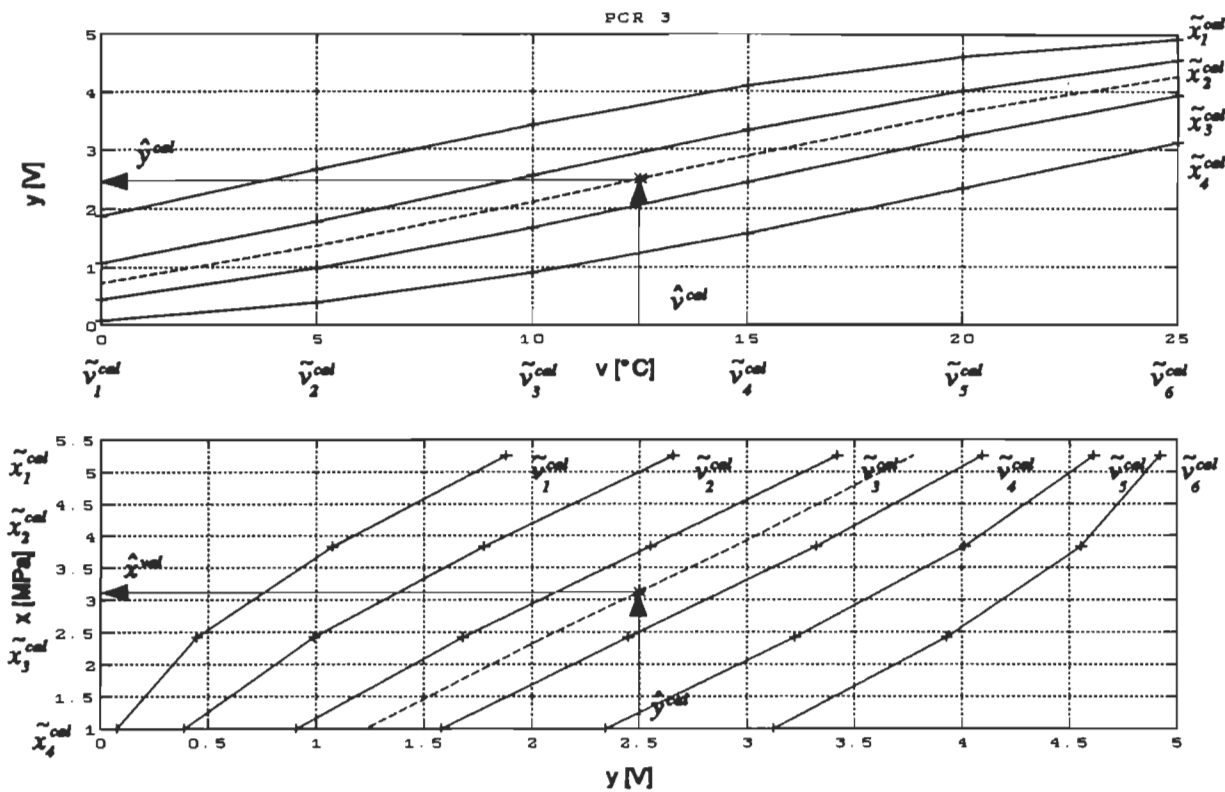


Fig.4-2. Le schéma de la procédure PCR3.

#### 4.4. Procédure PCR4

La dernière procédure d'étalonnage/reconstitution, PCR4, est définie pour la série de données d'étalonnage  $D_2^{cal}$  comme la suivante:

4.4.1. Pour chaque valeur de référence de la grandeur d'influence  $\tilde{v}_i^{cal}$  ( $i = 1, \dots, N_v^{cal}$ ), le vecteur des paramètres  $p_i$  de la fonction spline

$$x = \hat{G}(y; p_i) \tag{4-26}$$

basée sur l'approximation de la séquence des points

$$\left\{ \tilde{y}_{i,j}^{cal}, \tilde{x}_{i,j}^{cal} \mid j = 1, \dots, N_x^{cal} \right\} \tag{4-27}$$

est déterminé.

4.4.2. Pour le résultat brut de mesure  $y$ , la séquence des valeurs

$$\left\{ x_i = \hat{G}(y; p_i) \mid i = 1, \dots, N_v^{cal} \right\} \quad (4-28)$$

est calculée et utilisée pour la détermination de la fonction spline

$$x = \hat{G}(v; p(y)) \quad (4-29)$$

4.4.3. L'estimé du mesurande, correspondant à la valeur mesurée  $\hat{v}$  et au résultat brut de mesure  $y$ , est calculé comme suit:

$$\hat{x} = \hat{G}(\hat{v}; p(y)) \quad (4-30)$$

## **CHAPITRE 5.**

# **MÉTHODOLOGIE D'ÉTUDE**

### **5.1. Synthèse et acquisition des données de mesure**

Tous les travaux sont faits sur des systèmes de mesure qui donnent des signaux expérimentaux. Les signaux expérimentaux sont obtenus dans des conditions de laboratoire et perturbés par l'erreur de mesure. Si nous ne considérons pas l'erreur en établissant un modèle mathématique avec une méthode d'identification de système, telle que la méthode de moindres carrés, le système décrit par le modèle mathématique est le système idéal. Nous pouvons déterminer la variation de l'erreur durant l'expérimentation et l'ajouter au système idéal. De cette manière, nous obtenons le système idéal avec les erreurs ou le système entaché par les erreurs. En général, nous pouvons utiliser le système entaché par les erreurs pour remplacer le système réel étant donné qu'ils ont presque les mêmes caractéristiques dans la perturbation et peuvent être utilisés pour étudier la qualité des algorithmes et l'optimisation des paramètres.

Une méthodologie d'étalonnage basée sur les fonctions spline est développée et évaluée avec des données synthétiques qui représentent trois systèmes de mesure qualitativement différents.

Afin de démontrer l'applicabilité des fonctions spline aux systèmes de mesure comme méthode d'étalonnage statique et de reconstitution de mesurandes, nous avons utilisé des systèmes de mesure précis. Dans ce chapitre nous allons introduire les bases du fonctionnement de ces systèmes de mesure de la manière suivante. Dans une première partie nous allons présenter le principe de fonctionnement et la caractéristique d'entrée-sortie d'un capteur. La deuxième partie décrit le système de mesure de mesurande dans son ensemble. Nous soulèverons aussi un problème important lié au fonctionnement d'un capteur, il présente le désagrément d'être très sensible aux variations des grandeurs d'influence et cet effet doit donc être pris en considération par la méthode d'étalonnage et de reconstitution de mesurandes.

Pour tester la validité de la méthode d'étalonnage et de reconstitution de mesurandes, deux types de données sont à générer:

- les données pour l'étalonnage, notées avec l'exposant *cal*;

- les données pour l'évaluation, notées avec l'exposant *val*.

Les données d'étalonnage sont générées de façon suivante. Nous supposons que  $N_v^{cal}$  valeurs de la grandeur d'influence  $v_i^{cal}$  ( $i = 1, \dots, N_v^{cal}$ ):

$$V_{\min} \cong v_1^{cal} < v_2^{cal} < \dots < v_{N_v^{cal}}^{cal} \cong V_{\max} \quad (5-1)$$

et  $N_x^{cal}$  valeurs du mesurande  $x_j^{cal}$  ( $j = 1, \dots, N_x^{cal}$ ) telles que:

$$X_{\min} \cong x_1^{cal} < x_2^{cal} < \dots < x_{N_x^{cal}}^{cal} \cong X_{\max} \quad (5-2)$$

et  $N_v^{cal} \times N_x^{cal}$  valeurs du résultat brut de mesure  $y_{ij}^{cal}$ :

$$\begin{bmatrix} y_{1,1}^{cal} & y_{1,2}^{cal} & \dots & y_{1,N_x^{cal}}^{cal} \\ y_{2,1}^{cal} & y_{2,2}^{cal} & \dots & y_{2,N_x^{cal}}^{cal} \\ \dots & \dots & \dots & \dots \\ y_{N_v^{cal},1}^{cal} & y_{N_v^{cal},2}^{cal} & \dots & y_{N_v^{cal},N_x^{cal}}^{cal} \end{bmatrix} \quad (5-3)$$

Afin de tenir compte des erreurs de mesure lors de l'étalonnage et des erreurs de mesure, les données exactes générées précédemment sont entachées additivement par les erreurs à l'aide de nombres pseudo-aléatoires  $\{\Delta v_i\}$ ,  $\{\Delta x_j\}$  et  $\{\Delta y_{ij}\}$  uniformément distribués autour de zéro:

$$\tilde{v}_i^{cal} = \hat{v}_i^{cal} + \Delta v_i \quad |\Delta v_i| \leq \bar{\Delta} v^{cal} \quad (5-4)$$

$$\tilde{x}_j^{cal} = \hat{x}_j^{cal} + \Delta x_j \quad |\Delta x_j| \leq \bar{\Delta} x^{cal} \quad (5-5)$$

$$\tilde{y}_{ij}^{cal} = \hat{y}_{ij}^{cal} + \Delta y_{ij} \quad |\Delta y_{ij}| \leq \bar{\Delta} y^{cal} \quad (5-6)$$

où  $\bar{\Delta} v^{cal}$ ,  $\bar{\Delta} x^{cal}$ , et  $\bar{\Delta} y^{cal}$  sont les limites d'erreur absolue de la grandeur d'influence  $\hat{v}^{cal}$ , du mesurande  $\hat{x}^{cal}$  et du résultat brut de mesure  $\hat{y}^{cal}$ .

Il y a  $R+1$  groupes de données pour l'étalonnage:

$$\hat{D}^{cal} = \left\{ \hat{x}_j^{cal}, \{ \hat{v}_i^{cal}, \hat{y}_{ij}^{cal} \mid i=1, \dots, N_v^{cal} \} \mid j=1, \dots, N_x^{cal} \right\} \quad (5-7)$$

$$\tilde{D}_r^{cal} = \left\{ \tilde{x}_{j,r}^{cal}, \{ \tilde{v}_{i,r}^{cal}, \tilde{y}_{ij,r}^{cal} \mid i=1, \dots, N_v^{cal} \} \mid j=1, \dots, N_x^{cal} \right\} \quad \text{pour } r=1, \dots, R \quad (5-8)$$

où,

$$\tilde{y}_{ij}^{cal} \cong G(\tilde{x}_j^{cal}, \tilde{v}_i^{cal}) \quad (5-9)$$

pour  $j=1, \dots, N_x^{cal}$  et  $i=1, \dots, N_v^{cal}$ .

### 5.1.1. Données synthétiques: un capteur ultrasonique pour mesurer la concentration d'alcool dans les solutions d'eau - CAS1

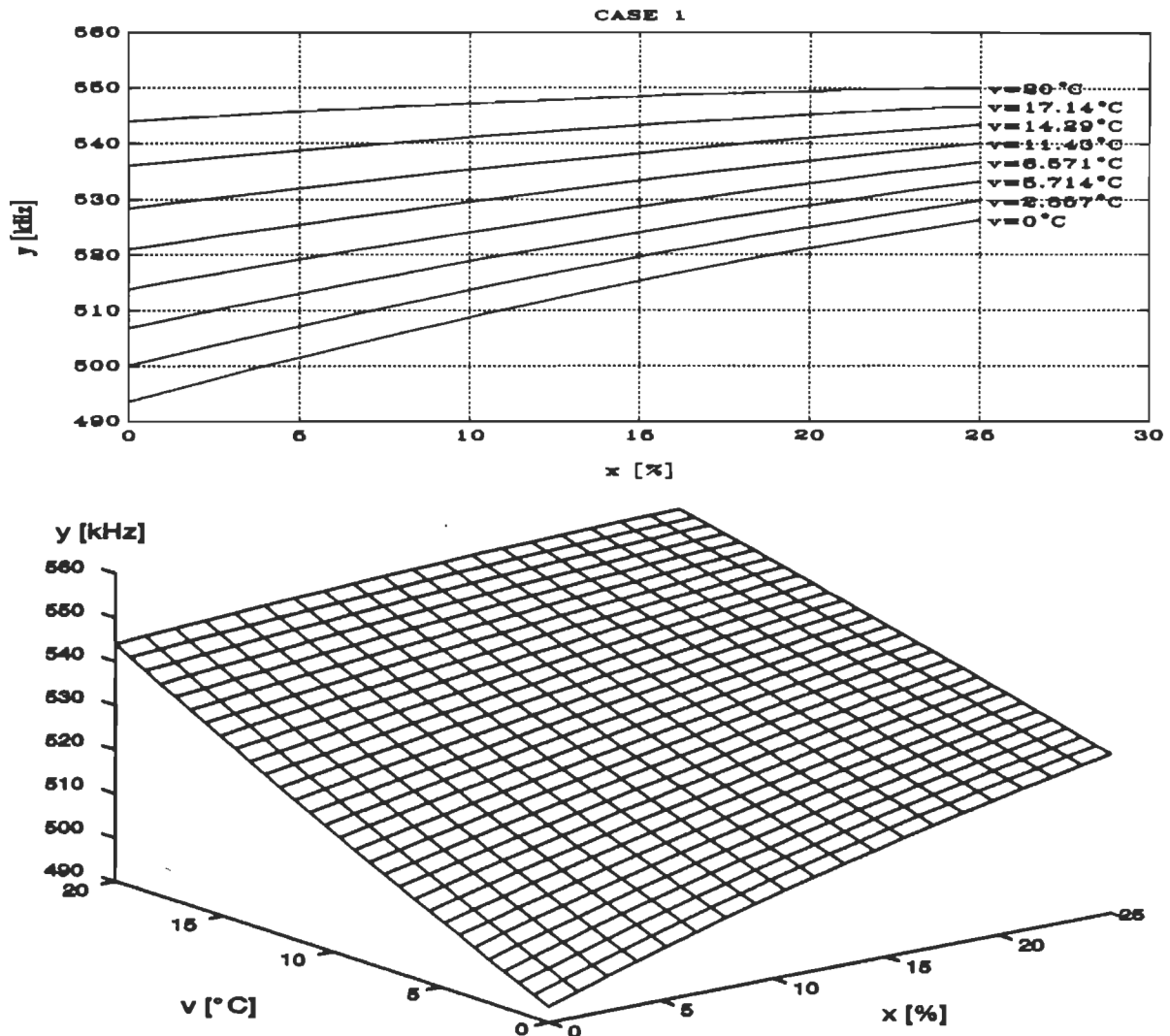


Fig.5-1. La caractéristique statique du capteur ultrasonique pour mesurer la concentration d'alcool dans les solutions d'eau (CAS 1).

Il s'agit d'un système de mesure contenant un capteur ultrasonique pour mesurer la concentration d'alcool dans les solutions d'eau. Sa fréquence résonnante ( $y$ ) dépend de la concentration ( $x$ ) et de la température ( $v$ ) - [1], [6]. Les caractéristiques exemplaires du capteur ultrasonique (CAS1) sont légèrement non-linéaires et monotones par rapport aux deux variables:  $x$  et  $v$ . Les caractéristiques exemplaires  $y = G(x; v)$  du capteur ultrasonique (CAS1) sont montrées à la Fig.5-1. Les données synthétiques sont générées selon la formule suivante:

$$y = k_0(x) + k_1(x)v + k_2(x)v^2 \quad (5-10)$$

avec

$$k_0(x) = 493.6 + 1.66x - 0.014x^2$$

$$k_1(x) = 2.24 - 0.064x + 0.00095x^2$$

$$k_2(x) = 0.014 - 0.00005x - 0.000024x^2$$

pour  $x \in [0, 25]$  et  $v \in [0, 20]$ .

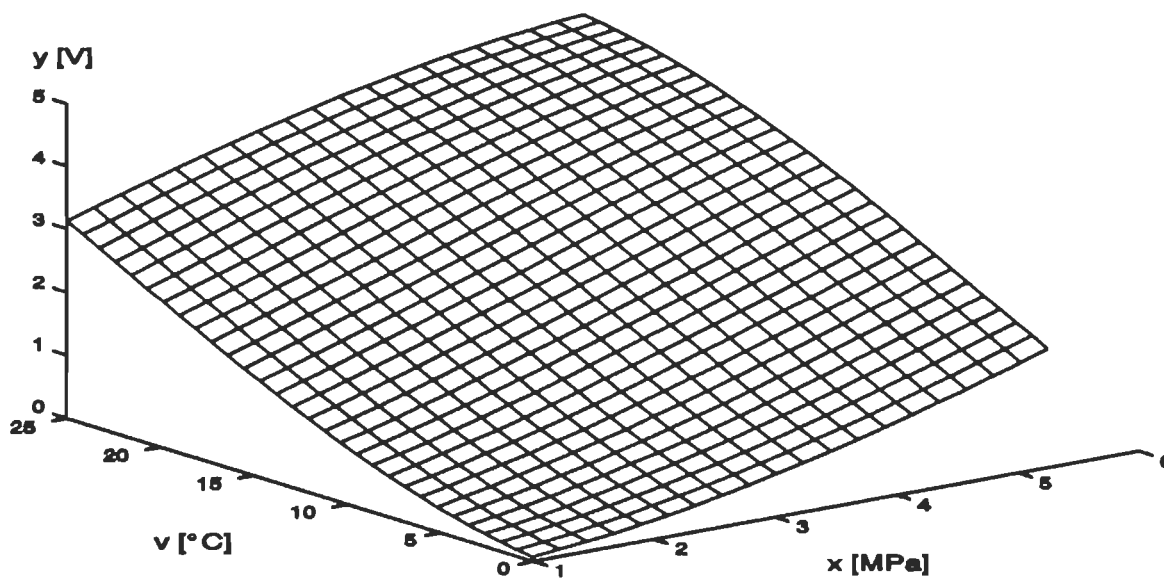
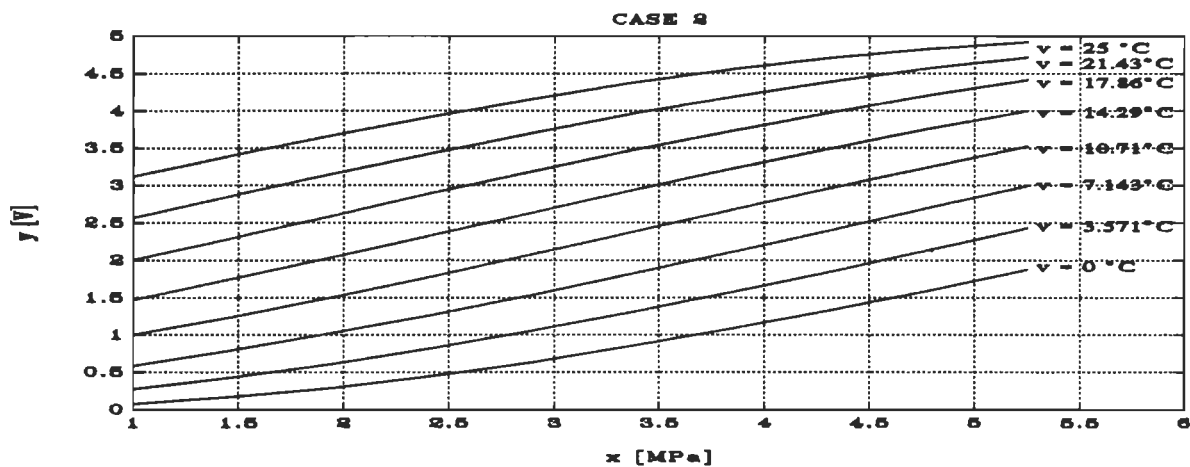


Fig.5-2. La caractéristique statique du capteur à fibre-optique pour mesurer la haute pression (CAS 2).



### 5.1.2. Données synthétiques: un capteur à fibre-optique pour mesurer la haute pression - CAS2

Le capteur à fibre optique considéré, capable de mesurer des pressions allant jusqu'à 100 MPa, peut être vu comme une fibre interférométrique, possédant deux modes de polarisation différents dont les écarts de phase produisent différentes réponses à une pression extérieure.

La caractéristique d'entrée/sortie du capteur est une fonction périodique de la pression. La variation du signal de sortie du capteur est une fonction quasi-sinusoïdale de la pression dont la période est fonction de la longueur  $L$  de la fibre ( $U$  est la tension de sortie du capteur et  $p$  est la pression d'entrée). Il est très important de noter qu'une telle caractéristique d'entrée/sortie est obtenue à température fixe, car le capteur est fortement influencé par celle-ci.

La tension de sortie ( $y$ ) d'un système de mesure contenant un capteur fibre-optique pour mesurer la haute pression dépend de la pression ( $x$ ) et de la température ( $v$ ) - [2], [7]. Les caractéristiques  $y = G(x; v)$  du capteur à fibre-optique (CAS2) sont montrées à la Fig.5-2. Les données synthétiques sont générées selon la formule suivante [7]:

$$y = 5.0 \cos^2[(0.04x + 0.01v + 0.5)\pi] \quad (5-11)$$

pour  $x \in [1, 5.25]$  et  $v \in [0, 25]$ , correspondant aux fragments monotones.

### 5.1.3. Données synthétiques: un capteur thermique à enzyme pour mesurer la concentration d'urée - CAS3

La tension de sortie ( $y$ ) d'un système de mesure contenant un capteur thermique à enzyme pour mesurer la concentration d'urée dépend de la concentration d'urée ( $x$ ) et de pH ( $v$ ) - [8]. Les caractéristiques  $y = G(x; v)$  simulantes du comportement du capteur thermique à enzyme (CAS3) sont montrées à la Fig.5-3. Les données synthétiques sont générées selon la formule suivante:

$$y = (622.67 - 62.104v) \{1 - \exp[-(27.77 - 2.7626v)(x - 3.6216x^2)]\} \quad (5-12)$$

pour  $x \in [0, 0.1]$  et  $v \in [7, 8.5]$ .

### 5.1.4. Données réelles: un capteur à fibre-optique pour mesurer les hautes pressions - CAS2

Les données réelles du CAS2 ont les champs de la grandeur d'influence  $v$  et du mesurande  $x$  suivants:

$$V_{\min} = 0 \text{ [}^\circ\text{C]}, \quad V_{\max} = 35 \text{ [}^\circ\text{C]} \quad (5-13)$$

$$X_{\min} = 0 \text{ [Bar]}, \quad X_{\max} = 240 \text{ [Bar]} \quad (5-14)$$

Les nombres de données réelles pour l'étalonnage sont:

$$N_v^{\text{cal}} = 7 \quad (5-15)$$

$$N_x^{\text{cal}} = 9 \quad (5-16)$$

Les nombres de données réelles pour l'évaluation de la méthode de reconstitution de mesurandes

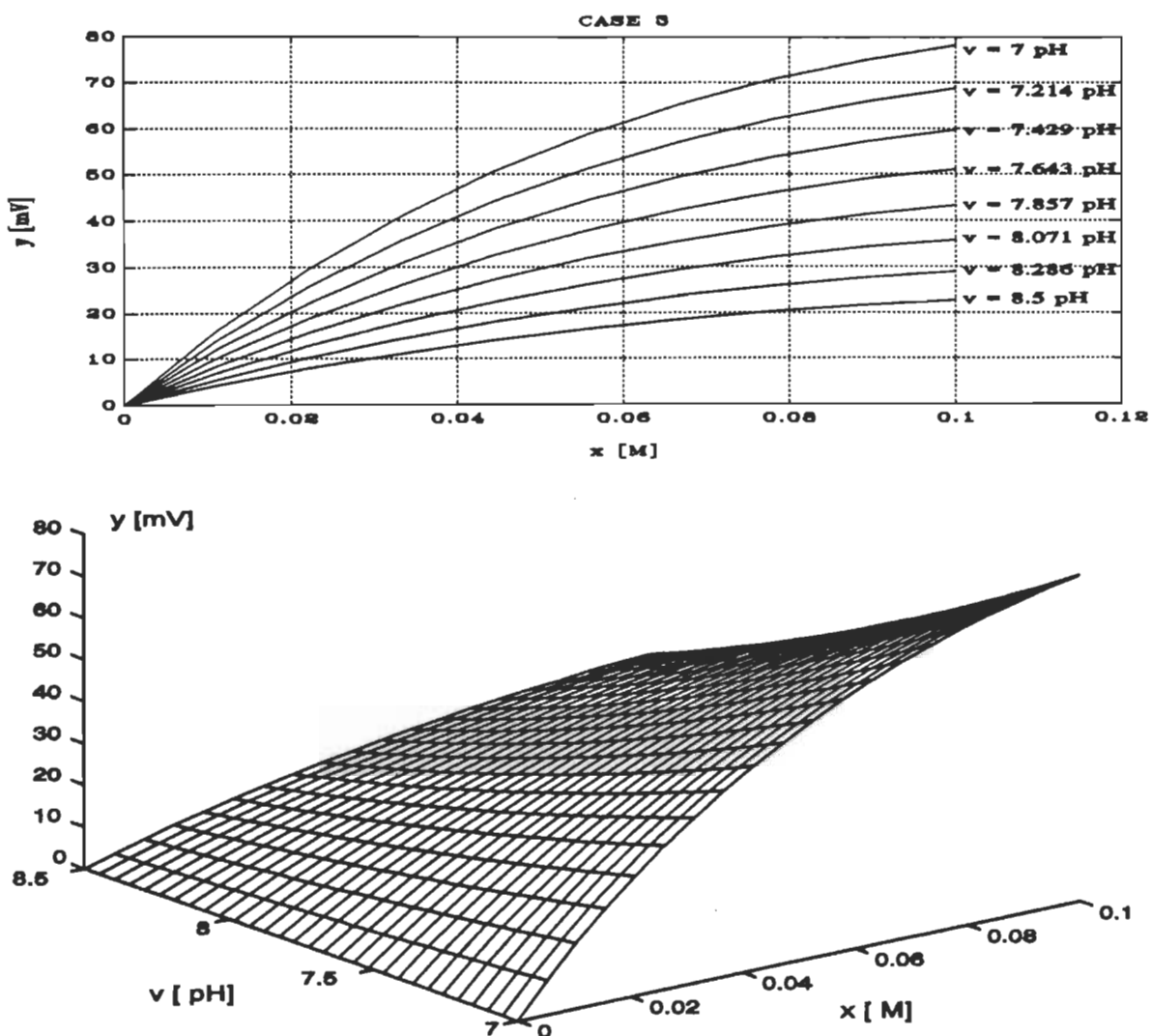


Fig.5-3. La caractéristique statique du capteur thermique à enzyme pour mesurer la concentration d'urée (CAS 3).

sont:

$$N_v^{val} = 13 \tag{5-17}$$

$$N_x^{val} = 17 \tag{5-18}$$

Les données réelles du CAS2 sont montrées sur la Fig.5-4. Les valeurs de la température sont 0, 5.5, 11, 16.5, 22, 27.5, 33 [°C]. Les valeurs de la pression sont 0, 30, 60, 90, 120, 150, 180, 210, 240 [Bar].

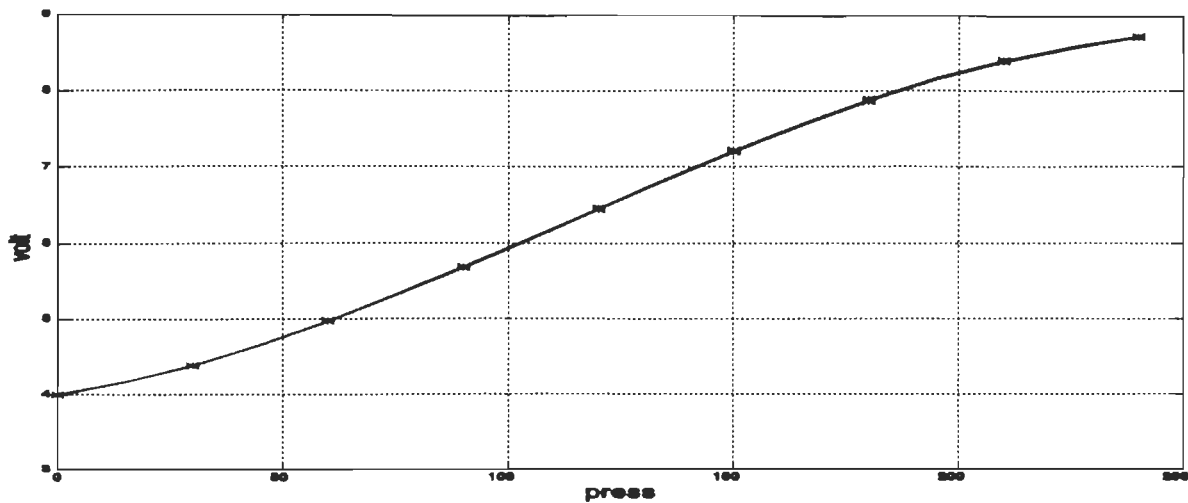


Fig.5-4. Les données réelles du CAS2.

## 5.2. Évaluation de la qualité de l'étalonnage

### 5.2.1. Synthèse et acquisition des données pour l'évaluation

Les données pour l'évaluation sont :

$$\dot{D}^{val} = \left\{ \dot{x}_j^{val}, \{ \dot{v}_i^{val}, y_{i,j}^{val} \mid i=1, \dots, N_v^{val} \} \mid j=1, \dots, N_x^{cal} \right\} \quad (5-19)$$

où les valeurs de mesurande et de la grandeur d'influence sont choisies approximativement égales à:

$$x_j^{val} = \left\{ \dot{x}_i^{cal}, \frac{\dot{x}_{i+1}^{cal} + \dot{x}_i^{cal}}{2} \mid i=2, \dots, N_x^{cal}-2 \right\} \mid j=1, \dots, N_x^{val} \quad (5-20)$$

et

$$v_j^{val} = \left\{ \dot{v}_i^{cal}, \frac{\dot{v}_{i+1}^{cal} + \dot{v}_i^{cal}}{2} \mid i=2, \dots, N_v^{cal}-2 \right\} \mid j=1, \dots, N_v^{val} \quad (5-21)$$

$$y_{i,j}^{val} = G(\dot{x}_j^{val}, \dot{v}_i^{val}) \quad (5-22)$$

où

$$V_{\min} < \dot{v}_1^{val} < \dot{v}_2^{val} < \dots < \dot{v}_{N_v^{val}}^{val} < V_{\max} \quad (5-23)$$

$$X_{\min} < \dot{x}_1^{val} < \dot{x}_2^{val} < \dots < \dot{x}_{N_x^{val}}^{val} < X_{\max} \quad (5-24)$$

$$\begin{bmatrix} y_{1,1}^{val} & y_{1,2}^{val} & \dots & y_{1,N_r^{val}}^{val} \\ y_{2,1}^{val} & y_{2,2}^{val} & \dots & y_{2,N_r^{val}}^{val} \\ \dots & \dots & \dots & \dots \\ y_{N_r^{val},1}^{val} & y_{N_r^{val},2}^{val} & \dots & y_{N_r^{val},N_r^{val}}^{val} \end{bmatrix} \quad (5-25)$$

La Fig.5-5 montre le choix des données d'étalonnage et d'évaluation dans (5-20) et (5-21).

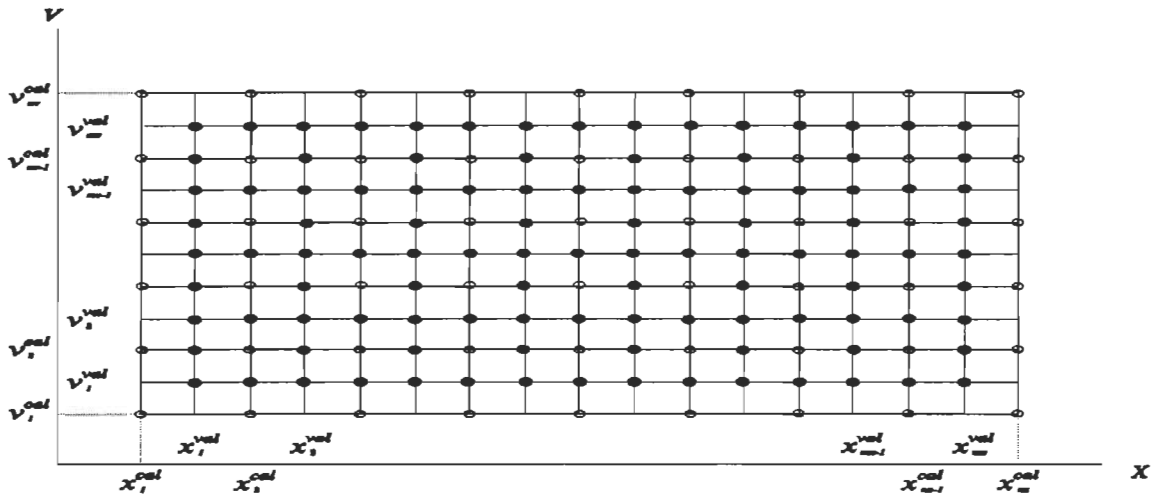


Fig.5-5. Les données d'étalonnage (○) et les données d'évaluation (● et ○).

Le nombre de données d'évaluation doit être plus grand que le nombre de données d'étalonnage. Les données d'évaluation sont considérées comme sans erreurs.

La procédure de l'évaluation est exprimée comme suit:

1. La procédure  $Par(\cdot)$  est utilisée pour l'étalonnage:

$$\hat{p} = Par(\hat{x}^{cal}, \hat{y}^{cal}, \hat{v}^{cal}) \text{ ou } \hat{p}_r = Par(\hat{x}_r^{cal}, \hat{y}_r^{cal}, \hat{v}_r^{cal}) \quad (5-26)$$

À partir de cette procédure, nous pouvons obtenir les paramètres du modèle mathématique: la fonction spline.

2. La procédure  $Val(\cdot)$  est utilisée pour l'évaluation:

$$\hat{x}_{i,j}^{val} = Val(\hat{y}_{i,j}^{val}, \hat{v}_j^{val}; \hat{p}) \quad \text{pour } i = 1, \dots, N_v^{val} \text{ et } j = 1, \dots, N_x^{val} \quad (5-27)$$

où, les paramètres  $\hat{p}$  viennent de la procédure d'étalonnage décrite par (5-26).

### 5.2.2. Critères de l'évaluation

La procédure d'évaluation est de comparer les valeurs provenant de la mesure et les valeurs provenant de la reconstitution. L'erreur de reconstitution est utilisée pour évaluer la qualité d'étalonnage.

L'évaluation des performances des fonctions spline se fait à l'aide de l'erreur absolue maximale de reconstitution:

$$\Delta x = \sup \left\{ | \hat{x}_i^{val} - \tilde{x}_i^{val} | \mid i = 1, \dots, N_x^{val} \right\} \quad (5-28)$$

où  $\tilde{x}_i^{val}$  est le mesurande désiré correspondant à  $y_i^{val}$  et  $\hat{x}_i^{val}$  est l'estimé du mesurande.

Vu la nature aléatoire de l'erreur sur les données d'étalonnage, il faut répéter la simulation  $R$  fois. L'erreur finale absolue maximale de reconstitution est:

$$\bar{\Delta x} = \sup \left\{ \Delta x^{(r)} \mid r = 1, \dots, R \right\} \quad (5-29)$$

où  $\Delta x^{(r)}$  est le résultat de la  $r^{ième}$  simulation qui est défini par (5-28).

La comparaison est effectuée selon la méthodologie suivante: deux types d'erreur: l'erreur quadratique moyenne relative de l'estimé  $\delta_2$  et l'erreur maximale relative de l'estimé  $\delta_\infty$ , et la valeur de *FLOPS* effectuées sont enregistrées pour l'évaluation de la rapidité de calcul et de l'exactitude des fonctions spline.

*FLOPS* compte les opérations de virgule flottante et retourne le nombre cumulatif d'opérations. Il n'est pas faisable de compter absolument toutes les opérations de virgule flottante, mais la majorité sont comptées. L'addition et la soustraction sont comptées comme un flop si réel et deux flop si complexe. La multiplication et la division sont comptées comme un flop chacune si le résultat est réel et six flops s'il ne l'est pas. Les fonctions élémentaires sont comptées un si réel et plus si complexe.

Quant aux données pour l'étalonnage sans erreurs, les formules d'erreur suivantes sont utilisées:

- l'erreur quadratique moyenne relative de l'estimé  $\hat{x}^{val}$ :

$$\delta_2[\hat{x}^{val}] = \frac{\| \Delta x^{val} \|_2}{\| \hat{x}^{val} \|_2} = \frac{\| \hat{x}^{val} - \tilde{x}^{val} \|_2}{\| \hat{x}^{val} \|_2} = \sqrt{\frac{\sum_{i=1}^{N_x^{val}} (\hat{x}_i^{val} - \tilde{x}_i^{val})^2}{\sum_{i=1}^{N_x^{val}} (\hat{x}_i^{val})^2}} \quad (5-30)$$

- l'erreur maximale relative de l'estimé  $\hat{x}^{val}$ :

$$\delta_\infty[\hat{x}^{val}] = \frac{\| \Delta x^{val} \|_\infty}{\| \hat{x}^{val} \|_\infty} = \frac{\| \hat{x}^{val} - \tilde{x}^{val} \|_\infty}{\| \hat{x}^{val} \|_\infty} = \frac{\sup_i \left\{ | \hat{x}_i^{val} - \tilde{x}_i^{val} | \mid i = 1, \dots, N_x^{val} \right\}}{\sup_i \left\{ | \hat{x}_i^{val} | \mid i = 1, \dots, N_x^{val} \right\}} \quad (5-31)$$

Les valeurs de  $\delta_2$  et  $\delta_\infty$  et la valeur de *FLOPS* dans le cas de données sans erreur sont présentées au Tableau 6-5.

Si l'étalonnage est effectué à l'aide de données entachées par les erreurs, alors il est répété  $R$  fois pour  $\tilde{D}_r^{cal}$  et pour chaque résultat de simulation, les erreurs sont calculées:

$$\delta_{2,r} = \delta_2[\hat{x}_r^{val}] \quad (5-32)$$

et

$$\delta_{\infty,r} = \delta_\infty[\hat{x}_r^{val}] \quad (5-33)$$

Ensuite:

$$\delta_2 = \sqrt{\frac{1}{R} \sum_{r=1}^R \delta_{2,r}^2} \quad (5-34)$$

et

$$\delta_\infty = \sup \{ \delta_{\infty,r} \mid r=1, \dots, R \} \quad (5-35)$$

où  $\hat{x}_r^{val}$  correspond la valeur donnée par l'estimé des paramètres  $\hat{p}$  obtenu sur la base des données  $\tilde{D}_r^{cal}$ .

Les valeurs des  $\delta_2$  et  $\delta_\infty$  et la valeur de *FLOPS* dans le cas de données entachées par les erreurs sont présentées au Tableau 6-6. Les valeurs de *FLOPS* sont obtenues pour une seule réalisation d'erreur.

### 5.2.3. Traitement postérieur des résultats d'évaluation

Après les procédures d'étalonnage et de reconstitution, nous avons obtenu plusieurs courbes d'erreur qui dépendent du nombre de points d'échantillon et de l'amplitude de l'erreur de mesure. Notre objectif est d'obtenir la valeur minimale du produit  $N_x^{cal} \times N_v^{cal}$ . La Fig.5-6 montre les courbes d'erreur typiques.

Dans la figure, nous pouvons trouver un point sur les courbes  $\bar{\Delta}x_{\min} = \inf \{ \bar{\Delta}x_i \}$ , la valeur  $\bar{\Delta}x_i$  est définie comme l'erreur de l'estimé du mesurande dans (5-29). Cette valeur correspond aux valeurs de  $N_x^{cal}$  et  $N_v^{cal}$ . Pour exprimer la caractéristique de la relation entre l'exactitude et les données pour l'étalonnage, une valeur finale d'erreur absolue maximale est la suivante:

$$\Delta x^* = 110 \% \bar{\Delta}x_{\min} \quad (5-36)$$

où,  $\Delta x^*$  est le résultat final d'erreur absolue maximale qui correspond à la valeur de  $(N_x^{cal} \times N_v^{cal})_{\min}$ , il est montré aux Tableaux 6-2 ~ 6-4. Dans les Tableaux 6-2, 6-3 et 6-4, la quantité dans les parenthèses est la valeur  $(N_x^{cal} \times N_v^{cal})_{\min}$  qui est obtenue à la position minimale d'erreur absolue maximale et correspond au produit minimal du  $N_x^{cal} \times N_v^{cal}$  sur la même valeur

d'erreur  $\Delta x^*$ .

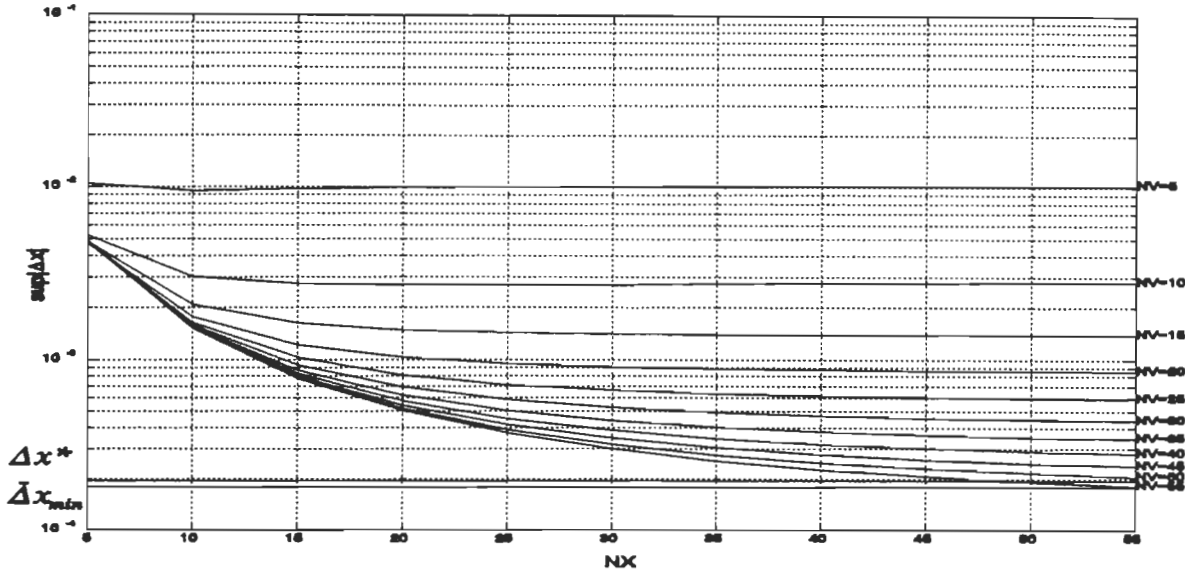


Fig.5-6. Les courbes d'erreur typiques.

## **CHAPITRE 6.**

# **RÉSULTATS NUMÉRIQUES DE L'ÉTUDE**

Afin de vérifier l'applicabilité des fonctions spline définies au *chapitre 3* à l'étalonnage et à la reconstitution de mesurandes dans les systèmes de mesure, nous allons maintenant présenter plusieurs résultats obtenus pour les systèmes de mesure.

Nous avons défini de façon précise la tâche à effectuer par la fonction spline, c'est-à-dire l'étalonnage des systèmes de mesure et la reconstitution des mesurandes, nous avons vu ces opérations dans le domaine des fonctions spline. À la suite de cela, nous avons défini les signaux à traiter, soit les sorties des capteurs (la tension, la fréquence), les mesurandes à reconstituer (la concentration, la pression) et les influences (la température, la valeur de pH).

Deux étapes d'évaluation seront réalisées, dans un premier temps avec des données synthétiques à l'influence variable et pour finir avec des données réelles. Afin d'évaluer les différents résultats obtenus avec les méthodes, une comparaison est faite avec les résultats obtenus avec les méthodes d'étalonnage et de reconstitution basées sur des approximations par les fonctions spline différentes.

Dans ce chapitre, la méthode d'étalonnage et de reconstitution de mesurandes utilisant les fonctions spline a été testée. La méthode a été appliquée dans différents cas afin d'évaluer ses qualités, d'une part avec des données synthétiques et d'autre part avec des données réelles. Le principal avantage d'utiliser des données synthétiques est la flexibilité qu'elles apportent en termes de quantité et de variété; il est possible de générer les données qui nous intéressent sans être limités par les contraintes imposées par la mesure dans le cas des données réelles.

Les fonctions spline mises au point ont été testées pour le cas relativement simple où les données à traiter sont sous l'influence variable. Cette partie des tests nous a permis de mieux comprendre le comportement des fonctions spline en fonction de différents paramètres de l'algorithme, de la structure des fonctions spline et de l'ensemble des données d'optimisation. Nous nous permettons d'ajouter les contraintes imposées par un facteur d'influence agissant directement sur les données à traiter par les fonctions spline. Dans le cas des systèmes de mesure, il faut prendre en considération l'effet de l'influence. En exploitant la capacité d'estimation des fonctions spline, nous allons montrer qu'il est possible d'étalonner les systèmes de mesure avec des informations



limitées sur l'influence. La capacité d'optimisation de l'évaluation des données permet donc de minimiser l'effet du grandeur d'influence contenu dans cette évaluation et la capacité d'estimation permet de reconstituer des mesurandes dans le cas de données d'entrée entachées par les erreurs. Les résultats obtenus ont été comparés avec la méthode basée sur des fonctions spline différentes, les qualités de reconstitution de mesurandes sont comparables.

Le dernier groupe de traitement de données a permis de tester les performances des fonctions spline dans le cas des données réelles. Malgré quelques difficultés dues en grande partie à la quantité limitée de données disponibles, nous avons montré que notre méthode s'applique dans le cas de données réelles. Le système de mesure a été étalonné et les mesurandes sont reconstitués avec une exactitude satisfaisante.

## 6.1. Programme de tests accomplis

La Fig.6-1 montre le schéma du programmes de tests. Les procédures d'étalonnage et de reconstitution *PCR1* et *PCR3* sont décrites au *chapitre 4*; les fonctions spline sont décrites au *chapitre 3*. Dans les cas des données entachées par les erreurs, l'étalonnage est répété  $R=10$  fois pour différentes réalisations des erreurs de données.

Les procédures des fonctions spline peuvent être classées en deux types:

(1) procédures d'estimation de paramètres: *ParIntSplP1*, *ParIntSplC1*, *ParLmsSplP1*, *ParLmsSplC1*, *ParSmoSplP1*, *ParSmoSplC1*, *ParSmoSplβ1*;

(2) procédures d'approximation de valeurs: *ValSplP1*, *ValSplC1*.

Les procédures d'estimation de paramètres peuvent générer deux formats des paramètres:

(1) les paramètres du type I sont utilisés dans la formule suivante:

$$y = p_{0,i} + p_{1,i}x + p_{2,i}x^2 + \dots + p_{k,i}x^k \quad (6-1)$$

où,  $p_i = [ p_{0,i}, p_{1,i}, p_{2,i}, \dots, p_{k,i} ]$ , pour  $i = 1, \dots, N-1$ .

(2) les paramètres du type II sont utilisés dans la formule suivante:

$$y = p_{0,i} + p_{1,i}(x - x_i) + p_{2,i}(x - x_i)^2 + \dots + p_{k,i}(x - x_i)^k \quad (6-2)$$

où,  $p_i = [ p_{0,i}, p_{1,i}, p_{2,i}, \dots, p_{k,i} ]$ , pour  $i = 1, \dots, N-1$ .

Normalement, les procédures d'estimation de paramètres nous offrent la possibilité de choix de conditions aux limites différentes.

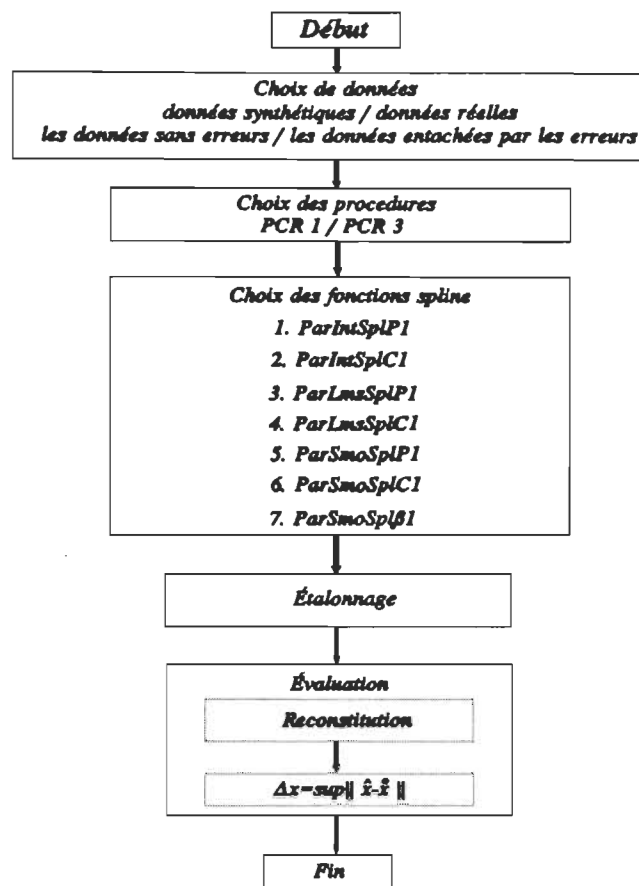


Fig.6-1. Le programme des tests.

## 6.2. Résultats d'évaluation obtenus pour les données synthétiques

Nous avons avant tout un vecteur d'entrée ( $x$ ), un vecteur d'influence ( $v$ ) et un modèle mathématique ( $G(x, v)$ ). Nous faisons une estimation des deux signaux pour obtenir une matrice de sortie pure ( $y$ ). Cette dernière est ensuite entachée par les erreurs avec une série de nombres aléatoires d'une certaine amplitude. Cette amplitude donne une idée de la variance d'erreur. Nous avons maintenant les signaux d'entrée ( $x$ ), d'influence ( $v$ ) et de sortie brut ( $y$ ). Le résultat de reconstitution ( $\hat{x}$ ) est mis sous forme de graphique et comparé au signal d'entrée original ( $x$ ). L'erreur absolue est calculée selon:

$$\Delta x = | \hat{x} - x | \quad (6-3)$$

Les données d'étalonnage sont entachées additivement à l'aide de nombres pseudo-aléatoires

uniformément distribués autour de zéro:

$$\text{CAS1: } \bar{\Delta x}^{cal} = 1E-2, \bar{\Delta v}^{cal} = 1E-2, \bar{\Delta y}^{cal} = 1E-3;$$

$$\text{CAS2: } \bar{\Delta x}^{cal} = 1E-2, \bar{\Delta v}^{cal} = 1E-2, \bar{\Delta y}^{cal} = 2E-3;$$

$$\text{CAS3: } \bar{\Delta x}^{cal} = 1E-7, \bar{\Delta v}^{cal} = 1E-2, \bar{\Delta y}^{cal} = 1E-2.$$

Pour toutes les valeurs  $x_{ij}^{cal}$  correspondant aux  $y_{ij}^{cal}$ , les fonctions spline sont calculées avec l'algorithme d'étalonnage. Chaque test d'estimation des paramètres et d'approximation de données synthétiques entachées par les erreurs, réalisé pour des valeurs particulières de  $N_y$  et  $N_x$ , est répété pour 10 réalisations d'erreur différentes. Les tests de performances des fonctions spline sont effectués selon les critères suivants:

- ***l'estimation de paramètres***: étude des fonctions spline en termes de rapidité et de stabilité, les résultats sont présentés en fonction de l'erreur absolue maximale.
- ***l'approximation de valeurs***: étude de la capacité des fonctions spline à approximer les informations acquises à partir des données inconnues. C'est ce qui permet de reconstituer les mesurandes dans notre application en fonction de sortie des capteurs. Les résultats sont présentés en fonction de l'erreur absolue maximale commise sur le mesurande reconstitué  $\hat{x}$ , soit  $\Delta\hat{x}$ .

De nombreux tests d'estimation de paramètres ont été effectués en variant les différents paramètres des fonctions spline: le nombre de données, le niveau d'erreur sur les données d'étalonnage. Nous allons donc nous intéresser aux performances des fonctions spline pour la reconstitution de mesurandes, ce qui revient à tester sa capacité d'approximation des données. Nous allons pouvoir montrer la validité des méthodes d'étalonnage et de reconstitution de mesurandes dans les cas des systèmes de mesure à l'étude.

Les résultats des fonctions spline différentes utilisées dans les différents cas avec les niveaux d'erreur différents sont montrés dans les Tableaux 6-2, 6-3 et 6-4. Les erreurs sont les erreurs absolues maximales et les quantités dans les parenthèses sont les valeurs  $(N_x^{cal} \times N_v^{cal})_{\min}$  qui sont obtenues à la position d'erreur minimale et correspondent aux produits minimaux des  $N_x^{cal} \times N_v^{cal}$  sur la même valeur d'erreur.

### 6.3 Résultats d'évaluation obtenus pour les données réelles

Les résultats d'étalonnage et d'évaluation des méthodes avec des données réelles ont été obtenus pour CAS2. Les données réelles sont présentées dans *chapitre 5*.

Les résultats d'évolution de la qualité de l'étalonnage et de la reconstitution de mesurandes sont présentés sur la Tableau 6-1. Les résultats sont obtenus en utilisant la procédure PCR3. *IntSplC1* manifeste la meilleur exactitude. *IntSplP1* et *LmsSplP1* manifestent la pire exactitude.

**Tableau 6-1. Résultats de la qualité de l'étalonnage et de la reconstitution de mesurandes pour les données réelles du CAS2.**

FONCTIONS SPLINE	$\delta_2 [\hat{x}^{val}]$	$\delta_\infty [\hat{x}^{val}]$
<i>IntSplP1</i>	$1.545 \times 10^{-2}$	$1.626 \times 10^{-2}$
<i>IntSplC1</i>	$2.412 \times 10^{-3}$	$3.858 \times 10^{-3}$
<i>LmsSplP1</i>	$1.545 \times 10^{-2}$	$1.626 \times 10^{-2}$
<i>LmsSplC1</i>	$8.172 \times 10^{-3}$	$1.354 \times 10^{-2}$
<i>SmoSplP1</i>	$3.754 \times 10^{-3}$	$4.036 \times 10^{-3}$
<i>SmoSplC1</i>	$4.409 \times 10^{-3}$	$6.978 \times 10^{-3}$
<i>SmoSplβ1</i>	$1.102 \times 10^{-2}$	$1.106 \times 10^{-2}$

## 6.4 Discussion des résultats

### 6.4.1. Influence du nombre de données sur la qualité de l'étalonnage

Nous considérons les résultats de l'étalonnage obtenus pour différentes valeurs de  $N_x^{cal} \in [5, 55]$  et de  $N_y^{cal} \in [5, 55]$ . Les exemples des résultats pour les données sont présentés respectivement dans les Fig.6-2 et Fig.6-3.

La Fig.6-2 correspond au CAS2, PCR1, *IntSplC1* et les données pour l'étalonnage dans le cas de données sans erreurs. Dans ce cas, la valeur minimale de l'erreur absolue maximale est de l'ordre de  $10^{-5}$ . L'étude de ce cas nous démontre qu'un nombre plus élevé de données permet une meilleure qualité d'étalonnage, ce qui est logique, d'autant plus que le nombre de données est plus grand, il offre plus d'informations aux fonctions spline pour reconstituer les mesurandes.

La Fig.6-3 correspond au CAS2, PCR1, *IntSplC1* et les données pour l'étalonnage dans le cas de données entachées par les erreurs. Dans ce cas, l'étalonnage pour les dix réalisations des erreurs se fait en un nombre faible de répétition et la valeur minimale de l'erreur absolue

maximale est de l'ordre de  $10^{-3}$ . Quand le nombre de mesurandes augmente à une certaine quantité, la valeur d'erreur paraît plus grande par rapport à l'intervalle de données pour l'étalonnage. Alors, l'effet de perturbation sera renforcé et l'erreur devient plus grande. Normalement, la courbe d'erreur présente le profil de vallée.

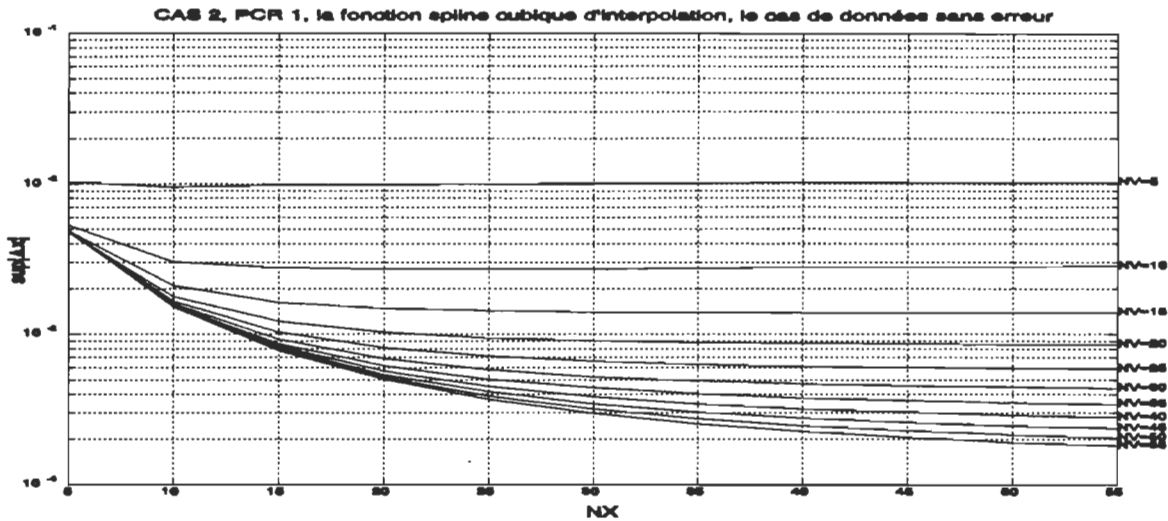


Fig.6-2. CAS2, PCR1, IntSplC1 et le cas de données sans erreurs.

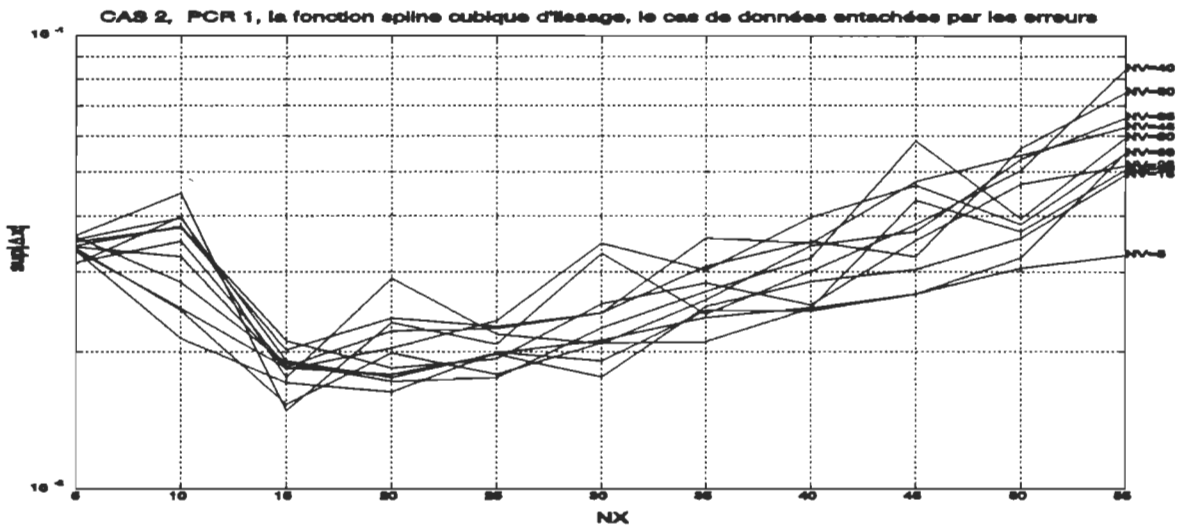


Fig.6-3. CAS2, PCR1, IntSplC1 et le cas de données entachées par les erreurs.

#### 6.4.2. Influence du nombre de données $N_v^{cal}$

L'un des principaux objectifs de ce projet est de limiter le rôle de l'influence sur la méthode d'étalonnage et de reconstitution de mesurandes pour les systèmes de mesure qui présentent l'inconvénient d'être très sensibles à l'influence. En général, les meilleures reconstitutions sont obtenues pour  $N_v^{cal} \cong 5 \sim 10$  dans les cas des données entachées par les erreurs. La fonction spline est capable de donner des résultats satisfaisants pour les influences utilisées pour l'estimation de paramètres.

#### 6.4.3. Influence des erreurs sur le mesurande $\bar{\Delta x}^{cal}$ , la grandeur d'influence $\bar{\Delta y}^{cal}$ et le résultat brut de mesure $\bar{\Delta y}^{cal}$

Les limites différentes d'erreur absolue du mesurande  $\bar{\Delta x}^{cal}$ , de la grandeur de l'influence  $\bar{\Delta y}^{cal}$  et du résultat brut de  $\bar{\Delta y}^{cal}$  sont respectivement présentées aux Fig.6-4, Fig.6-5 et Fig.6-6. Chaque simulation dépend de différents niveaux d'erreur. Les figures nous permettent de mettre en évidence le peu d'exactitude requise sur le mesurande, la grandeur d'influence et le résultat brut de mesure pour réaliser l'étalonnage du système, les résultats d'estimation sont comparables pour les différents niveaux d'erreur.

*SmoSplβ1* manifeste l'erreur maximale jusqu'à une valeur  $\leq 10^{-2}$ , mais, il est insensible à l'erreur dans ce champ. Il manifeste l'erreur minimale quand l'erreur  $\geq 10^{-3} \sim 10^{-2}$ .

*LmsSplC1*, *SmoSplC1*, *IntSplC1* et *LmsSplP1* manifestent l'erreur moyenne et ils sont insensibles à l'erreur jusqu'à l'erreur  $\leq 10^{-3}$ .

*IntSplP1* et *SmoSplP1* manifestent l'erreur minimale et ils sont insensibles à l'erreur jusqu'à l'erreur  $\leq 10^{-5}$ . Mais, ils manifestent l'erreur maximale quand l'erreur  $\geq 10^{-3}$  et sont sensibles à l'erreur.

L'influence des erreurs sur le mesurande  $\bar{\Delta x}^{cal}$  et le résultat brut de mesure  $\bar{\Delta y}^{cal}$  sont semblables pour toutes fonctions spline. L'effet de l'erreur sur la grandeur d'influence  $\bar{\Delta y}^{cal}$  est moindre que sur les deux autres.

Les résultats pour *CAS1*, *CAS2* et *CAS3* sont semblables.

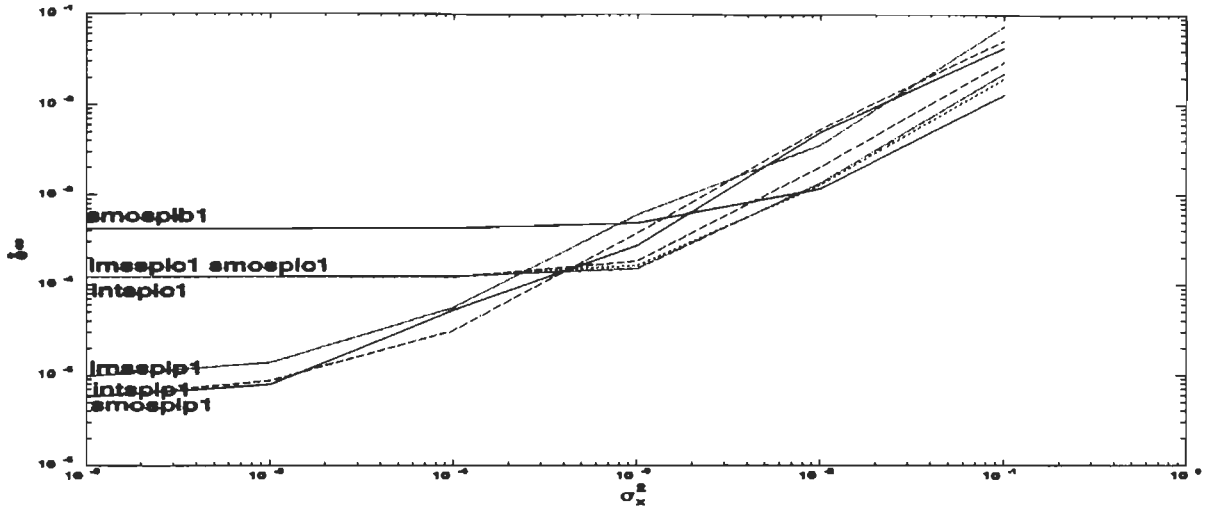


Fig.6-4. Les courbes d'erreur maximale relative pour les fonctions spline différentes dans le CAS2 avec les niveaux d'erreur différents sur le mesurande x.

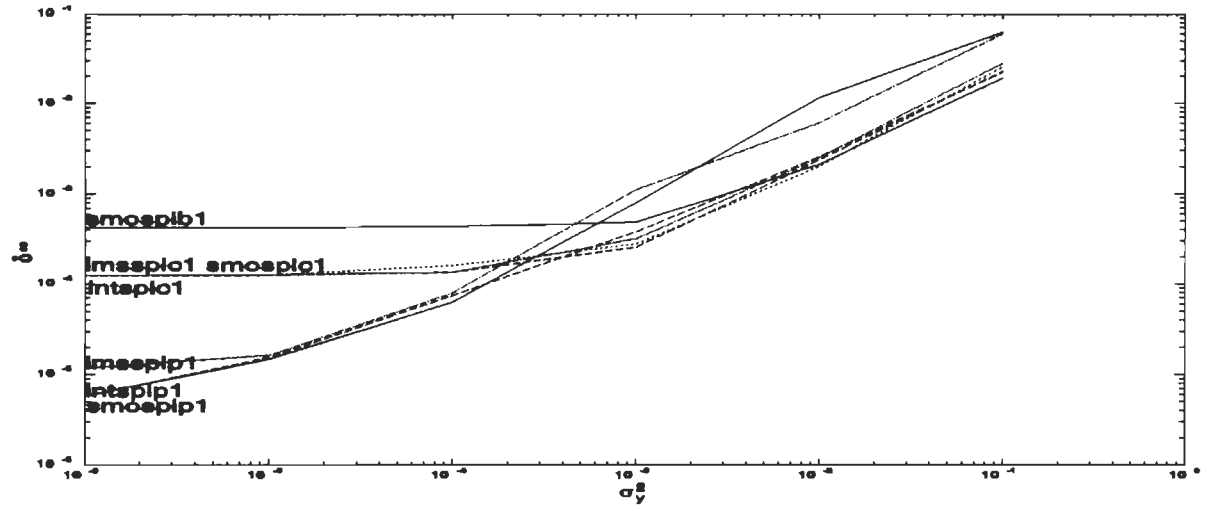


Fig.6-5. Les courbes d'erreur maximale relative pour les fonctions spline différentes dans le CAS2 avec les niveaux d'erreur différents sur le résultat brut y.

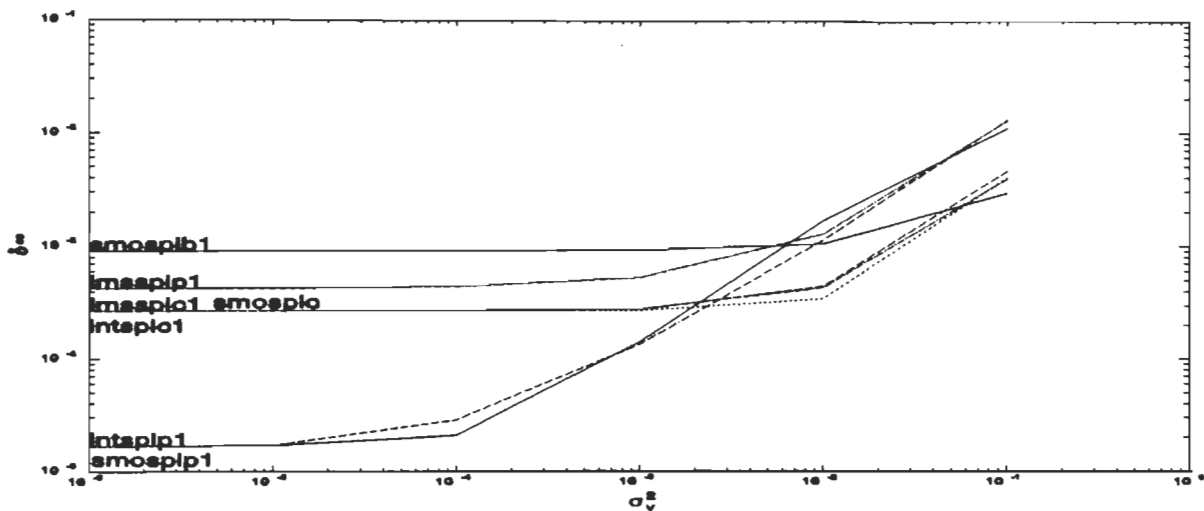


Fig.6-6. Les courbes d'erreur maximale relative pour les fonctions spline différentes dans le CAS2 avec les niveaux d'erreur différents sur la grandeur d'influence  $v$ .

#### 6.4.4. Influence du degré de la fonction spline

Les fonctions spline étudiées dans ce projet sont classées en deux types: deuxième degré et troisième degré. Selon la théorie d'approximation, pour toutes les approximations avec un polynôme l'exactitude dépend du degré du polynôme d'approximation.

Les Fig.6-7 et Fig.6-8 montrent l'évolution de l'erreur absolue maximale pour la fonction spline parabolique d'interpolation (*IntSplP1*) et de la fonction spline cubique d'interpolation (*IntSplC1*) pour le cas des données entachées par les erreurs (CAS2, PCR3). Nous pouvons voir que *IntSplC1* offre une meilleure performance sur la capacité d'anti-perturbation et une meilleure exactitude. Aussi, elle est insensible au nombre de la grandeur d'influence  $v$ .

Bien sûr, nous pouvons modifier la méthode d'approximation des fonctions spline afin d'améliorer l'exactitude. Par exemple, la fonction spline cubique d'interpolation (*IntSplC1*) et la fonction spline cubique de lissage (*SmoSplC1*) montrent la différence d'exactitude d'approximation sur les Fig.6-8 et Fig.6-9. Elles ont le même degré.

Aussi, nous pouvons comparer la fonction spline cubique de lissage (*SmoSplC1*) et la fonction *Beta*-spline (*SmpSplβ1*). Les Fig.6-9 et Fig.6-10 montrent l'évolution de l'erreur absolue maximale pour la fonction spline cubique de lissage et de la fonction *Beta*-spline cubique pour le cas de données entachées par les erreurs (CAS2, PCR1).

Les Fig.6-11 et Fig.6-12 montrent l'évolution de l'erreur absolue maximale pour la fonction spline parabolique d'interpolation (*IntSplP1*) et de la fonction spline parabolique de lissage



(*SmoSplP1*) pour le cas des données sans erreur. Nous pouvons voir que la courbe d'erreur de la fonction spline parabolique de lissage (*SmoSplP1*) est insensible aux nombres de données  $N_x^{cal}$  et  $N_y^{cal}$  quand  $N_x^{cal} \geq 30$  et  $N_y^{cal} \geq 30$ .

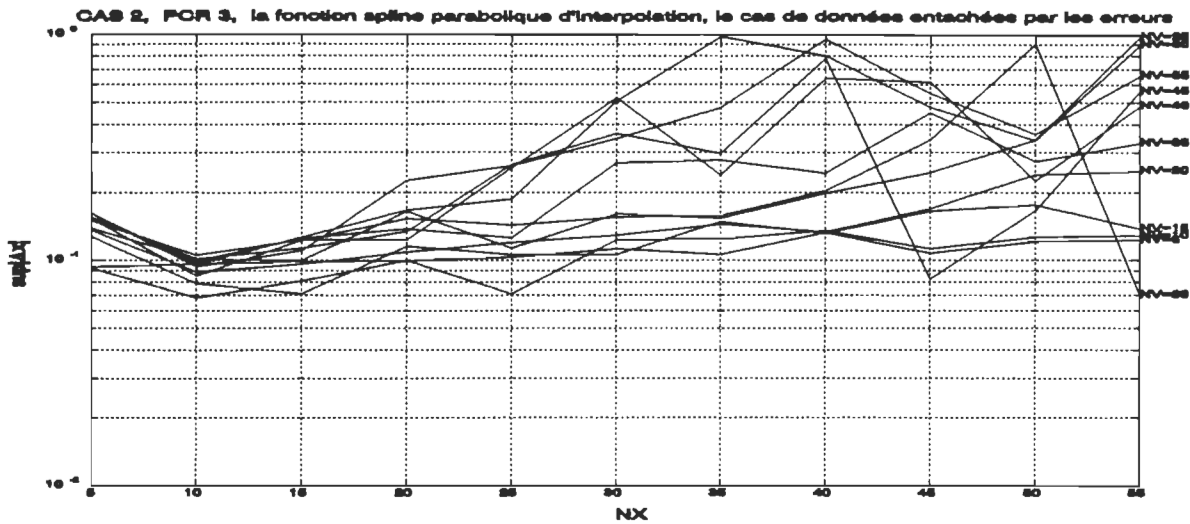


Fig.6-7. CAS2, PCR3, *IntSplP1* et le cas de données entachées par les erreurs.

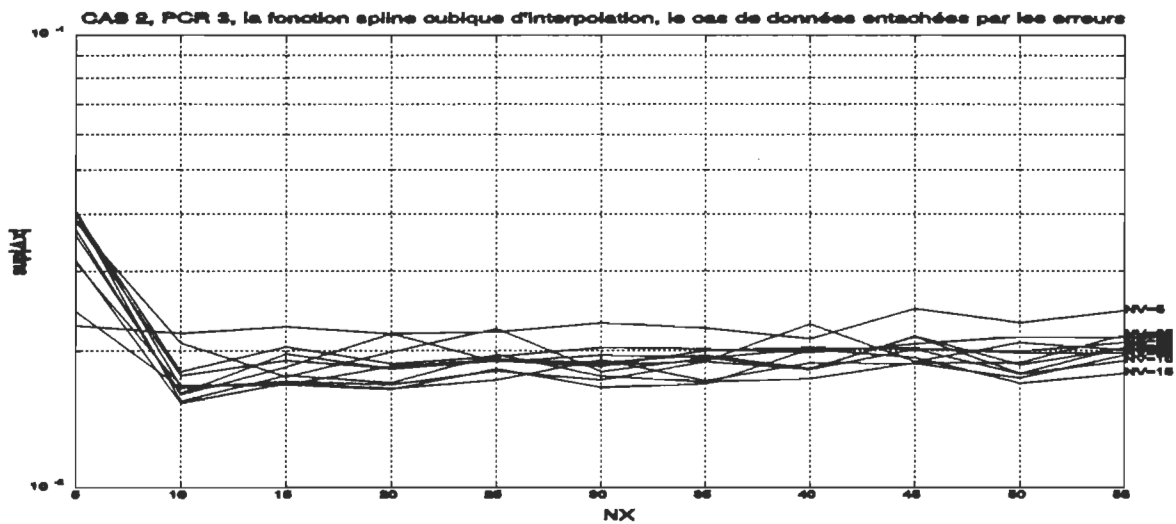


Fig.6-8. CAS2, PCR3, *IntSplC1* et le cas de données entachées par les erreurs.

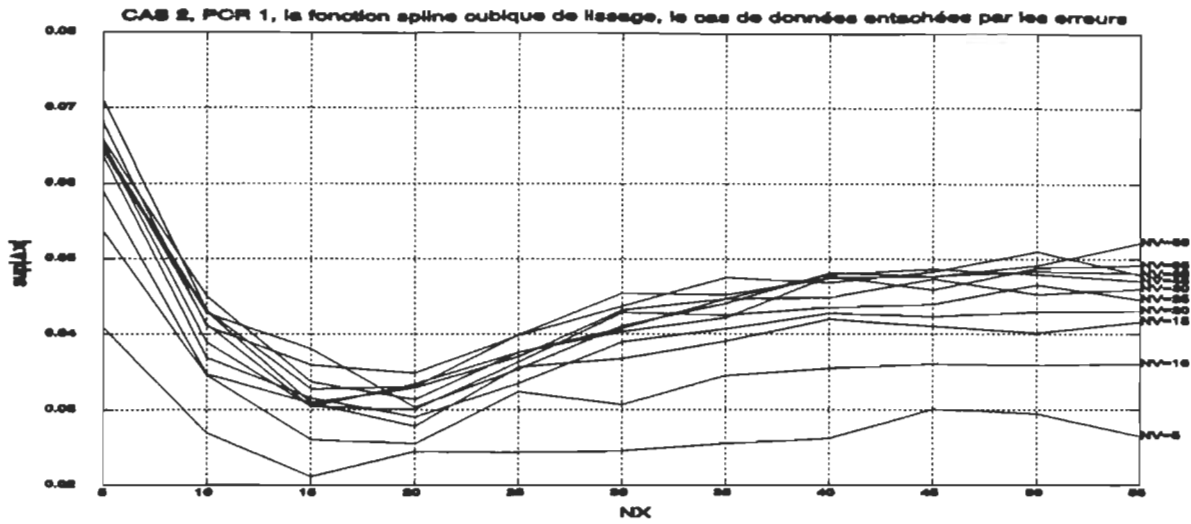


Fig.6-9. CAS2, PCR1, SmoSplC1 et le cas de données entachées par les erreurs.

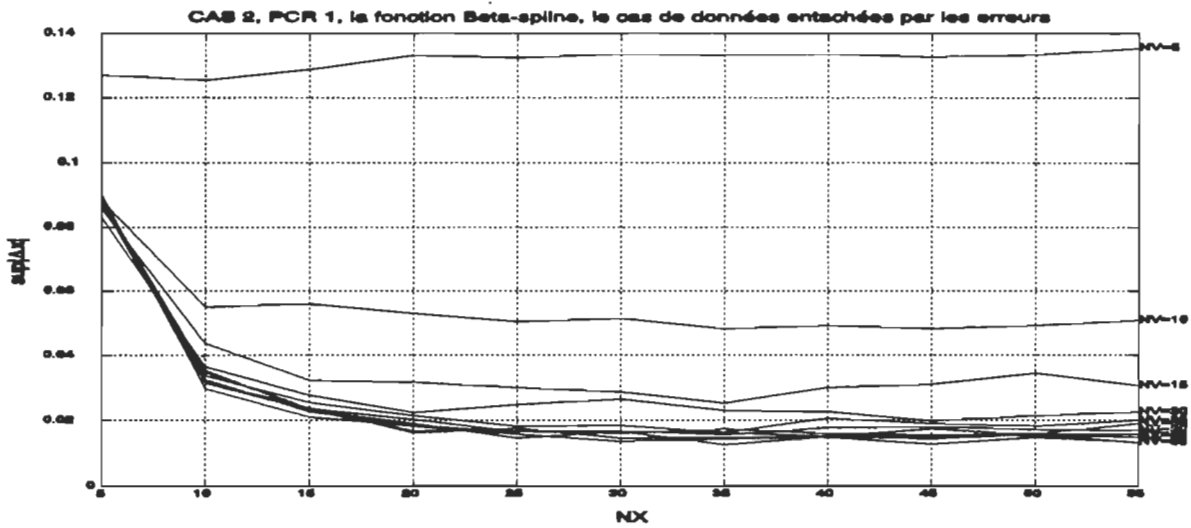


Fig.6-10. CAS2, PCR1, SmoSplβ1 et le cas de données entachées par les erreurs.

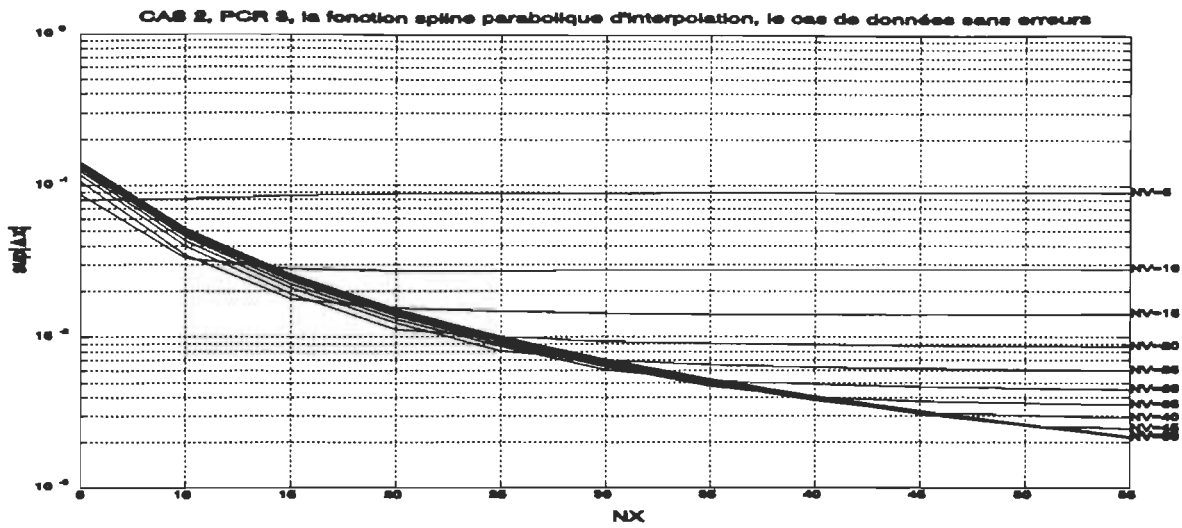


Fig.6-11. CAS2, PCR3, IntSplP1 et le cas de données sans erreurs.

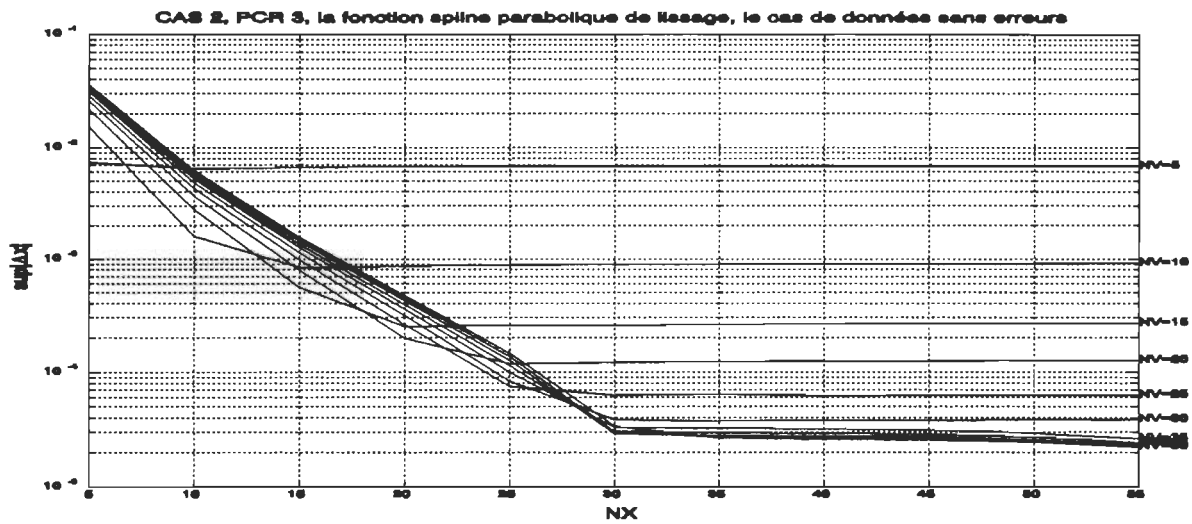


Fig.6-12. CAS2, PCR3, SmoSplP1 et le cas de données sans erreurs.

#### 6.4.5. Sensibilité des fonctions spline au niveau d'erreur

Nous comparons quatre fonctions spline: la fonction spline cubique de lissage (*SmoSplC1*), la fonction spline cubique d'interpolation (*IntSplC1*), la fonction spline cubique de moindres carrés (*LmsSplC1*) et la fonction *Beta*-spline (*SmoSplβ1*) pour CAS2. Les résultats d'approximation de des fonctions (pour dix réalisations) et les niveaux d'erreur sur les données pour l'étalonnage sont montrés sur la Fig.6-13.

Quand le niveau d'erreur est moins que 0.15, la fonction spline cubique de lissage (*SmoSplC1*), la fonction spline cubique de moindres carrés (*LmsSplC1*) et la fonction spline d'interpolation (*IntSplC1*) manifestent l'erreur minimale et la fonction *Beta*-spline (*SmoSplβ1*) manifeste l'erreur maximale.

Quand le niveau d'erreur est entre 0.15~0.4, la fonction spline cubique d'interpolation (*IntSplC1*) et la fonction spline cubique de moindres carrés (*LmsSplC1*) manifestent l'erreur minimale et la fonction spline cubique de lissage (*SmoSplC1*) manifeste l'erreur maximale.

Quand le niveau d'erreur est plus grand que 0.4, la fonction *Beta*-spline (*SmoSplβ1*) manifeste l'erreur minimale et maintient cette caractéristique. Mais la fonction spline cubique de lissage (*SmoSplC1*) manifeste toujours l'erreur maximale.

Alors, la fonction *Beta*-spline (*SmoSplβ1*) possède la capacité d'anti-perturbation quand le niveau d'erreur est plus grand.

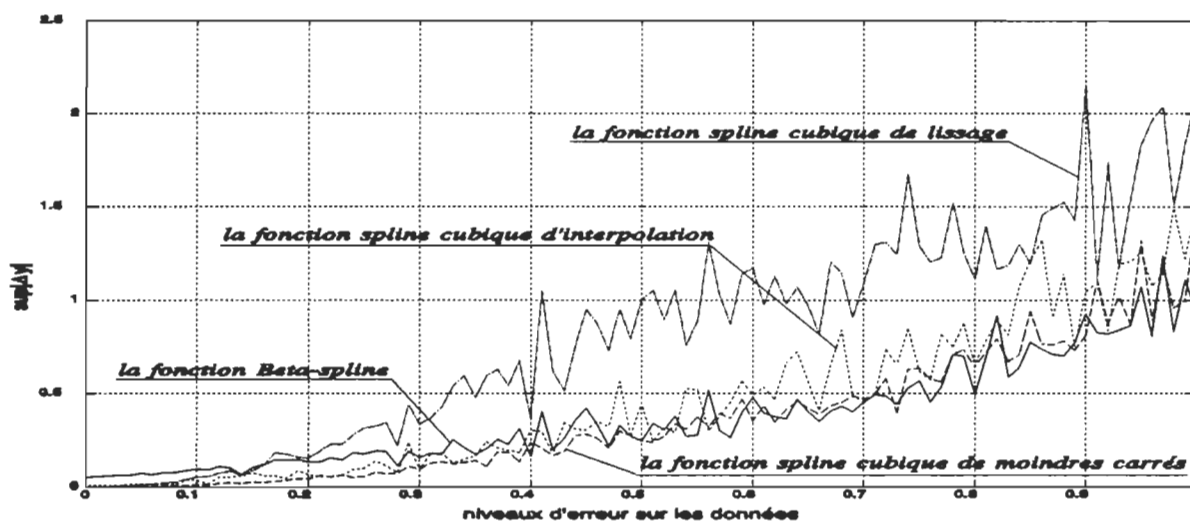


Fig.6-13 Les résultats d'approximation des fonctions et les niveaux d'erreur sur les données pour l'étalonnage.

#### 6.4.6. Rapidité des fonctions spline

Les valeurs minimales d'erreur maximale relative  $\delta_\infty$  et les valeurs de *FLOPS* des sept fonctions spline pour l'étalonnage pour *CAS2* sont montrées aux Fig.6-14 et Fig.6-15. La valeur dans les parenthèses est le nombre correspondant à l'erreur  $\delta_\infty$  minimale. Les valeurs de  $\delta_2$ ,  $\delta_\infty$  et *FLOPS* sont dans les Tableaux 6-5 et 6-6.

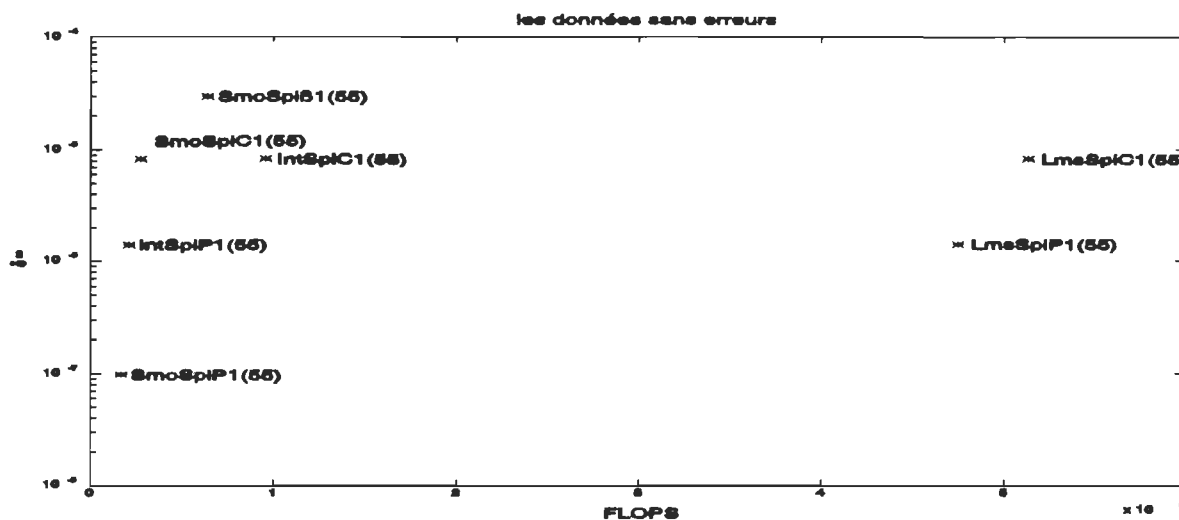


Fig.6-14 La valeur minimale de l'erreur maximale relative et la valeur de *FLOPS* dans le cas de données sans erreurs.

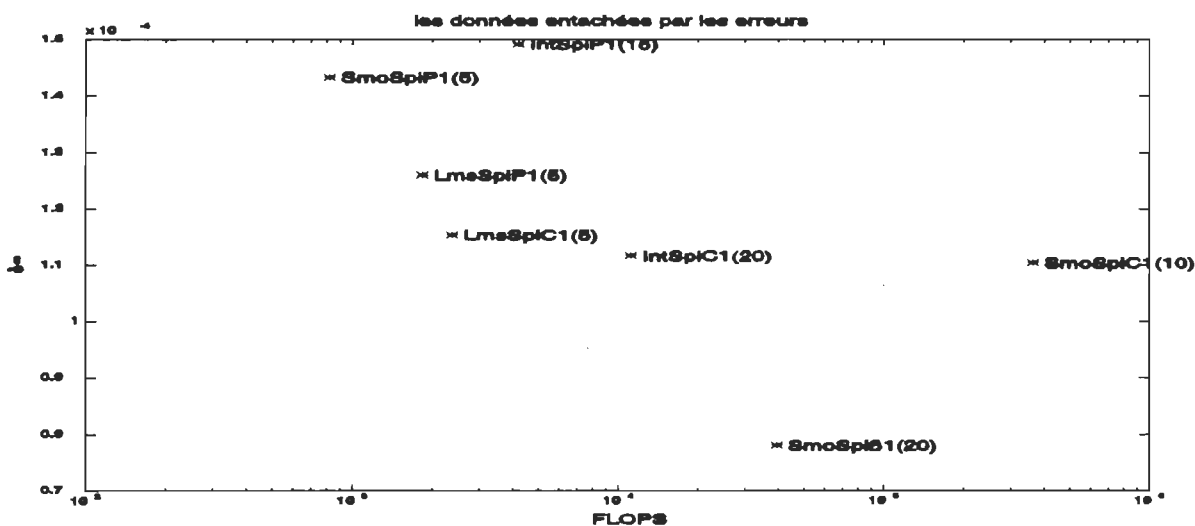


Fig.6-15 La valeur minimale de l'erreur maximale relative et la valeur de *FLOPS* dans le cas de données entachées par les erreurs.

Dans le cas de données sans erreurs, la fonction spline parabolique de lissage (*SmoSpiP1*)

possède l'erreur minimale et la valeur minimale de *FLOPS*, mais la fonction spline cubique d'interpolation (*IntSplC1*) possède la valeur maximale de *FLOPS*.

Dans le cas de données entachées par les erreurs, la fonction spline parabolique de lissage (*SmoSplP1*) possède la valeur minimale de *FLOPS*, mais la fonction *Beta*-spline (*SmoSplβ1*) possède l'erreur minimale et la fonction spline cubique de lissage possède la valeur maximale de *FLOPS*.

#### 6.4.7. Analyse des résultats d'évaluation de la qualité de l'étalonnage pour les CAS1, CAS2 et CAS3

L'analyse des résultats de calcul montrée dans les Tableaux 6-2, 6-3 et 6-4 nous permet de répartir la performance d'étalonnage accompli par les combinaisons des procédures d'étalonnage et de reconstitution et les fonctions spline suivantes:

<i>IntSplP1 - PCR1</i> ,	<i>IntSplP1 - PCR3</i> ,
<i>IntSplC1 - PCR1</i> ,	<i>IntSplC1 - PCR3</i> ,
<i>LmsSplP1 - PCR1</i> ,	<i>LmsSplP1 - PCR3</i> ,
<i>LmsSplC1 - PCR1</i> ,	<i>LmsSplC1 - PCR3</i> ,
<i>SmoSplP1 - PCR1</i> ,	<i>SmoSplP1 - PCR3</i> ,
<i>SmoSplC1 - PCR1</i> ,	<i>SmoSplC1 - PCR3</i> ,
<i>SmoSplβ1 - PCR1</i> ,	<i>SmoSplβ1 - PCR3</i> .

Dans les Tableaux 6-2, 6-3 et 6-4, nous avons le résultat final d'erreur absolue maximale  $\Delta x^*$  qui correspond au valeur de  $(N_x^{cal} \times N_v^{cal})_{\min}$  selon (5-36). La quantité dans les parenthèses est la valeur  $(N_x^{cal} \times N_v^{cal})_{\min}$  qui est obtenue à la position minimale d'erreur absolue maximale et correspond au produit minimal du  $N_x^{cal} \times N_v^{cal}$  sur la même valeur d'erreur. Les résultats dans les tableaux aboutissent aux conclusions suivantes:

- Pour les données sans erreurs:
  - les combinaisons *IntSplC1 - PCR1* et *SmoSplC1 - PCR1* toujours montrent la meilleure exactitude pour tous les cas dans les tests;
  - les combinaisons *SmoSplβ1 - PCR1* et *SmoSplβ1 - PCR3* pour CAS1 montrent la pire exactitude;
  - la combinaison *SmoSplC1 - PCR1* pour CAS2 montre la pire exactitude;
  - la combinaison *IntSplP1 - PCR1* pour CAS3 montre la pire exactitude.
 Les différences entre les meilleures exactitudes et les pires exactitudes sont 15 à 45 fois.
- Pour les données entachées par les erreurs:
  - la combinaison *SmoSplβ1 - PCR3* pour CAS1 montre la meilleure exactitude;
  - la combinaison *SmoSplP1 - PCR1* pour CAS2 montre la meilleure exactitude;

la combinaison *SmoSplC1 - PCR1* pour *CAS3* montre la meilleure exactitude;  
la combinaison *LmsSplP1 - PCR3* pour *CAS1* montre la pire exactitude;  
la combinaison *IntSplP1 - PCR3* pour *CAS2* montre la pire exactitude;  
la combinaison *IntSplP1 - PCR1* pour *CAS3* montre la pire exactitude.  
Les différences entre les meilleures exactitudes et les pires exactitudes sont 5 à 10 fois.

- Les combinaisons *IntSplC1 - PCR1* et *SmoSplC1 - PCR1* peuvent être recommandées pour l'étalonnage de différentes catégories de canaux de mesure (détecteurs, capteurs, convertisseurs, etc.) - en raison de leurs caractéristiques statiques très diversifiées sur les données d'étalonnage entachées par les erreurs dans une échelle "réaliste".
- Pour les données entachées par les erreurs, la meilleure exactitude n'est pas fixée sur une seule fonction spline et une seule procédure *PCR1* ou *PCR3*. Mais, nous pouvons trouver que les fonctions spline de lissage, incluant parabolique, cubique et *Beta*-spline, montrent souvent la meilleure exactitude; les fonctions spline paraboliques d'interpolation et de moindres carrés montrent souvent la pire exactitude.
- Les procédures *PCR1* et *PCR3* ne donnent pas une exactitude très différente.

Tableau 6-2. Résultats d'évaluation de la qualité de l'étalonnage pour le CAS1.

FONCTIONS SPLINE	PCR1		PCR3	
	$\bar{\Delta x}^{cal} = 0$ $\bar{\Delta v}^{cal} = 0$ $\bar{\Delta y}^{cal} = 0$	$\bar{\Delta x}^{cal} = 1E-2$ $\bar{\Delta v}^{cal} = 1E-2$ $\bar{\Delta y}^{cal} = 1E-3$	$\bar{\Delta x}^{cal} = 0$ $\bar{\Delta v}^{cal} = 0$ $\bar{\Delta y}^{cal} = 0$	$\bar{\Delta x}^{cal} = 1E-2$ $\bar{\Delta v}^{cal} = 1E-2$ $\bar{\Delta y}^{cal} = 1E-3$
<i>IntSplP1</i>	$2.265 \times 10^{-4}$ (50×55)	$1.378 \times 10^{-2}$ (14×7)	$2.037 \times 10^{-4}$ (49×55)	$1.047 \times 10^{-2}$ (12×10)
<i>IntSplC1</i>	$1.836 \times 10^{-5}$ (54×48)	$4.010 \times 10^{-3}$ (11×6)	$2.014 \times 10^{-5}$ (55×45)	$3.800 \times 10^{-3}$ (15×10)
<i>LmsSplP1</i>	$1.737 \times 10^{-4}$ (53×47)	$1.430 \times 10^{-2}$ (12×10)	$2.273 \times 10^{-4}$ (45×55)	$2.153 \times 10^{-2}$ (37×10)
<i>LmsSplC1</i>	$1.866 \times 10^{-5}$ (48×44)	$4.380 \times 10^{-3}$ (11×7)	$2.397 \times 10^{-5}$ (50×27)	$4.683 \times 10^{-3}$ (5×31)
<i>SmoSplP1</i>	$1.078 \times 10^{-4}$ (55×47)	$4.0 \times 10^{-3}$ (12×10)	$1.673 \times 10^{-4}$ (54×47)	$3.812 \times 10^{-3}$ (11×10)
<i>SmoSplC1</i>	$1.836 \times 10^{-5}$ (54×48)	$3.809 \times 10^{-3}$ (17×8)	$2.0145 \times 10^{-5}$ (55×45)	$4.83 \times 10^{-3}$ (19×16)
<i>SmoSplβ1</i>	$2.8105 \times 10^{-4}$ (55×42)	$4.8 \times 10^{-3}$ (24×20)	$2.8105 \times 10^{-4}$ (55×42)	$2.197 \times 10^{-3}$ (32×40)



Tableau 6-3. Résultats d'évaluation de la qualité de l'étalonnage pour le CAS2.

FONCTIONS SPLINE	PCR1		PCR3	
	$\bar{\Delta x}^{cal} = 0$ $\bar{\Delta v}^{cal} = 0$ $\bar{\Delta y}^{cal} = 0$	$\bar{\Delta x}^{cal} = 1E-2$ $\bar{\Delta v}^{cal} = 1E-2$ $\bar{\Delta y}^{cal} = 2E-3$	$\bar{\Delta x}^{cal} = 0$ $\bar{\Delta v}^{cal} = 0$ $\bar{\Delta y}^{cal} = 0$	$\bar{\Delta x}^{cal} = 1E-2$ $\bar{\Delta v}^{cal} = 1E-2$ $\bar{\Delta y}^{cal} = 2E-3$
<i>IntSplP1</i>	$2.026 \times 10^{-4}$ (47×55)	$5.459 \times 10^{-3}$ (15×10)	$2.371 \times 10^{-4}$ (54×48)	$7.627 \times 10^{-3}$ (11×14)
<i>IntSplC1</i>	$1.995 \times 10^{-5}$ (51×54)	$2.100 \times 10^{-3}$ (20×10)	$2.1903 \times 10^{-5}$ (49×55)	$1.935 \times 10^{-3}$ (11×14)
<i>LmsSplP1</i>	$5.037 \times 10^{-4}$ (47×55)	$2.477 \times 10^{-3}$ (7×10)	$2.403 \times 10^{-4}$ (55×48)	$3.798 \times 10^{-3}$ (9×10)
<i>LmsSplC1</i>	$2.984 \times 10^{-5}$ (48×55)	$1.950 \times 10^{-3}$ (13×7)	$2.759 \times 10^{-5}$ (55×47)	$2.795 \times 10^{-3}$ (13×9)
<i>SmoSplP1</i>	$8.957 \times 10^{-4}$ (47×55)	$1.867 \times 10^{-3}$ (17×10)	$1.673 \times 10^{-4}$ (53×53)	$3.848 \times 10^{-3}$ (10×5)
<i>SmoSplC1</i>	$1.995 \times 10^{-5}$ (47×55)	$1.925 \times 10^{-3}$ (13×7)	$2.190 \times 10^{-5}$ (49×55)	$3.515 \times 10^{-3}$ (11×7)
<i>SmoSplβ1</i>	$2.919 \times 10^{-4}$ (48×55)	$2.106 \times 10^{-3}$ (21×41)	$2.919 \times 10^{-4}$ (48×55)	$2.044 \times 10^{-3}$ (29×44)

Tableau 6-4. Résultats d'évaluation de la qualité de l'étalonnage pour le CAS3.

FONCTIONS SPLINE	PCR1		PCR3	
	$\bar{\Delta x}^{cal} = 0$ $\bar{\Delta v}^{cal} = 0$ $\bar{\Delta y}^{cal} = 0$	$\bar{\Delta x}^{cal} = 1E-7$ $\bar{\Delta v}^{cal} = 1E-2$ $\bar{\Delta y}^{cal} = 1E-2$	$\bar{\Delta x}^{cal} = 0$ $\bar{\Delta v}^{cal} = 0$ $\bar{\Delta y}^{cal} = 0$	$\bar{\Delta x}^{cal} = 1E-7$ $\bar{\Delta v}^{cal} = 1E-2$ $\bar{\Delta y}^{cal} = 1E-2$
<i>IntSplP1</i>	$4.491 \times 10^{-6}$ (55×37)	$6.789 \times 10^{-5}$ (23×24)	$1.133 \times 10^{-6}$ (44×55)	$6.587 \times 10^{-5}$ (30×10)
<i>IntSplC1</i>	$1.7723 \times 10^{-7}$ (48×55)	$2.0218 \times 10^{-5}$ (12×7)	$1.8817 \times 10^{-7}$ (54×53)	$2.523 \times 10^{-5}$ (15×10)
<i>LmsSplP1</i>	$3.887 \times 10^{-6}$ (55×32)	$1.863 \times 10^{-5}$ (13×13)	$1.141 \times 10^{-6}$ (43×55)	$4.024 \times 10^{-5}$ (25×10)
<i>LmsSplC1</i>	$1.795 \times 10^{-7}$ (55×55)	$1.981 \times 10^{-5}$ (13×5)	$1.794 \times 10^{-7}$ (55×34)	$2.936 \times 10^{-5}$ (28×5)
<i>SmoSplP1</i>	$1.877 \times 10^{-6}$ (55×32)	$1.85 \times 10^{-5}$ (18×13)	$1.8405 \times 10^{-6}$ (55×34)	$2.16 \times 10^{-5}$ (21×6)
<i>SmoSplC1</i>	$1.772 \times 10^{-7}$ (48×55)	$1.41 \times 10^{-5}$ (13×8)	$1.882 \times 10^{-7}$ (54×53)	$2.522 \times 10^{-5}$ (16×12)
<i>SmoSplβ1</i>	$1.114 \times 10^{-6}$ (54×49)	$2.232 \times 10^{-5}$ (36×30)	$1.114 \times 10^{-6}$ (54×49)	$2.2559 \times 10^{-5}$ (41×29)

**Tableau 6-5. Résultats des valeurs  $\delta_2$ ,  $\delta_\infty$  et FLOPS pour l'étalonnage du CAS2 dans le cas de données sans erreurs.**

		<i>InuSplP1</i>	<i>InuSplC1</i>	<i>LmsSplP1</i>	<i>LmsSplC1</i>	<i>SmoSplP1</i>	<i>SmoSplC1</i>	<i>SmoSplB1</i>
$N_x^{ord} = 5$	$\delta_2$	$3.350462 \times 10^{-4}$	$7.675612 \times 10^{-4}$	$3.3505814 \times^{-4}$	$7.6758493 \times^{-4}$	$1.792358 \times 10^{-4}$	$7.675612 \times 10^{-4}$	$3.716177 \times 10^{-3}$
	$\delta_\infty$	$3.664193 \times 10^{-4}$	$1.542464 \times 10^{-3}$	$3.6643343 \times^{-4}$	$1.5424985 \times^{-3}$	$2.157598 \times 10^{-4}$	$1.542464 \times 10^{-3}$	$4.610969 \times 10^{-3}$
	Flops	866	864	1830	2367	637	1080	5579
$N_x^{ord} = 10$	$\delta_2$	$6.383347 \times 10^{-4}$	$9.800446 \times 10^{-3}$	$6.3201362 \times^{-4}$	$9.7988718 \times^{-3}$	$1.799715 \times 10^{-3}$	$9.800446 \times 10^{-3}$	$7.829702 \times 10^{-4}$
	$\delta_\infty$	$9.6633 \times 10^{-4}$	$2.982347 \times 10^{-4}$	$9.6017845 \times^{-4}$	$2.9818522 \times^{-4}$	$2.105133 \times 10^{-3}$	$2.982347 \times 10^{-4}$	$9.963406 \times 10^{-4}$
	Flops	2461	3178	6899	8776	1845	4959	11319
$N_x^{ord} = 15$	$\delta_2$	$9.178059 \times 10^{-4}$	$3.227741 \times 10^{-3}$	$8.9643926 \times^{-4}$	$3.2288416 \times^{-3}$	$4.86224 \times 10^{-4}$	$3.227741 \times 10^{-3}$	$3.29088 \times 10^{-4}$
	$\delta_\infty$	$9.907204 \times 10^{-4}$	$1.227146 \times 10^{-4}$	$9.7987572 \times^{-4}$	$1.2277915 \times^{-4}$	$5.649012 \times 10^{-4}$	$1.227146 \times 10^{-4}$	$4.235452 \times 10^{-4}$
	Flops	4219	6573	16782	20574	3240	12769	20656
$N_x^{ord} = 20$	$\delta_2$	$6.930816 \times 10^{-4}$	$1.500221 \times 10^{-3}$	$6.6593399 \times^{-4}$	$1.5010024 \times^{-3}$	$1.947698 \times 10^{-4}$	$1.500221 \times 10^{-3}$	$1.800678 \times 10^{-4}$
	$\delta_\infty$	$7.503246 \times 10^{-4}$	$6.650163 \times 10^{-3}$	$7.8136804 \times^{-4}$	$6.6507397 \times^{-3}$	$2.264249 \times 10^{-4}$	$6.650163 \times 10^{-3}$	$2.33063 \times 10^{-4}$
	Flops	6107	11191	33161	39423	4756	26152	26242
$N_x^{ord} = 25$	$\delta_2$	$5.057997 \times 10^{-4}$	$8.353031 \times 10^{-4}$	$5.7924019 \times^{-4}$	$8.3550295 \times^{-4}$	$9.673917 \times 10^{-7}$	$8.353031 \times 10^{-4}$	$1.1336 \times 10^{-4}$
	$\delta_\infty$	$5.477863 \times 10^{-4}$	$4.163588 \times 10^{-3}$	$7.0032528 \times^{-4}$	$4.1647413 \times^{-3}$	$1.123785 \times 10^{-4}$	$4.163588 \times 10^{-3}$	$1.47171 \times 10^{-4}$
	Flops	8094	17246	57765	67042	6371	46822	35262
$N_x^{ord} = 30$	$\delta_2$	$3.785664 \times 10^{-4}$	$5.199346 \times 10^{-4}$	$3.7869903 \times^{-4}$	$5.1990839 \times^{-4}$	$5.479299 \times 10^{-7}$	$5.199346 \times 10^{-4}$	$7.786552 \times 10^{-3}$
	$\delta_\infty$	$4.100293 \times 10^{-4}$	$2.84977 \times 10^{-3}$	$4.1074479 \times^{-4}$	$2.8495363 \times^{-3}$	$6.368656 \times 10^{-7}$	$2.84977 \times 10^{-3}$	$1.012826 \times 10^{-4}$
	Flops	10081	24826	92233	105025	7986	76367	42728
$N_x^{ord} = 35$	$\delta_2$	$2.918892 \times 10^{-4}$	$3.490985 \times 10^{-4}$	$4.2982350 \times^{-4}$	$3.5005549 \times^{-4}$	$3.399768 \times 10^{-7}$	$3.490985 \times 10^{-4}$	$5.676312 \times 10^{-3}$
	$\delta_\infty$	$3.161796 \times 10^{-4}$	$2.072297 \times 10^{-3}$	$5.7759760 \times^{-4}$	$2.0725299 \times^{-3}$	$3.950708 \times 10^{-7}$	$2.072297 \times 10^{-3}$	$7.393 \times 10^{-3}$
	Flops	12148	34307	138421	155263	9678	116663	35380
$N_x^{ord} = 40$	$\delta_2$	$2.31168 \times 10^{-4}$	$2.476077 \times 10^{-4}$	$2.3147874 \times^{-4}$	$2.4754409 \times^{-4}$	$2.251141 \times 10^{-7}$	$2.476077 \times 10^{-4}$	$4.320643 \times 10^{-3}$
	$\delta_\infty$	$2.504213 \times 10^{-4}$	$1.574482 \times 10^{-3}$	$2.5161484 \times^{-4}$	$1.5739379 \times^{-3}$	$2.617122 \times 10^{-7}$	$1.574482 \times 10^{-3}$	$5.632649 \times 10^{-3}$
	Flops	14349	46047	198110	219567	11513	169568	50813
$N_x^{ord} = 45$	$\delta_2$	$1.872672 \times 10^{-4}$	$1.830734 \times 10^{-4}$	$1.8771318 \times^{-4}$	$1.8301625 \times^{-4}$	$1.567288 \times 10^{-7}$	$1.830734 \times 10^{-4}$	$3.398407 \times 10^{-3}$
	$\delta_\infty$	$2.028781 \times 10^{-4}$	$1.236666 \times 10^{-3}$	$2.0438457 \times^{-4}$	$1.2361118 \times^{-3}$	$1.822024 \times 10^{-7}$	$1.236666 \times 10^{-3}$	$4.43354 \times 10^{-3}$
	Flops	16556	60062	272930	299502	13348	236598	59285
$N_x^{ord} = 50$	$\delta_2$	$1.546298 \times 10^{-4}$	$1.398407 \times 10^{-4}$	$1.5459897 \times^{-4}$	$1.3976893 \times^{-4}$	$1.134221 \times 10^{-7}$	$1.398407 \times 10^{-4}$	$2.742762 \times 10^{-3}$
	$\delta_\infty$	$1.675288 \times 10^{-4}$	$9.969644 \times 10^{-4}$	$1.6882748 \times^{-4}$	$9.9628476 \times^{-4}$	$1.318923 \times 10^{-7}$	$9.969644 \times 10^{-4}$	$3.580197 \times 10^{-3}$
	Flops	18763	76602	364630	396817	15183	24529	70816
$N_x^{ord} = 55$	$\delta_2$	$1.297532 \times 10^{-4}$	$1.096557 \times 10^{-4}$	$1.2890492 \times^{-4}$	$1.0964857 \times^{-4}$	$8.472634 \times 10^{-8}$	$1.096557 \times 10^{-4}$	$2.260047 \times 10^{-3}$
	$\delta_\infty$	$1.405845 \times 10^{-4}$	$8.207576 \times 10^{-4}$	$1.4072177 \times^{-4}$	$8.2068914 \times^{-4}$	$9.852213 \times 10^{-8}$	$8.207576 \times 10^{-4}$	$2.95143 \times 10^{-3}$
	Flops	20970	95917	474953	513255	17018	27514	64000

**Tableau 6-6. Résultats des valeurs  $\delta_2$ ,  $\delta_\infty$  et FLOPS pour l'étalonnage du CAS2 dans le cas de données entachées par les erreurs.**

		<i>IntSplP1</i>	<i>IntSplC1</i>	<i>LmsSplP1</i>	<i>LmsSplC1</i>	<i>SmoSplP1</i>	<i>SmoSplC1</i>	<i>SmoSplB1</i>
$N_x^{ent} = 5$	$\delta_2$	$4.480909 \times 10^{-4}$	$4.088239 \times 10^{-4}$	$3.2095215 \times 10^{-4}$	$2.7322625 \times 10^{-4}$	$3.053059 \times 10^{-4}$	$3.26095 \times 10^{-4}$	$1.248202 \times 10^{-3}$
	$\delta_\infty$	$1.91266 \times 10^{-3}$	$1.811438 \times 10^{-3}$	$1.2605862 \times 10^{-3}$	$1.1537285 \times 10^{-3}$	$1.433604 \times 10^{-3}$	$1.377337 \times 10^{-3}$	$4.584058 \times 10^{-3}$
	<i>Flops</i>	866	864	1830	2367	822	85816	7684
$N_x^{ent} = 10$	$\delta_2$	$3.610743 \times 10^{-4}$	$2.709622 \times 10^{-4}$	$5.6046407 \times 10^{-4}$	$3.4345439 \times 10^{-4}$	$3.115022 \times 10^{-4}$	$2.385291 \times 10^{-4}$	$1.754965 \times 10^{-4}$
	$\delta_\infty$	$1.762101 \times 10^{-3}$	$1.224641 \times 10^{-3}$	$3.1916905 \times 10^{-3}$	$1.9160751 \times 10^{-3}$	$1.503329 \times 10^{-3}$	$1.104158 \times 10^{-3}$	$9.439514 \times 10^{-4}$
	<i>Flops</i>	2450	3178	6881	8760	2230	363977	18177
$N_x^{ent} = 15$	$\delta_2$	$2.581616 \times 10^{-4}$	$2.547767 \times 10^{-4}$	$5.7376118 \times 10^{-4}$	$2.7573479 \times 10^{-4}$	$3.765523 \times 10^{-4}$	$2.739614 \times 10^{-4}$	$2.389884 \times 10^{-4}$
	$\delta_\infty$	$1.491996 \times 10^{-3}$	$1.590919 \times 10^{-3}$	$2.9374607 \times 10^{-3}$	$1.4108749 \times 10^{-3}$	$1.789594 \times 10^{-3}$	$1.21197 \times 10^{-3}$	$1.235507 \times 10^{-3}$
	<i>Flops</i>	4208	6573	16772	20558	3836	1199803	29102
$N_x^{ent} = 20$	$\delta_2$	$3.494109 \times 10^{-4}$	$2.40462 \times 10^{-4}$	$7.9594324 \times 10^{-4}$	$2.6459254 \times 10^{-4}$	$3.526403 \times 10^{-4}$	$3.118213 \times 10^{-4}$	$1.792016 \times 10^{-4}$
	$\delta_\infty$	$2.277235 \times 10^{-3}$	$1.116314 \times 10^{-3}$	$4.2400482 \times 10^{-3}$	$1.7752456 \times 10^{-3}$	$2.935642 \times 10^{-3}$	$1.727485 \times 10^{-3}$	$7.810971 \times 10^{-4}$
	<i>Flops</i>	6096	11173	33154	39392	5530	2263553	39511
$N_x^{ent} = 25$	$\delta_2$	$4.052642 \times 10^{-4}$	$2.658032 \times 10^{-4}$	$4.3484995 \times 10^{-4}$	$3.4654655 \times 10^{-4}$	$3.033502 \times 10^{-4}$	$2.827526 \times 10^{-4}$	$1.706589 \times 10^{-4}$
	$\delta_\infty$	$3.338933 \times 10^{-3}$	$1.299486 \times 10^{-3}$	$3.4557340 \times 10^{-3}$	$2.1888736 \times 10^{-3}$	$1.55862 \times 10^{-3}$	$1.551476 \times 10^{-3}$	$1.209533 \times 10^{-3}$
	<i>Flops</i>	8083	17264	57739	67046	14237	3641418	50022
$N_x^{ent} = 30$	$\delta_2$	$5.926909 \times 10^{-4}$	$2.546588 \times 10^{-4}$	$1.2746034 \times 10^{-3}$	$2.5059910 \times 10^{-4}$	$3.732346 \times 10^{-4}$	$2.685763 \times 10^{-4}$	$1.906449 \times 10^{-4}$
	$\delta_\infty$	$4.837661 \times 10^{-3}$	$1.651232 \times 10^{-3}$	$7.7822052 \times 10^{-3}$	$1.5103426 \times 10^{-3}$	$2.521565 \times 10^{-3}$	$1.565525 \times 10^{-3}$	$1.043125 \times 10^{-3}$
	<i>Flops</i>	10059	24826	92221	105051	15064	6792322	59573
$N_x^{ent} = 35$	$\delta_2$	$4.201873 \times 10^{-4}$	$2.676874 \times 10^{-4}$	$8.3779652 \times 10^{-4}$	$3.0385124 \times 10^{-4}$	$3.826557 \times 10^{-4}$	$2.279744 \times 10^{-4}$	$1.931543 \times 10^{-4}$
	$\delta_\infty$	$2.803047 \times 10^{-3}$	$1.460647 \times 10^{-3}$	$5.1785641 \times 10^{-3}$	$1.8760988 \times 10^{-3}$	$2.638923 \times 10^{-3}$	$1.4875 \times 10^{-3}$	$1.242185 \times 10^{-3}$
	<i>Flops</i>	12148	34307	138432	155264	11063	10154240	72076
$N_x^{ent} = 40$	$\delta_2$	$1.021344 \times 10^{-3}$	$2.679111 \times 10^{-4}$	$6.0013910 \times 10^{-4}$	$2.5901815 \times 10^{-4}$	$3.119754 \times 10^{-4}$	$2.777489 \times 10^{-4}$	$1.916318 \times 10^{-4}$
	$\delta_\infty$	$5.343543 \times 10^{-3}$	$1.493008 \times 10^{-3}$	$4.2837560 \times 10^{-3}$	$1.9252597 \times 10^{-3}$	$2.020495 \times 10^{-3}$	$1.540485 \times 10^{-3}$	$1.129325 \times 10^{-3}$
	<i>Flops</i>	14327	46047	198131	219584	19386	11243110	81495
$N_x^{ent} = 45$	$\delta_2$	$9.433044 \times 10^{-4}$	$2.787945 \times 10^{-4}$	$7.4296690 \times 10^{-4}$	$3.0353490 \times 10^{-4}$	$5.414299 \times 10^{-4}$	$2.986925 \times 10^{-4}$	$1.93843 \times 10^{-4}$
	$\delta_\infty$	$5.787086 \times 10^{-3}$	$1.511696 \times 10^{-3}$	$4.6120567 \times 10^{-3}$	$1.9583903 \times 10^{-3}$	$3.951117 \times 10^{-3}$	$1.503864 \times 10^{-3}$	$1.082773 \times 10^{-3}$
	<i>Flops</i>	16545	60080	272911	299482	15100	22536720	94166
$N_x^{ent} = 50$	$\delta_2$	$5.334632 \times 10^{-4}$	$2.723514 \times 10^{-4}$	$1.0988804 \times 10^{-3}$	$2.9545203 \times 10^{-4}$	$3.731787 \times 10^{-4}$	$2.924179 \times 10^{-4}$	$1.663158 \times 10^{-4}$
	$\delta_\infty$	$3.731745 \times 10^{-3}$	$1.583029 \times 10^{-3}$	$6.7813226 \times 10^{-3}$	$2.1652177 \times 10^{-3}$	$2.752633 \times 10^{-3}$	$1.604213 \times 10^{-3}$	$8.841305 \times 10^{-4}$
	<i>Flops</i>	18807	76638	364585	396876	17157	1100768	104179
$N_x^{ent} = 55$	$\delta_2$	$1.041791 \times 10^{-3}$	$2.846153 \times 10^{-4}$	$1.8588634 \times 10^{-3}$	$3.2536736 \times 10^{-4}$	$2.503685 \times 10^{-4}$	$3.115875 \times 10^{-4}$	$2.106301 \times 10^{-4}$
	$\delta_\infty$	$6.160504 \times 10^{-3}$	$1.959703 \times 10^{-3}$	$1.1692332 \times 10^{-2}$	$1.7497414 \times 10^{-3}$	$1.829934 \times 10^{-3}$	$1.785 \times 10^{-3}$	$1.462284 \times 10^{-3}$
	<i>Flops</i>	20959	95899	474959	513237	93381	1296532	115158

## CHAPITRE 7.

# CONCLUSION

L'objectif principal du mémoire consiste à apporter une contribution à la qualité d'un outil métrologique par des méthodes avancées de traitement des signaux de mesure et de développement des logiciels, et à étudier les possibilités d'application des fonctions spline aux problèmes d'étalonnage et de reconstitution de mesurandes statiques dans un système de mesure. Ceci revient à faire le lien entre deux domaines de recherche: les systèmes de mesure et les fonctions spline. Ce lien doit nous permettre de proposer une nouvelle méthode basée sur les propriétés des fonctions spline et se présenter comme une alternative aux méthodes existantes d'étalonnage. Les résultats espérés de ce travail de recherche sont essentiellement dans l'application métrologique de données non-linéaires et statiques.

Quelques méthodes d'étalonnage et de reconstitution de mesurandes analytiques ont été présentées. Elles donnent de très bons résultats dans le cas où l'on dispose d'informations suffisantes sur le modèle mathématique du système de mesure et sur les erreurs de perturbation. Cependant il est parfois impossible d'avoir ces informations et donc ces méthodes ne sont pas toujours applicables pour résoudre n'importe quel problème de reconstitution, c'est le cas des systèmes de mesure fortement non-linéaires.

À la suite d'une introduction au domaine des fonctions spline faite au *chapitre 3*, où les notions de base et les principaux types de fonctions spline sont passés en revue, nous avons pu choisir le type de fonction spline le plus approprié pour l'application. Les fonctions spline s'accordent avec le type de données à traiter, leurs capacités d'estimation sont élevées et elles sont relativement faciles d'utilisation. Les résultats d'étalonnage et de reconstitution de mesurandes pour les systèmes de mesure ont été réalisés pour des données synthétiques et réelles. Les résultats, présentés au *chapitre 6*, montrent la flexibilité dont font preuve les fonctions spline comme outils d'étalonnage dans le cas d'un système de mesure fortement non-linéaire et sensible à des facteurs d'influence. Les méthodes analytiques proposées prennent en considération ce facteur d'influence en mesurant l'influence et en la traitant comme donnée d'entrée. La méthode proposée basée sur les fonctions spline à une variable a permis d'être utilisée directement sur les caractéristiques d'entrée/sortie/influence des capteurs avec trois variables. En effet, durant la phase d'estimation de paramètres, la fonction spline étalonne le système de mesure en deux axes  $x$  et  $y$ . Par la suite, il est possible de reconstruire des mesurandes à partir de la sortie des

capteurs de mesurandes et de l'influence.

Une comparaison des résultats a été obtenue avec les méthodes basées sur les fonctions spline, les résultats de reconstitution sont comparables dans le cas d'une faible erreur de mesure sur le mesurande et l'influence lors de l'étalonnage.

## 7.1. Contribution algorithmique

Cette contribution consiste essentiellement au développement de sept algorithmes: *IntSplP1*, *IntSplC1*, *LmsSplP1*, *LmsSplC1*, *SmoSplP1*, *SmoSplC1* et *SmoSpl $\beta$ 1*.

Le développement de sept algorithmes de reconstitution de mesurandes basés sur les données non-linéaires, statiques et tridimensionnelles a été réalisé dans le but de les implanter dans une technologie de logiciel (Tableaux 6-2 ~ 6-4). Les conclusions sont les suivantes:

- En ce qui concerne la qualité des résultats de reconstitution dans le cas de données sans erreurs, les algorithmes d'approximation cubique d'interpolation, de moindres carrés et de lissage améliorent par dix fois l'exactitude par rapport aux algorithmes d'approximation parabolique d'interpolation, de moindres carrés et de lissage. Seule la fonction *Beta-spline* cubique présente la même classe d'exactitude que les algorithmes d'approximation parabolique. Selon la théorie d'approximation, l'exactitude de toutes les approximations avec le polynôme dépend du degré du polynôme d'approximation. Bien sûr, nous pouvons modifier la méthode d'approximation des fonctions spline afin d'améliorer l'exactitude.
- *LmsSplP1* présente une exactitude médiocre dans le cas de données sans erreurs par rapport aux autres algorithmes d'approximation parabolique et cubique. Mais, *LmsSplC1* présente la même classe d'exactitude dans le cas de données sans erreurs par rapport aux autres algorithmes d'approximation parabolique et cubique.
- Les algorithmes d'approximation parabolique, cubique et *Beta-spline* de lissage offrent une amélioration d'exactitude sur la qualité des résultats de reconstitution dans le cas de données entachées par les erreurs par rapport aux algorithmes d'approximation parabolique et cubique d'interpolation et de moindres carrés.
- *SmoSpl $\beta$ 1* présente une caractéristique particulière: cet algorithme semble être moins sensible aux erreurs de données quand le niveau d'erreur est plus grand. Le choix des paramètres  $\beta_1$  et  $\beta_2$  assure la continuité de la fonction spline.
- Les procédures *PCR1* et *PCR3* n'affectent pas l'exactitude sur la qualité des résultats de reconstitution dans le cas de données entachées par les erreurs ou de données sans erreurs,

c'est-à-dire que l'inverse de la fonction spline et l'inverse du modèle de données pour l'étalonnage ont la même exactitude;

- La grandeur d'influence  $v$  est considérée comme un deuxième mesurande à reconstruire à partir des résultats de conversion.
- Les algorithmes d'interpolation sont très souples et demandent peu de connaissances *a priori* sur les signaux à traiter. Les fonctions spline de lissage nécessitent un traitement plus complexe par rapport aux fonctions spline d'interpolation.
- Pour les données réelles, *IntSplC1* manifeste la meilleure exactitude. *IntSplP1* et *LmsSplP1* manifestent la pire exactitude.
- Dans le cas de données sans erreurs, l'étude nous démontre qu'un nombre plus élevé de données permet une meilleure qualité d'étalonnage, ce qui est logique, d'autant plus que le nombre de données est plus grand, il offre plus d'information aux fonctions spline pour reconstituer des mesurandes.
- Dans le cas de données entachées par les erreurs et quand le nombre de mesurandes augmente à une certaine quantité, la valeur d'erreur paraît plus grande par rapport à l'intervalle de données pour l'étalonnage. Alors, l'effet de perturbation augmente et l'erreur sera plus grande. Normalement, la courbe d'erreur présente le profil de vallée.
- En général, les meilleures reconstitutions sont obtenues pour  $N_v^{cal} \cong 5 \sim 10$  dans les cas des données entachées par les erreurs. La fonction spline est capable de donner des résultats satisfaisants pour les influences utilisées pour l'estimation de paramètres.
- Pour les influences des erreurs sur le mesurande  $\bar{\Delta}x^{cal}$ , la grandeur d'influence  $\bar{\Delta}v^{cal}$  et le résultat brut de mesure  $\bar{\Delta}y^{cal}$ : *SmoSplβ1* manifeste la pire exactitude jusqu'à une valeur  $\leq 10^{-2}$ , mais, il est insensible au niveau d'erreur dans ce champ. Il manifeste la meilleure exactitude quand l'erreur  $\geq 10^{-3} \sim 10^{-2}$ . *LmsSplC1*, *SmoSplC1*, *IntSplC1* et *LmsSplP1* manifestent l'exactitude moyenne et ils sont insensibles au niveau d'erreur jusqu'à l'erreur  $\leq 10^{-3}$ . *IntSplP1* et *SmoSplP1* manifestent la meilleure exactitude et ils sont insensibles au niveau d'erreur jusqu'à l'erreur  $\leq 10^{-5}$ . Mais, ils manifestent la pire exactitude quand l'erreur  $\geq 10^{-3}$  et sont sensibles au niveau d'erreur. L'influence des erreurs sur le mesurande  $\bar{\Delta}x^{cal}$  et le résultat brut de mesure  $\bar{\Delta}y^{cal}$  sont semblables pour toutes les fonctions spline. L'effet de l'erreur sur la grandeur d'influence  $\bar{\Delta}v^{cal}$  est moindre que sur les deux autres. Les résultats pour *CAS1*, *CAS2* et *CAS3* sont semblables. Quand le niveau d'erreur est plus petit, *SmoSplC1*, *LmsSplC1* et *IntSplC1* manifestent la meilleure exactitude et *SmoSplβ1* manifeste la pire exactitude. Quand le niveau d'erreur est moyen,

*IntSplC1* et *LmsSplC1* manifestent la meilleure exactitude et *SmoSplC1* manifeste la pire exactitude. Quand le niveau d'erreur est plus grand, *SmoSplβ1* manifeste la meilleure exactitude et maintient cette caractéristique. Mais *SmoSplC1* manifeste toujours la pire exactitude.

- Dans le cas de données sans erreurs, *SmoSplP1* possède l'exactitude maximale et la rapidité maximale. *IntSplC1* possède la rapidité minimale. Dans le cas de données entachées par les erreurs, *SmoSplP1* possède la rapidité maximale. *SmoSplβ1* possède l'exactitude maximale et *SmoSplC1* possède la rapidité minimale.
- Les combinaisons *IntSplC1 - PCR1* et *SmoSplC1 - PCR1* peuvent être recommandées pour l'étalonnage de différentes catégories de canaux de mesure - en raison de leurs caractéristiques statiques très diversifiées sur les données d'étalonnage entachées par les erreurs dans une échelle "réaliste".
- Pour les données entachées par les erreurs, la meilleure exactitude n'est pas fixée par une seule fonction spline et une seule procédure *PCR1* ou *PCR3*. Mais, nous pouvons trouver que les fonctions spline de lissage, incluant parabolique, cubique et *Beta-spline*, montrent souvent la meilleure exactitude; les fonctions spline paraboliques d'interpolation et de moindres carrés montrent souvent la pire exactitude.

Une évaluation des algorithmes a été réalisée au *chapitre 6*. Le choix de l'algorithme pour un problème de reconstitution de mesurandes ne peut se préciser que par l'application envisagée et le type de signal à reconstruire. Nous avons proposé une classe d'algorithmes qui répond à de nombreux problèmes non-linéaires et statiques rencontrés dans la pratique.

## 7.2. Contribution méthodologique

Cette contribution consiste essentiellement à proposer une approche à l'implantation en technologie de logiciel d'une classe d'algorithmes pour la reconstitution de mesurandes. Une méthodologie de développement est proposée aux *chapitres 3, 4, 5*.

Nous avons proposé d'étudier la réalisation d'algorithmes et leur vérification selon des critères d'évaluation qui correspondent à des cas pratiques. Essentiellement, le problème de l'implantation d'une classe d'algorithmes en technologie de logiciel consiste à:

- choisir les conditions aux limites et l'estimation des paramètres nécessaires selon les données synthétiques différentes rencontrées lors de l'étude des algorithmes;
- réaliser les algorithmes dans le domaine de logiciel utilisant les technologies de programmation;



- appliquer les fonctions spline à une variable sur les problèmes d'étalonnage et de reconstitution de mesurandes à deux variables, i.e., en présence d'une grandeur d'influence;
- utiliser les algorithmes d'optimisation pour choisir les paramètres lors de l'étude des algorithmes, i.e., la procédure d'optimisation avec la *L*-courbe.

### **7.3. Application des résultats des travaux**

Les algorithmes sont d'abord implantés sur MATLAB. Le traitement de matrices d'une façon implicite qu'offre ce logiciel, en fait l'outil idéal pour le développement d'un tel algorithme. Les algorithmes ont été développés en MATLAB au départ, afin de trouver une façon demandant la moindre mémoire possible, tout en étant rapide, universel et précis. Nous devons maintenant analyser le travail qui a été fait. Voici les éléments à l'étude:

- Modèle des systèmes de mesure;
- Filtre statique;
- MATLAB versus C;
- Signal: forme, erreur;
- Influence de  $\nu$ ;
- Effet de la condition aux limites;
- Influence d'erreur selon la variance pour une distribution normale;
- Calcul d'erreur de reconstitution;
- Temps de calcul.

Les résultats de cette recherche ont été directement mis à profit dans les travaux concernant le développement des structures de mesure intégrées pour les applications instrumentales au Laboratoire de systèmes de mesure à l'U.Q.T.R.

Ces développements algorithmiques sont également applicables dans d'autres domaines où le même type de traitement basé sur le modèle de la reconstitution est utile.

Les algorithmes développés de reconstitution de mesurandes satisfont les critères d'évaluation formulés. Suivant cette approche nous pourrions aussi résoudre plus efficacement d'autres problèmes.

Les résultats obtenus et présentés dans ce mémoire indiquent la pertinence de la poursuite de la recherche concernant le développement des algorithmes de reconstitution de mesurandes. La poursuite de cette recherche est importante, en particulier pour la suite des travaux sur les systèmes de mesure intégrés développés au Laboratoire de systèmes de mesure à l'U.Q.T.R.

Ce travail a été fait dans le cadre d'une recherche sur la possibilité d'intégration de systèmes de mesure. Nous maîtrisons bien tous les éléments faisant partie de l'étude d'un algorithme de reconstitution.

Nous avons beaucoup à gagner en implantant le système de reconstitution de façon statique: meilleure exactitude, moins de mémoire nécessaire, moins de temps de traitement.

## 7.4. Perspectives de développements futurs

### 7.4.1. Procédures des fonctions spline à deux variables

Dans ce projet, nous utilisons les fonctions spline à une variable pour l'approximation de caractéristiques statiques à trois dimensions. Plus de calculs peuvent causer plus d'erreurs dans le résultat final et prendre plus de temps. Dans la prochaine étape, nous allons développer des fonctions spline à deux variables pour l'étalonnage et la reconstitution de mesurandes.

### 7.4.2. Développement d'autres procédures d'approximation

Nous allons développer d'autres procédures d'approximation pour les systèmes de mesure. Ces procédures permettront une meilleure exactitude et une plus haute vitesse.

1) L'approximation basée sur la fonction de Gauss à deux variables:

$$\hat{x} = \sum_{k=1}^K p_k \exp \left[ -\frac{1}{2} \begin{bmatrix} y - \mu_{y,k} \\ v - \mu_{v,k} \end{bmatrix} \begin{bmatrix} -\sigma_{y,k}^2 & \zeta_k \\ \zeta_k & \sigma_{v,k}^2 \end{bmatrix} \begin{bmatrix} y - \mu_{y,k} \\ v - \mu_{v,k} \end{bmatrix} \right] \quad (7-1)$$

est utilisée pour l'approximation de la caractéristique inverse  $x = F(y, v)$ . Dans (7-1),  $\exp$  est la fonction exponentielle,  $p = [p_1 | p_2 | \dots | p_K]$  sont les paramètres à être estimés en se basant sur les données  $\hat{D}^{cal}$  ou  $\tilde{D}_r^{cal}$ , les paramètres  $\{\mu_{y,k}, \mu_{v,k}, \sigma_{y,k}^2, \sigma_{v,k}^2, \zeta_k \mid k=1, \dots, K\}$  sont choisis *a priori*, et  $\sigma_{y,k}^2 > 0, \sigma_{v,k}^2 > 0, \zeta_k^2 \leq \sigma_{y,k}^2 \sigma_{v,k}^2$ .

2) L'étalonnage statique basé sur la superposition des fonctions à une variable. Cette méthode est utilisée dans l'approximation des caractéristiques statiques des capteurs. L'inverse de la caractéristique statique  $x = F(y, v)$  est approximée par

$$\hat{x} = \hat{F}(y; \begin{bmatrix} p_1(v; q_1) \\ \vdots \\ p_K(v; q_K) \end{bmatrix}) \quad (7-2)$$

où,  $\hat{F}$  est supposée être une approximation de lissage, et  $p_i$  est une approximation

d'interpolation. Donc la procédure d'étalonnage est de la forme:

$$\hat{p}_i = \text{ParSmo} \left\{ y_{ij}^{cal}, x_{ij}^{cal} \mid j = 1, 2, \dots, NX^{cal} \right\} \quad (7-3)$$

pour  $i = 1, \dots, NV^{cal}$ , et

$$\hat{q}_k = \text{ParInt} \left\{ v_i^{cal}; \hat{p}_{i,k} \mid i = 1, 2, \dots, NV^{cal} \right\} \quad (7-4)$$

pour  $k = 1, \dots, K$ .

#### 7.4.3. Implantation matérielle

L'implantation va être faite dans un microprocesseur, cette étape nous permet de vérifier l'exactitude de reconstitution des mesurandes, le temps de traitement des données et l'espace de la mémoire nécessaire, une comparaison des résultats entre l'ordinateur et le microprocesseur avec la méthode basée sur les fonctions spline est proposée.

## ANNEXES

### A.1. Estimation du paramètre de lissage optimal par la méthode de L-courbe

#### A.1.1. Description générale de la méthode de L-courbe

Dans la section 3.3.2., nous avons la procédure pour l'approximation de lissage par une fonction spline cubique. Le paramètre de lissage affecte la qualité de la fonction spline. Le critère pour faire ce choix est un des lissages satisfaisants, i.e., l'erreur minimale de l'approximation atteinte.

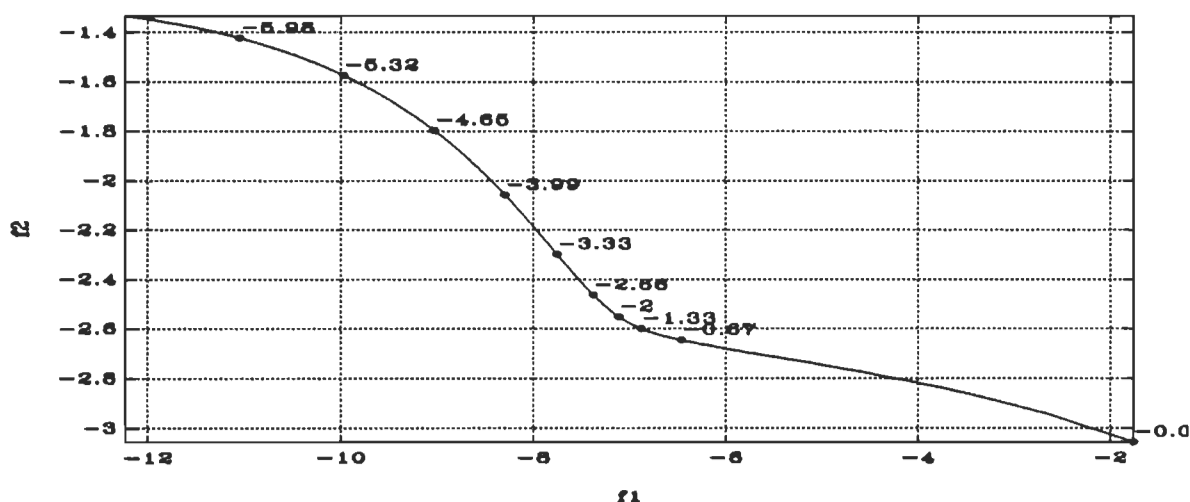


Fig.A-1. Une L-courbe exemplaire.

La fonction spline de lissage a pour principe de minimiser  $J_\alpha[p]$  pour un paramètre donné  $\alpha \in [0, 1]$ , ou correspondant à  $\alpha$ . La minimisation établit une sorte de compromis entre notre désir de rester proche des données fournies et notre désir d'obtenir une fonction de lissage. La valeur optimale du paramètre de lissage  $\alpha$  dépend de la forme de la fonction approximative et de l'espacement moyen des noeuds. Si nous considérons les conditions aux limites "naturelle" et l'exigence voulant que  $\hat{G}(x;p)$  ait une première dérivée continue, alors

$$J_\alpha[\mathbf{p}] = (1 - e^\alpha) \sum_{i=1}^N [y_i - \hat{G}_i(x_i; \mathbf{p})]^2 + \frac{4e^\alpha}{3} \sum_{i=1}^{N-1} \Delta x_i (\gamma_i^2 + \gamma_i \gamma_{i+1} + \gamma_{i+1}^2) \quad (\text{A-1})$$

où  $\gamma_i = \frac{d^2}{dx^2} \hat{G}(x; \mathbf{p}) \big|_{x=x_i}$  pour  $i = 1, \dots, N$ ; et  $\Delta x_i = x_{i+1} - x_i$ , pour  $i = 1, \dots, N-1$ .

La valeur optimale du paramètre de lissage  $s$  (ou  $\alpha$ ) dépend de plusieurs facteurs, tels que la forme de la courbe originale, le nombre de noeuds, le niveau d'erreur, etc. Nous voulons trouver le paramètre de lissage optimal automatiquement.

Les fonctions  $f_1(\alpha)$  et  $f_2(\alpha)$  définissent la  $L$ -courbe pour toutes les valeurs valides du paramètre de lissage  $s$ . La  $L$ -courbe affiche clairement le compromis entre la minimisation de ces deux quantités. Pour faciliter le calcul, nous utilisons la variable  $\alpha = \ln(1 - s)$  au lieu de  $s = 1 - e^\alpha$ , où  $\alpha \in (-\infty \sim 0]$ .

Quand  $\alpha$  varie de 0 à  $-\infty$ , nous obtenons la courbe  $f_2 = L(f_1)$  montrée à la Fig.A-1. Le paramètre de lissage optimal  $\alpha_{opt}$  peut être trouvé au coin de la  $L$ -courbe. Nous étudierons et comparerons plusieurs méthodes pour localiser ce coin.

#### A.1.2. Méthode pour l'estimation de $\alpha_{opt}$ et sa vérification

##### (1) MALFA1

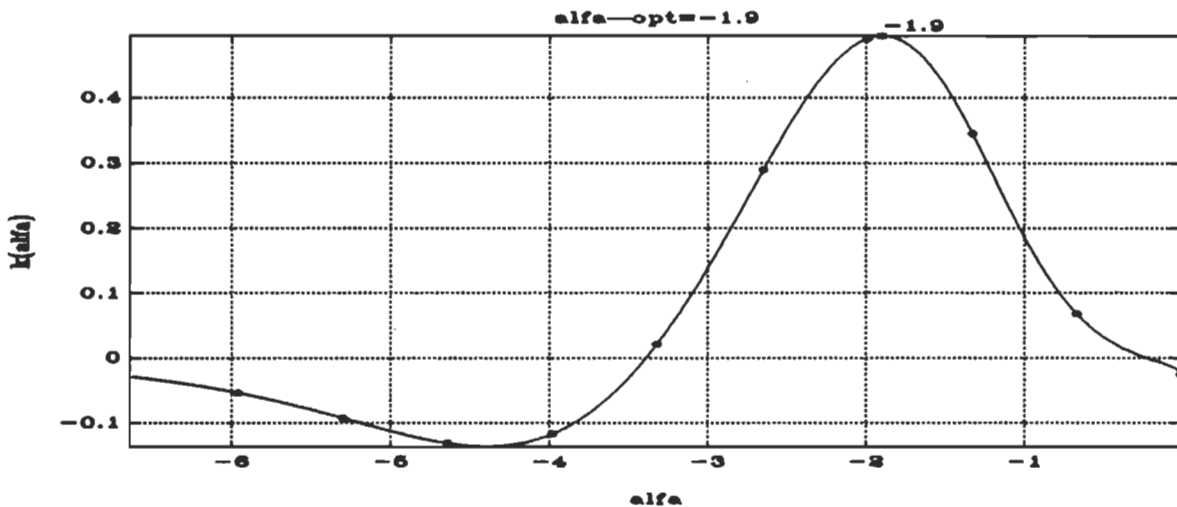


Fig.A-2. MALFA1:  $k(\alpha)$  et  $\alpha_{opt}^1 = -1.9$  pour  $L$ -courbe montrée à la Fig.A-1.

La première méthode, MALFA1, est la base de la notion de la courbure définie par

$$k(\alpha) = \frac{\frac{d^2 f_2(\alpha)}{df_1^2(\alpha)}}{\left[ 1 + \left( \frac{df_2(\alpha)}{df_1(\alpha)} \right)^2 \right]^{\frac{3}{2}}} \tag{A-2}$$

Cette méthode consiste en maximisation de  $k(\alpha)$ :

$$\alpha_{opt}^I = \arg_{\alpha} \sup \{ k(\alpha) \mid \alpha \in (-\infty, 0] \} \tag{A-3}$$

Les dérivées dans (A-2) sont calculées numériquement d'après les formules

$$\frac{df_2(\alpha)}{df_1(\alpha)} \Big|_{\alpha=\alpha_i} \equiv \frac{f_2(\alpha_{i+1}) - f_2(\alpha_i)}{f_1(\alpha_{i+1}) - f_1(\alpha_i)} \tag{A-4}$$

$$\frac{d^2 f_2(\alpha)}{df_1^2(\alpha)} \Big|_{\alpha=\alpha_i} \equiv \frac{f_2(\alpha_{i+1}) - 2f_2(\alpha_i) + f_2(\alpha_{i-1}))}{f_1(\alpha_{i+1}) - 2f_1(\alpha_i) + f_1(\alpha_{i-1}))} \tag{A-5}$$

pour  $i = 1, 2, \dots$ . Le résultat est montré sur la Fig.A-2.

(2) MALFA2

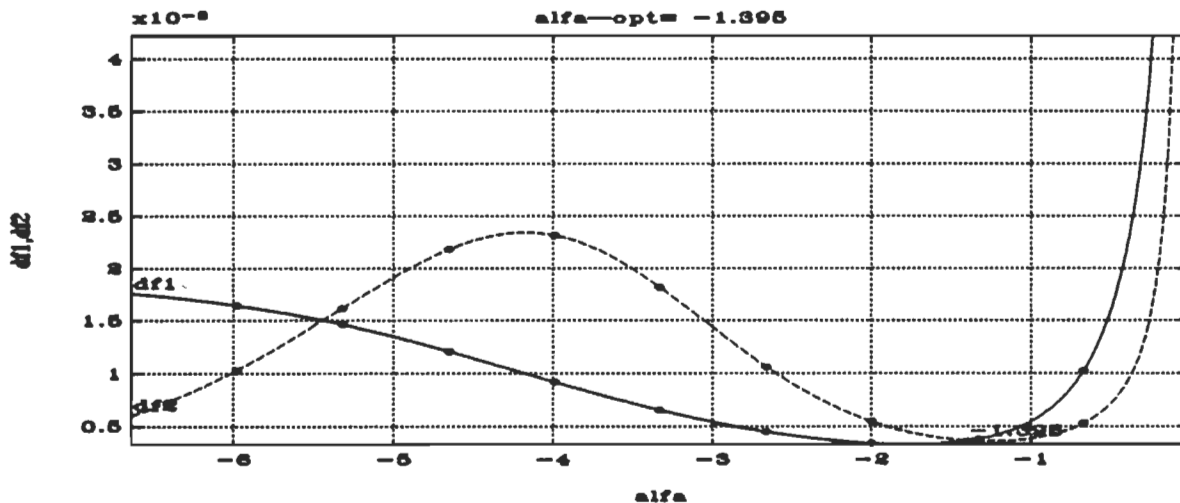


Fig.A-3. MALFA2:  $df_1(\alpha)$ ,  $df_2(\alpha)$  et  $\alpha_{opt}^II = -1.395$  pour L-courbe montrée à la Fig.A-1.

La deuxième méthode, MALFA2, consiste en comparaison des dérivées normalisées des  $f_1(\alpha)$  et  $f_2(\alpha)$  en respectant  $\alpha$ :

$$\alpha_{opt}^{II} = \arg_{\alpha} \left\{ \frac{f_1'(\alpha)}{\Delta F_1} = \frac{f_2'(\alpha)}{\Delta F_2} \mid \alpha \in (-\infty, 0] \right\} \tag{A-6}$$

où  $\Delta F_1$  et  $\Delta F_2$  sont les rangées des valeurs des  $f_1(\alpha)$  et  $f_2(\alpha)$ , respectivement, i.e.:

$$\Delta F_k = \sup\{f_k(\alpha) \mid \alpha \in (-\infty, 0]\} - \inf\{f_k(\alpha) \mid \alpha \in (-\infty, 0]\} \quad (A-7)$$

pour  $k=1, 2$ . Les dérivées,  $f'_1(\alpha)$  et  $f'_2(\alpha)$ , sont calculées numériquement:

$$f'_k(\alpha_i) \cong \frac{f_k(\alpha_{i+1}) - f_k(\alpha_i)}{\alpha_{i+1} - \alpha_i} \quad \text{pour } k=1, 2 \quad (A-8)$$

La valeur à droite  $\alpha_{opt}^II$  est sélectionnée parmi les solutions possibles de  $L$ -courbe comprises dans (A-6) de façon à ce que  $f'_1(\alpha_{opt}^II) > 0$  et  $f'_2(\alpha_{opt}^II) < 0$ . Le résultat est montré sur la Fig.A-3.

(3) MALFA3

La troisième méthode, MALFA3, consiste en maximisation de la deuxième dérivée de la  $L$ -courbe:

$$\alpha_{opt}^III = \arg_{\alpha} \sup \left\{ \frac{d^2 f_2(\alpha)}{d f_1^2(\alpha)} \mid \alpha \in (-\infty, 0] \right\} \quad (A-9)$$

où la deuxième dérivée est calculée numériquement d'après Éq.(A-5). Le résultat est montré sur la Fig.A-4.

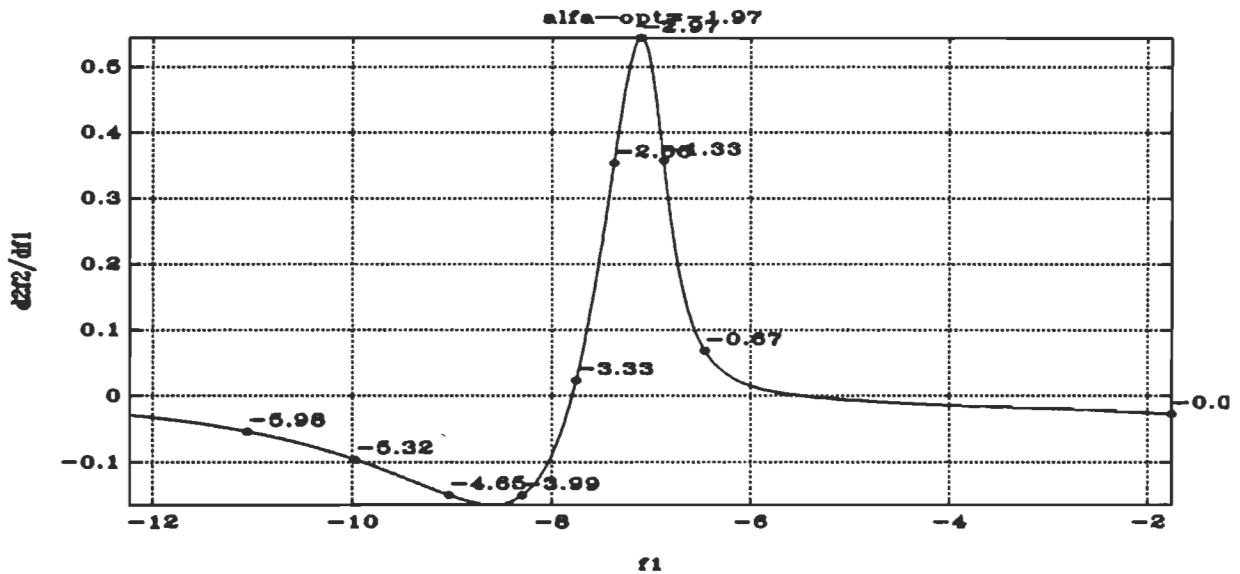


Fig.A-4. MALFA3: la deuxième dérivée de la  $L$ -courbe et  $\alpha_{opt}^III = -1.97$  pour la  $L$ -courbe montrée à la Fig.A-1.

(4) MALFA4

La quatrième méthode, MALFA4, est analytiquement équivalente à MALFA3 mais

numériquement différente:

$$\alpha_{opt}^{IV} = \arg_{\alpha} \sup \left\{ \frac{d^3 f_2(\alpha)}{d\alpha^3} = 0 \mid \alpha \in (-\infty, 0] \right\} \quad (A-10)$$

où la troisième dérivée de L-courbe est calculée numériquement d'après la formule:

$$\frac{d^3 f_2(\alpha)}{d\alpha^3} \Big|_{\alpha=\alpha_i} \cong \frac{f_2(\alpha_{i+2}) - 3f_2(\alpha_{i+1}) + 3f_2(\alpha_i) - f_2(\alpha_{i-1}))}{f_1(\alpha_{i+2}) - 3f_1(\alpha_{i+1}) + 3f_1(\alpha_i) - f_1(\alpha_{i-1}))} \quad (A-11)$$

Le résultat est montré sur la Fig.A-5.

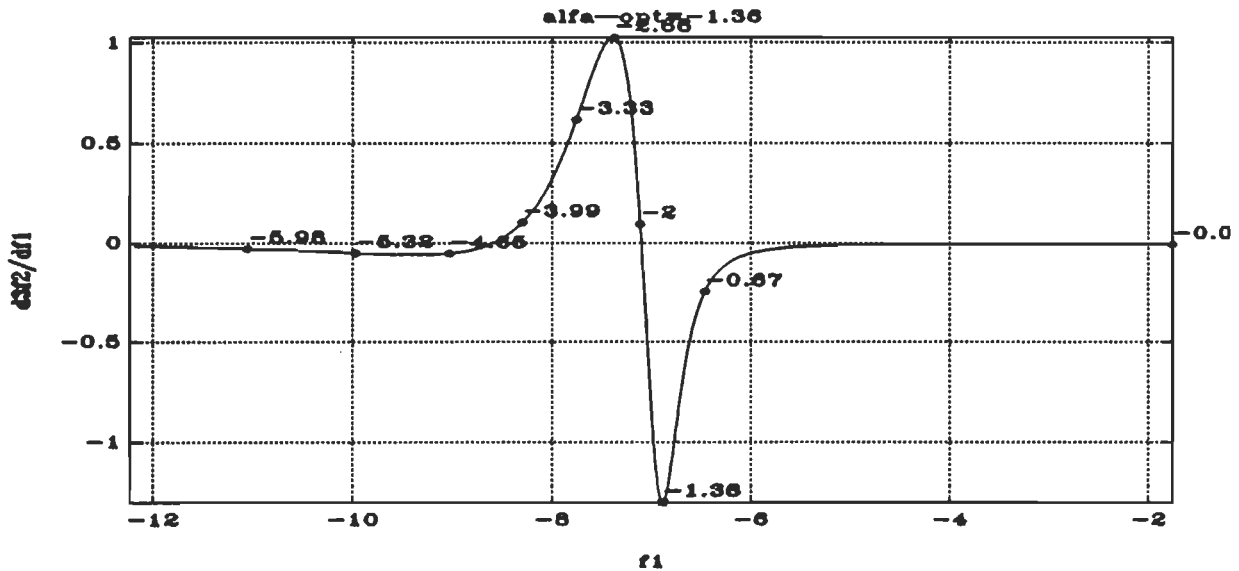


Fig.A-5. MALFA4: la troisième dérivée de la L-courbe et  $\alpha_{opt}^{IV} = -1.36$  pour la L-courbe montrée à la Fig.A-1.

(5) La valeur de référence de  $\alpha_{opt}$

La valeur de référence de  $\alpha_{opt}$  est trouvée en utilisant les données synthétiques produites par la façon suivante:

$$x_j^{ref} = \begin{cases} x_{(i+1)/2} & \text{pour } i = 1, 3, \dots, 2N-1 \\ \frac{x_{i/2} + x_{i/2+1}}{2} & \text{pour } i = 2, 4, \dots, 2N-2 \end{cases} \quad (A-12)$$

$$y_j^{ref} = G(x_j^{ref}) \quad \text{pour } j = 1, 2, \dots, 2N-1 \quad (A-13)$$

où,  $G(\cdot)$  est une fonction à être rapprochée. L'erreur quadratique moyenne absolue d'approximation est définie comme:

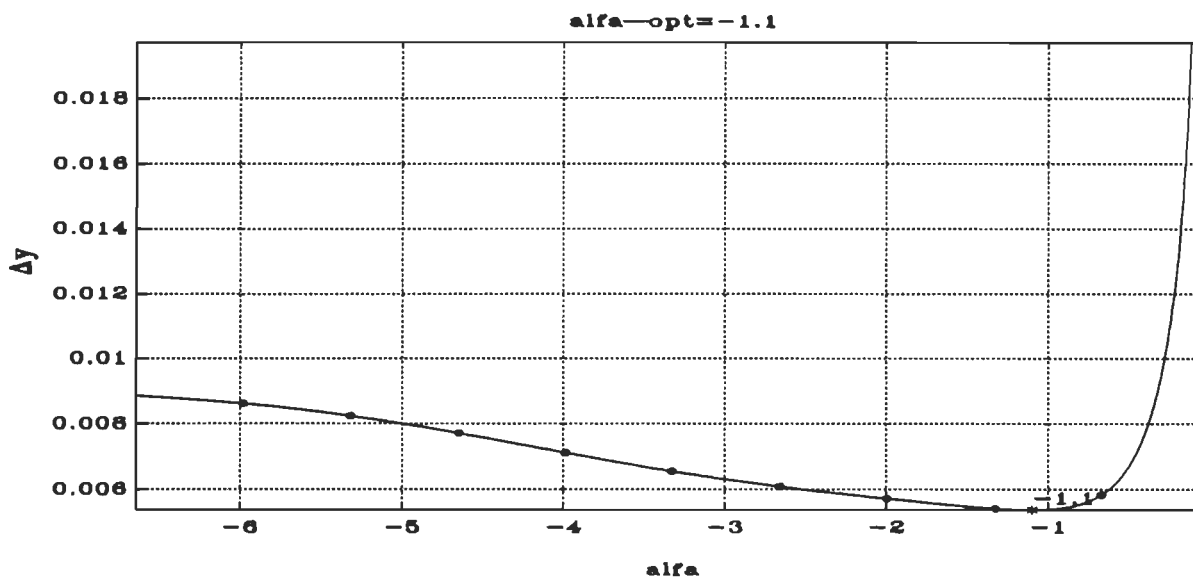


$$\Delta y(\alpha) = \left\{ \frac{1}{2N-1} \sum_{j=1}^{2N-1} [\hat{G}(x_j^{ref}; \hat{\rho}_\alpha) - y_j^{ref}]^2 \right\}^{\frac{1}{2}} \quad (\text{A-14})$$

Donc, la valeur de référence de  $\alpha_{opt}$  est

$$\alpha_{opt} = \arg_{\alpha} \inf \{ \Delta y(\alpha) \mid \alpha \in (-\infty, 0] \} \quad (\text{A-15})$$

Le résultat est montré sur la Fig.A-6.



**Fig.A-6.** L'erreur quadratique moyenne absolue d'estimé  $\Delta y$  versus  $\alpha$  correspondant à la L-courbe montrée à la Fig.A-1,  $\alpha_{opt} = -1.1$ .

Le tableau A-1 montre la comparaison des valeurs  $\alpha$  des méthodes différentes. La valeur dans les cadres à lignes doubles est la plus exacte correspondant à la valeur  $\alpha_{opt}$ . La valeur avec l'astérisque a l'erreur maximale entre la valeur  $\alpha_{opt}$  et cette valeur.

Tableau A-1. Comparaison des valeurs  $\alpha$  des méthodes différentes

Les cas pour le test	$\nu$	$N_x$	$\alpha_{i+1} - \alpha_i$	$\alpha_{opt}$	$\alpha_{opt}^I$ $k(\alpha)$	$\alpha_{opt}^{II}$ $\{df_1 \sim df_2\}$	$\alpha_{opt}^{III}$ $d^2f_2/df_1^2(\alpha)$	$\alpha_{opt}^{IV}$ $d^3f_2/df_1^3(\alpha)$
CAS1	0	15	-0.001	-1.333	-0.823°	-0.9121	-0.823°	-1.387
		30	-0.01	-1.1	-1.9	-1.395	-1.97°	-1.36
		50	-0.01	-1.46	-1.86	-1.335	-2.29°	-1.45
	20	20	-0.001	-2.117	-1.77	-1.663	-1.77	-1.444°
		30	-0.001	-2.06	-1.77	-1.622	-1.77	-1.22°
		50	-0.01	-1.63	-2.27	-2.493°	-2.45	-1.61
CAS2	0	15	-0.001	-4.789	-5.259	-4.911	-5.294	-6.093°
		30	-0.001	-5.044	-5.697	-5.32	-6.138°	-5.153
		50	-0.01	-4.68	-4.94	-4.757	-6.22°	-5.28
	25	20	-0.01	-4.97	-5.35	-5.149	-5.55°	-4.8
		30	-0.01	-4.06	-5.39	-4.206	-5.55°	-4.91
		50	-0.01	-5.33	-4.92	-4.751	-6.22°	-4.81
CAS3	7	20	-0.1	-22.3	-22.7	-21.16	-22.8	-20.2°
		30	-0.1	-21.5	-24.9°	-21.4	-24.9°	-20.2
		50	-0.1	-20.6	-20.6	-20.87°	-20.6	-20.5
	8.5	15	-0.1	-19.2	-21.7°	-19.25	-21.7°	-18.8
		30	-0.1	-18.9	-18.6	-19.11	-18.6	-19.8°
		50	-0.1	-19.	-19.	-19.72	-19.1	-18.1°

## A.2. Procédures d'approximation par les fonctions spline

### A.2.1. ParIntSplP1

```
function [Pv] = ParIntSplP1(xk,yk,choice)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% returns the parameters Pv
% of the parabolic interpolation spline
% according to (xk,yk).
% choice = 1: dy/dx=0 at end point
% choice = 2: d2y/dx2=0 at end point
% choice = 3: "periodical function" at end point
% choice = 4: specified slope (y2-y1)/(x2-x1) at end point
% choice = 5: specified slope at end point with Langage interpolation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n=length(xk);
[xi,ind]=sort(xk);xi=xi(:);xk=xk(ind);
yi=yk(ind);yi=yi(:);yk=yk(ind);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if choice == 1,
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k=1:n-1,
    if k == 1,
        my=[0 yk(1) yk(2)]';
        mx=[2*xk(k) 1 0
             xk(k)^2 xk(k) 1
             xk(k+1)^2 xk(k+1) 1];
        mp=inv(mx)*my;
    elseif k>1 & k<n,
        my=[2*a(k-1)*xk(k)+b(k-1) yk(k) yk(k+1)]';
        mx=[2*xk(k) 1 0
             xk(k)^2 xk(k) 1
             xk(k+1)^2 xk(k+1) 1];
        mp=inv(mx)*my;
    end %if
    a(k)=mp(1);
    b(k)=mp(2);
    c(k)=mp(3);
end %k
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
elseif choice == 2,
for k=1:n-1,
    if k == 1,
        my=[0 yk(1) yk(2)]';
        mx=[1 0 0
```

```

    0 xk(1) 1
    0 xk(2) 1];
    mp = inv(mx) * my;
elseif k > 1 & k < n,
    my = [2 * a(k-1) * xk(k) + b(k-1) yk(k) yk(k + 1)]';
    mx = [2 * xk(k) 1 0
          xk(k)^2 xk(k) 1
          xk(k + 1)^2 xk(k + 1) 1];
    mp = inv(mx) * my;
end %if
a(k) = mp(1);
b(k) = mp(2);
c(k) = mp(3);
end %k
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
elseif choice == 3,
    for k = 1:n,
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        if k == 1,
            my = [0 yk(n) yk(1)]';
            mx = [1 0 0
                  xk(n)^2 xk(n) 1
                  xk(1)^2 xk(1) 1];
            mp = inv(mx) * my;
        elseif k > 1 & k < n + 2,
            my = [2 * a(k-1) * xk(k-1) + b(k-1) yk(k-1) yk(k)]';
            mx = [2 * xk(k-1) 1 0
                  xk(k-1)^2 xk(k-1) 1
                  xk(k)^2 xk(k) 1];
            mp = inv(mx) * my;
        end %if
        a(k) = mp(1);
        b(k) = mp(2);
        c(k) = mp(3);
    end %k
    am(1:n-1) = a(2:n);
    bm(1:n-1) = b(2:n);
    cm(1:n-1) = c(2:n);
    clear a b c;
    a = am;
    b = bm;
    c = cm;
    clear am bm cm;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
elseif choice == 4,
    for k = 1:n-1,
        if k == 1,
            my = [(yk(2)-yk(1))/(xk(2)-xk(1)) yk(1) yk(2)]';
            mx = [2 * xk(1) 1 0
                  xk(1)^2 xk(1) 1

```

```

    (xk(2))^2 xk(2) 1];
    mp = inv(mx)*my;
elseif k > 1 & k < n,
    my = [2*a(k-1)*xk(k) + b(k-1) yk(k) yk(k+1)]';
    mx = [2*xk(k) 1 0
          xk(k)^2 xk(k) 1
          xk(k+1)^2 xk(k+1) 1];
    mp = inv(mx)*my;
end %if k
a(k) = mp(1);
b(k) = mp(2);
c(k) = mp(3);
end %k
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
elseif choice == 5,
    dx12 = xk(1)-xk(2);
    dx13 = xk(1)-xk(3);
    dx14 = xk(1)-xk(4);
    dx23 = xk(2)-xk(3);
    dx24 = xk(2)-xk(4);
    dx34 = xk(3)-xk(4);
    k1 = (dx13*dx14 + dx12*dx14 + dx12*dx13)/(dx12*dx13*dx14)*yk(1);
    k2 = (dx13*dx14)/(-dx12*dx23*dx24)*yk(2);
    k3 = (dx12*dx14)/(-dx13*(-dx23)*dx34)*yk(3);
    k4 = (dx12*dx13)/(-dx14*(-dx24)*(-dx34))*yk(4);
    y11 = k1 + k2 + k3 + k4;
    b(1) = y11;
    for k = 1:n-1,
        dt(k) = xk(k+1)-xk(k);
        s(k) = (yk(k+1)-yk(k))/dt(k);
        a(k) = (s(k)-b(k))/dt(k);
        c(k) = yk(k);
        if k < (n-1),
            b(k+1) = 2*s(k)-b(k);
        end %if
    end %k
    av = a;
    bv = -2*a.*xk(1:n-1) + b;
    cv = a.*(xk(1:n-1)).^2 - b.*xk(1:n-1) + c;
    clear a b c;
    a = av;
    b = bv;
    c = cv;
    clear av bv cv;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end %if choice
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Pv = [a
      b
      c];

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
return

```

### A.2.2. ParLmsSplP1

```

function [Pd] = parlmsp1(xk,yk)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% returns the values at xr of the parabolic
% least-square spline for the sample data
% (xk,yk).
% xk, yk: samples (knots)
% d2y/dx2 = 0 at end point "natural"
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% parlmsp1.m
% 1995.11.1.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xknum=length(xk);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
save temp1.mat xk yk;
a0=zeros(size(xk));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% least squares method
% to solve min problem
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
a_opt = fminu('eqlmsp1',a0,OPTIONS);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
p0=a_opt;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k=1:xknum-1,
    if k == 1,
        p2(k)=0;
        p1(k)=(p0(k+1)-p0(k))/(xk(k+1)-xk(k));
    else,
        p1(k)=p1(k-1)+2*p2(k-1)*(xk(k)-xk(k-1));
        p2(k)=(p0(k+1)-p0(k)-p1(k)*(xk(k+1)-xk(k)))/(xk(k+1)-xk(k))^2;
    end %if
end %k
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[n01 n02]=size(p0);
if n02 == 1,
    p0=p0';
end %if
Pd=[p0(1:xknum-1)
    p1
    p2];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
return

```

```

function Jmin = equilmsp1(a)
%%%%%%%%%%%%%%
% called by parlmsp1.m
%%%%%%%%%%%%%%
load temp1.mat;
%%%%%%%%%%%%%%
[a1 a2] = size(a);
[y1 y2] = size(yk);
if a1 ~= y1,
    a = a';
end %if
J_v = (a-yk).^2;
%%%%%%%%%%%%%%
    Jmin = sum(J_v);
%%%%%%%%%%%%%%
return

```

### A.2.3. *ParSmoSplP1*

```

function [Pd] = ParSmoSplP1(xk,yk,SigY,Sp)
%%%%%%%%%%%%%%
% returns the value Pd according to xk, yk
% of parabolic smoothing spline
% SigY: variance of yk
% Sp: smoothing parameter
%%%%%%%%%%%%%%
n=length(xk);
xnum = n;
[xi,ind] = sort(xk);xi = xi(:);xk = xi';
yi = yk(ind);yi = yi(:);yk = yi';
%%%%%%%%%%%%%%
% initiale parameters
%%%%%%%%%%%%%%
    der1 = der1cal(xk(1),xk(2),xk(3),xk(4),yk(1),yk(2),yk(3),yk(4));
    b(1) = der1;
    for ki = 1:xnum-1,
        dx(ki) = xk(ki + 1)-xk(ki);
        c(ki) = yk(ki);
    end %ki
    c(xnum) = yk(xnum);
    for ki = 1:xnum-1,
        a(ki) = (c(ki + 1)-c(ki))/dx(ki)-b(ki);
        b(ki + 1) = b(ki) + 2*a(ki);
        a(ki) = a(ki)/dx(ki);
    end %ki
%%%%%%%%%%%%%%
% valeur initial of norme L2 of the second derive of spline
%%%%%%%%%%%%%%

```

```

J = 0;
for ki = 1:xnum-1,
    J = J + a(ki)^2 * dx(ki);
end %ki
J = J/2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% value initial of parameters of iteration
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
w0 = -Sp * SigY^2;
w1 = 0;
w2 = 0;
La = 0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k1 = 1;
k2 = 0;
while ~ (k1 < k2),
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    db1 = 0;
    for ki = 1:xnum-1,
        if ki/2 ~ = fix(ki/2),
            db1 = db1 - a(ki);
        else,
            db1 = db1 + a(ki);
        end %if
    end %ki
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    dq(xnum-1) = a(xnum-1);
    for ki = xnum-2:-1:2,
        dq(ki) = a(ki) - a(ki + 1) - dq(ki + 1);
    end %ki
    dq(1) = a(1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    da = dq(1) - db1;
    sn = da * a(1);
    sd = (da^2) / dx(1);
    for ki = 2:xnum-1,
        da = dq(ki) - dq(ki-1) - da;
        sn = sn + da * a(ki);
        sd = sd + (da^2) / dx(ki);
    end %ki
    La0 = -sn / sd;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    dc(1) = 0;
    for ki = 1:xnum-1,
        dc(1) = dc(1) - (1 - ki / xnum) * dx(ki) * dq(ki);
    end %ki
    for ki = 2:xnum,
        dc(ki) = dc(ki-1) + dq(ki-1) * dx(ki-1);
    end %ki
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```



```

% resolution of equation quadratic with La1 and La2 respects
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
w0 = w0 + La*(La*w2 + 2*w1);
w1 = 0;
w2 = 0;
for ki = 1:xnum,
    w1 = w1 + dc(ki)*(c(ki)-yk(ki));
    w2 = w2 + dc(ki)^2;
end %ki
wsign = wsigncal(w1);
La2 = (-w1-wsign*sqrt(w1^2-w0*w2))/w2;
if La2 == 0,
    La1 = 0;
else,
    La1 = w0/w2/La2;
end %if
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% choice among La0, La1, La2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if La1 > La2,
    La = La1;
    La1 = La2;
    La2 = La;
end %if
if La0 < 0,
    La = 0.9999*max([La1,La0]);
else,
    La = 0.9999*min([La0,La2]);
end %if
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for ki = 1:xnum,
    c(ki) = c(ki) + La*dc(ki);
end %ki
for ki = 1:xnum-1,
    a(ki) = (c(ki+1)-c(ki))/dx(ki)-b(ki);
    b(ki+1) = b(ki) + 2*a(ki);
    a(ki) = a(ki)/dx(ki);
end %ki
k1 = abs(La*(sn + La*sd/2));
k2 = 0.001*J;
end %while
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% new J is less than 0.1% of its initial valur
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Pd(1,1:xnum-1) = a;
Pd(2,1:xnum-1) = b(1,1:xnum-1);
Pd(3,1:xnum-1) = c(1,1:xnum-1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
return

```

```

function wsign = wsigncal(w)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% returns the sign of variable w: 1 or -1
% called by ParSmoSplP1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if w > 0,
    wsign = 1;
else
    wsign = -1;
end %if
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
return

```

#### A.2.4. ValSplP1

```

function [yest] = ParValSplP1(xk,xdat,Par,choice)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% returns the value yest at xdat
% according to Par and (xk,yk)
% yest: results
% xk, yk: samples (knots)
% choice: option of the format of parameter Par
% choice = 1: yest = pv1 * xdat^2 + pv2 * xdat + pv3
% choice = 2: yest = ai * dx^2 + bi * dx + ci (dx = xdat - xk)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n = length(xk);
[xi,ind] = sort(xk);xi = xi(:);xk = xi';
yi = yk(ind);yi = yi(:);yk = yi';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if choice == 1,
    av = Par(1,1:n-1);
    bv = Par(2,1:n-1);
    cv = Par(3,1:n-1);
elseif choice == 2,
    a = Par(1,1:n-1);
    b = Par(2,1:n-1);
    c = Par(3,1:n-1);
end %choice
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for ki = 1:length(xdat),
    xt = xr(ki);
    il = 1;
    ih = n;
    while ih > il + 1,
        it = fix((il + ih)/2);
        if xk(it) > xt,
            ih = it;
        else

```

```

    il = it;
end %if
if il = ih,
    if il = n,
        il = il-1;
    else
        ih = ih + 1;
    end %if
end %if
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if choice == 1,
    yt = av(il)*xt^2 + bv(il)*xt + cv(il);
    ytd = 2*av(il)*xt + bv(il);
elseif choice == 2,
    dx = xt-xk(il);
    yt = (a(il)*dx + b(il))*dx + c(il);
    ytd = 2*a(il)*dx + b(il);
end %choice
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end %while
yest(ki) = yt;
yestd(ki) = ytd;
end %ki
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
return

```

### A.2.5. *ParIntSplC1*

```

function [Pd] = ParIntSplC1(xk,yk)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% returns the parameters Pd
% according to xk, yk
% xk, yk: samples (knots)
% end condition: natural
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n = length(xk);
[xi,ind] = sort(xk);xi = xi(:);xk = xi';
yi = yk(ind);yi = yi(:);yk = yi';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
h = (xk(n)-xk(1))/(n-1);
h2 = h^2;
h3 = h^3;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for it = 1:n-1,
    if it == 1,
        c1 = (3*yk(1)-3*yk(2) + yk(3))/2;
        c2 = (yk(it))/2;
        c3 = (yk(it + 1))/2;
    end
end

```

```

    c4 = (yk(it + 2))/2;
elseif it == n-1,
    c1 = (yk(it-1))/2;
    c2 = (yk(it))/2;
    c3 = (yk(it + 1))/2;
    c4 = (3*yk(n)-3*yk(n-1) + yk(n-2))/2;
else,
    c1 = (yk(it-1))/2;
    c2 = (yk(it))/2;
    c3 = (yk(it + 1))/2;
    c4 = (yk(it + 2))/2;
end %if
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
a = [a (-c1 + 3*c2 - 3*c3 + c4)/h3];
b = [b (2*c1 - 5*c2 + 4*c3 - c4)/h2];
c = [c (-c1 + c3)/h];
d = [d 2*c2];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end %it
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Pd = [a
      b
      c
      d];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
return

```

### A.2.6. *ParLmsSplC1*

```

function [Pd] = parlmsc1(xk,yk)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% returns the values at xr of the cubic
% least-square spline for the sample data
% (xk,yk).
% xk, yk: samples (knots)
% d2y/dx2 = 0 at end point "natural"
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% parlmsp1.m
% 1995.11.1.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xknum = length(xk);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
save temp1.mat xk yk;
a0 = zeros(size(xk));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% least squares method
% to solve min problem
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

a_opt = fminu('equilmisc1',a0,OPTIONS);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
p0 = a_opt;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Am = zeros(xknum,xknum);
Bv = zeros(xknum,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for ki = 1:xknum,
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if ki == 1,
        Am(ki,1) = 2*(xk(2)-xk(1));
        Am(ki,2) = xk(2)-xk(1);
        Bv(ki) = 3*(p0(2)-p0(1));
    elseif ki == xknum,
        Am(ki,xknum) = 2*(xk(xknum)-xk(xknum-1));
        Am(ki,xknum-1) = xk(xknum)-xk(xknum-1);
        Bv(ki) = 3*(p0(xknum)-p0(xknum-1));
    elseif ki ~ = 1 & ki ~ = xknum,
        Am(ki,ki-1) = xk(ki+1)-xk(ki);
        Am(ki,ki) = 4*(xk(ki+1)-xk(ki));
        Am(ki,ki+1) = xk(ki+1)-xk(ki);
        Bv(ki) = 3*(p0(ki+1)-p0(ki-1));
    end %if
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end %ki
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Xv = inv(Am)*Bv;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
p1 = Xv';
for k = 1:xknum-1,
    p2(k) = (3*(p0(k+1)-p0(k))-2*Xv(k)*(xk(k+1)-xk(k))-Xv(k+1)*(xk(k+1)-xk(k)))/((xk(k+1)-xk(k))^2);
    p3(k) = (-2*(p0(k+1)-p0(k)) + Xv(k)*(xk(k+1)-xk(k)) + Xv(k+1)*(xk(k+1)-xk(k)))/((xk(k+1)-xk(k))^3);
end %k
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[n1 n2] = size(p0);
if n1 ~ = 1,
    p0 = p0';
end %if
[n1 n2] = size(p1);
if n1 ~ = 1,
    p1 = p1';
end %if
[n1 n2] = size(p2);
if n1 ~ = 1,
    p2 = p2';
end %if
[n1 n2] = size(p3);
if n1 ~ = 1,
    p3 = p3';
end %if

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Pd = [p3
      p2
      p1(1:xknum-1)
      p0(1:xknum-1)];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
return

```

```

function Jmin = equimsc1(a)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% called by parimsc1.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load temp1.mat;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[a1 a2] = size(a);
[y1 y2] = size(yk);
if a1 ~ = y1,
    a = a';
end %if
J_v = (a-yk).^2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Jmin = sum(J_v);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
return

```

### A.2.7. *ParSmoSplC1*

```

function [Pv_opt, Pd_opt, alfa_opt, p_opt] = ParSmoSplC1(xk,yk,alfa_min,alfa_max,alfa_st)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% smopar3p
% choice of optimal smoothing parameter alfa
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
alfa_opt = 0;
p_opt = 0;
f1_v = [];
f2_v = [];
xkn = length(xk);
alfa_pre = (alfa_max + alfa_min)/2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
alfa_st = abs(alfa_st);
alfa_opt = 0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
while alfa_opt >= 0,
    [alfa_opt] = paralfa2(xk,yk,alfa_max,alfa_min,alfa_st);
end %while
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
p_opt = 1-exp(alfa_opt);

```

```
[Pv_opt,Pd_opt]=smospl3s(xk,yk,p_opt);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
return
```

```
function [Pv,Pd]=smospl3s(xk,yk,Sp)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% [Pv,Pd]=smospl3s(xk,yk,Sp)
% 1:Pv
% 2:Pd
% returns the parameters Pv or Pd
% of the cubic smoothing spline
% for the sample data (xk,yk)
% the 'natural' cubic smoothing spline
% from csaps.m
% xk, yk: samples (knots)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n=length(xk);[xind,ind]=sort(xk);xind=xind(:);
yind=yk(ind);yind=yind(:);yk=yind;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
output = csaps(xk,yk,Sp);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
a=output(5+n:2*n+3);
b=output(2*n+4:3*n+2);
c=output(3*n+3:4*n+1);
d=output(4*n+2:5*n);
Pd=[a
    b
    c
    d];
Pv=[a
    b-3*a.*xk(1:n-1)
    3*a.*(xk(1:n-1)).^2-2*b.*xk(1:n-1)+c
    -a.*(xk(1:n-1)).^3+b.*(xk(1:n-1)).^2-c.*xk(1:n-1)+d];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
return
```

```
function [alfa_opt]=paralfa2(xk,yk,alfa1,alfa2,alfa_st)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% called by smopar3p.m
% find optimal parameter alfa: alfa_opt
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
save temp1.mat xk yk alfa1 alfa2 alfa_st;
alfa_opt=fmin('alfaequ2',alfa1,alfa2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
return
```

```

function kf = alfaequ2(alfa)
%%%%%%%%%%%%%%
% call by paralfa2.m
%%%%%%%%%%%%%%
load temp1.mat; %xk yk alfa_st
%%%%%%%%%%%%%%
% calculate K
%%%%%%%%%%%%%%
p_s = 1-exp(alfa);
[Pv,Pd] = smospl3s(xk,yk,p_s);
[f1] = f1p(xk,yk,Pv);
f2 = f2p(xk,Pv);
%%%%%%%%%%%%%%
p_s = 1-exp(alfa-alfa_st);
[Pv,Pd] = smospl3s(xk,yk,p_s);
[f1_1] = f1p(xk,yk,Pv);
[f2_1] = f2p(xk,Pv);
%%%%%%%%%%%%%%
p_s = 1-exp(alfa + alfa_st);
[Pv,Pd] = smospl3s(xk,yk,p_s);
[f1_2] = f1p(xk,yk,Pv);
f2_2 = f2p(xk,Pv);
%%%%%%%%%%%%%%
der1 = (f2_2-f2_1)/(f1_2-f1_1);
der1n = length(der1);
der2 = (f2_1-2*f2 + f2_2)/(f1_2-f1_1)^2;
der2n = length(der2);
%%%%%%%%%%%%%%
kf = -(der2./(1 + der1(1:der2n).^2).^2).^2);
%%%%%%%%%%%%%%
clear f1 f2 f1_1 f2_1 f1_2 f2_2 der2 der1;
%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
return

```

```

function [f1pv] = f1p(xk,yk,Pv)
%%%%%%%%%%%%%%
% calculate f1(p)
% calculate optimal p
% called by alfaequ2.m
%%%%%%%%%%%%%%
n = length(xk);
eflag1 = 0;
f1pv = 0;
%%%%%%%%%%%%%%
xdat = xk;
    av = Pv(1,1:n-1);
    bv = Pv(2,1:n-1);
    cv = Pv(3,1:n-1);

```



```

    dv = Pv(4,1:n-1);
    %%%%%%%%%%
    [yest] = ParValSplC1(xk,xdat,Par,1);
    f1pv = sum((yk-yest).^2);
    if f1pv == 0 | f1pv < 0,
        yk;
        yest;
        f1pv;
    else,
        f1pv = log(f1pv);
    end %f1pv
    %%%%%%%%%%
    return

```

```

function f2pv = f2p(xk,Pv)
    %%%%%%%%%%
    % calculate f2(p)
    % calculate optimal p
    % called by alfaequ2.m
    %%%%%%%%%%
    f2pv = 0;
    xdat = xk;
    bet = zeros(1,length(xdat)-1);
    %%%%%%%%%%
    av = Pv(1,1:n-1);
    bv = Pv(2,1:n-1);
    cv = Pv(3,1:n-1);
    dv = Pv(4,1:n-1);
    %%%%%%%%%%
    for xi = 1:length(xdat)-1,
        xt = xdat(xi);
        %%%%%%%%%%
        il = xi;
        bet(il) = 6*av(il)*xk(il) + 2*bv(il);
        if xi == length(xr)-1,
            bet(il+1) = 6*av(il)*xk(il+1) + 2*bv(il);
        else,
            bet(il+1) = 6*av(il+1)*xk(il+1) + 2*bv(il+1);
        end
        yt = (xk(il+1)-xk(il)) * (bet(il)^2 + bet(il)*bet(il+1) + bet(il+1)^2);
        %%%%%%%%%%
        yest(xi) = yt;
    end %xi
    %%%%%%%%%%
    f2pv = sum(yest) * 4/3;
    if f2pv == 0 | f2pv < 0,
        yest
        f2pv
        pause;

```

```

end %f2pv
f2pv = log(f2pv);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
return

function [Pv,Pd] = smospl3s(xk,yk,Sp)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% returns the parameters values
% of the cubic smoothing spline
% for the sample data (xk,yk)
% the 'natural' cubic
% smoothing spline
% from csaps.m
% xk, yk: samples (knots)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n = length(xk);[xind,ind] = sort(xk);xind = xind(:);
yind = yk(ind);yind = yind(:);yk = yind;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
output = csaps(xk,yk,Sp);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
a = output(5 + n:2*n + 3);
b = output(2*n + 4:3*n + 2);
c = output(3*n + 3:4*n + 1);
d = output(4*n + 2:5*n);
Pd = [a
      b
      c
      d];
Pv = [a
      b-3*a.*xk(1:n-1)
      3*a.*(xk(1:n-1)).^2-2*b.*xk(1:n-1)+c
      -a.*(xk(1:n-1)).^3+b.*(xk(1:n-1)).^2-c.*xk(1:n-1)+d];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
return

```

### A.2.8. ValSplC1

```

function [yest] = ParValSplC1(xk,xdat,Par,choice)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% returns the value yest at xdat
% according to Par and (xk,yk)
% yest: results
% xk, yk: samples (knots)
% choice: option of the format of parameter Par
% choice = 1: yest = pv1 * xdat^3 + pv2 * xdat^2 + pv3 * xdat + pv4
% choice = 2: yest = ai * dx^3 + bi * dx^2 + ci * dx + di (dx = xdat-xk)

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
n=length(xk);
[xi,ind]=sort(xk);xi=xi(:);xk=xi';
yi=yk(ind);yi=yi(:);yk=yi';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if choice == 1,
    av=Par(1,1:n-1);
    bv=Par(2,1:n-1);
    cv=Par(3,1:n-1);
    dv=Par(4,1:n-1);
elseif choice == 2,
    a=Par(1,1:n-1);
    b=Par(2,1:n-1);
    c=Par(3,1:n-1);
    d=Par(4,1:n-1);
end %if
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for xi=1:length(xdat),
    xt=xdat(xi);
    il=1;
    ih=n;
    while ih>il+1,
        it=fix((il+ih)/2);
        if xk(it)>xt,
            ih=it;
        else
            il=it;
        end %if
        if il==ih,
            if il==n,
                il=il-1;
            else
                ih=ih+1;
            end %if
        end %if
    %%%%%%%%%%%%%%%
    if choice == 2,
        dx=xt-xk(il);
        yt=((a(il)*dx+b(il))*dx+c(il))*dx+d(il);
        ytd=(3*a(il)*dx+2*b(il))*dx+c(il);
    %%%%%%%%%%%%%%%
    elseif choice == 1,
        yt=av(il)*xt^3+bv(il)*xt^2+cv(il)*xt+dv(il);
        ytd=3*av(il)*xt^2+2*bv(il)*xt+cv(il);
    end %if
    %%%%%%%%%%%%%%%
end %while
yest(xi)=yt;
yestd(xi)=ytd;
end %xi

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
return

```

### A.2.9. *ParSmoSplb1*

```

function [yest] = ParSmoSplb1(xk,yk,xdat)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% beta-spline algorithm
% xk, yk: knots (samples)
% B1, B2: shape parameters
% yest: results at xdat
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
B1 = 1;
B2 = 0;
knum = length(xk);
n = knum;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
yest = [];
[xind,ind] = sort(xk);xind = xind(:);xk = xind;
yind = yk(ind);yind = yind(:);yk = yind;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% calculate parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Fm = 2*B1^3 + 4*B1^2 + 4*B1 + 2 + B2;
Cb(1,1) = 2*B1^3/Fm;
Cb(2,1) = -6*B1^3/Fm;
Cb(3,1) = 6*B1^3/Fm;
Cb(4,1) = -2*B1^3/Fm;
Cb(1,2) = (4*B1^2 + 4*B1 + B2)/Fm;
Cb(2,2) = (6*B1^3 - 6*B1)/Fm;
Cb(3,2) = 3*(-2*B1^3 - 2*B1^2 - B2)/Fm;
Cb(4,2) = 2*(B1^3 + B1^2 + B1 + B2)/Fm;
Cb(1,3) = 2/Fm;
Cb(2,3) = 6*B1/Fm;
Cb(3,3) = 3*(2*B1^2 + B2)/Fm;
Cb(4,3) = -2*(B1^2 + B1 + B2 + 1)/Fm;
Cb(1,4) = 0;
Cb(2,4) = 0;
Cb(3,4) = 0;
Cb(4,4) = 2/Fm;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
xk = xk';
yk = yk';
xd = zeros(1,knum + 2);
yd = zeros(1,knum + 2);
xd(1,2:knum + 1) = xk(1,1:knum);
yd(1,2:knum + 1) = yk(1,1:knum);
xd(1,1) = xk(1,1);

```

```

yd(1,1) = yk(1,1);
xd(1,knum + 2) = xk(1,knum);
yd(1,knum + 2) = yk(1,knum);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for kn = 1:length(xdat),
    xt = xdat(kn);
    for kj = 1:n-1,
        if xt >= xk(kj),
            it = kj;
        else
            break;
        end %if
    end %kj
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if xt == xk(1),
        yt = yk(1);
    elseif xt == xk(knum),
        yt = yk(knum);
    else
        for ki = 2:knum,
            if xt >= xd(ki),
                it = ki;
            else,
                break;
            end %if
        end %ki
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for ki = 1:4,
        Bb(ki) = 0;
        for kj = 1:4,
            Bb(ki) = Bb(ki) + Cb(ki,kj) * xd(it + kj-2);
        end %kj
    end %ki
    for ki = 1:4,
        ppx(ki) = Bb(5-ki);
    end %ki
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    ppx(4) = ppx(4) - xt;
    rt = roots(ppx);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    rtnum = length(rt);
    Tp = rt(1);
    for ki = 2:rtnum,
        if abs(Tp) > abs(rt(ki)),
            Tp = rt(ki);
        end %if
    end %ki
    Tp = real(Tp);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for kj = 1:4,

```

```
Bb(kj) = 0;
for kk = 1:4,
    Bb(kj) = Bb(kj) + Cb(kk,kj)*Tp^(kk-1);
end %kk
end %kj
yt = 0;
for kj = 1:4,
    yt = yt + Bb(kj)*yd(it + kj-2);
end %kj
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for ki = 1:4,
    Pv(ki) = 0;
    for kj = 1:4,
        Pv(ki) = Pv(ki) + Cb(5-ki,kj);
    end %kj
end %ki
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end %if
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
yest(kn) = yt;
end %kn
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
return
```

## BIBLIOGRAPHIE

- [1] A.Barwicz, J.-L.Dion et R.Z.Morawski, "Calibration of an Electronic Measuring System for Ultrasonic Analysis of Solutions," *IEEE Transaction Instrumentation & Measurement*, Vol.39, No.6, pp.1030-1033, Dec. 1990.
- [2] R.Z.Morawski, L.Lemire, A.Barwicz et W.J.Bock, "Calibration of an Electronic Multisensor System for Measuring High Pressures," *IEEE Transaction on Instrumentation & Measurement*, Vol.41, No.2, pp.269-273, Avril 1992.
- [3] B.A.Barsky, *Computer Graphics and Geometric Modelling Using Beta-splines*, SPRINGER Verlag, Berlin-Heidelberg-New-York-London-Paris-Tokyo, 1988.
- [4] R.Z.Morawski, "Application of Splines for Static Calibration of Measurement Channels," in: *Research Rep. CPBP 02.20/I.21/V*, Warsaw University of Technology, Institute of Radioelectronics, Warsaw 1990, Chapitre II (en Polonais).
- [5] C.de Boor, "Spline Toolbox for use with MATLAB™," *User's Guide*, pp.2-12, The MathWorks, Inc., 1990.
- [6] A.Barwicz et J.-L.Dion, "Electronic Measuring System for Ultrasonic Analysis of Solutions," *IEEE Transaction on Instrumentation & Measurement*, Vol.39, No.1, pp.269-273, Février. 1990.
- [7] A.Barwicz et W.J.Bock, "An Electronic High-Pressure Measuring System Using a Polarimetric Fibre-Optic Sensor" *IEEE Transaction on Instrumentation & Measurement*, Vol.39, No.6, pp.976-981, Dec. 1990.
- [8] T.M Canh, *Les Biocapteurs - Principes, Construction et Applications*, MASSON, Paris-Milan-Barcelone-Bonn, 1991, pp.103-107.
- [9] R.H.Bartels, J.C.Beatty et B.A.Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*, Morgan Kaufmann Publishers, Inc. 1987.
- [10] C.de Boor, *Applied Mathematical Science 27: A Practical Guide to Splines*, 1978, Springer-Verlag, 392p.
- [11] L.Lemire, "L'étalonnage statique d'un système de mesure avec les capteurs en fibres optiques," *Mémoire de maîtrise*, Dépt. d'ingénierie, UQTR, 1992.

- [12] R.G.Keys, "Cubic Convolution Interpolation for Digital Image Processing," *IEEE Transactions on Acoustics, Speech, And Signal Processing*, Vol. ASSP-29, No. 6, Decembre 1981, pp. 1153-1160.
- [13] J.H.Ahlberg, E.N.Nilson et J.L.Walsh, *The Theory of Splines and Their Applications*, Academic Press, 1967.
- [14] P.J.Davis, *Interpolation and Approximation*, Blaisdell Publishing Company, 1963.
- [15] T.J.Rivlin, *An Introduction to the Approximation of Functions*, Blaisdell Publishing Company, 1969.
- [16] D.Hearn et M.P.Baker, *Computer Graphics*, 1986, Prentice-Hall, 352p.
- [17] S.Harrington, *Computer Graphics--A Programming Approach*, 1983, Mcgraw-Hill, 448p.
- [18] G.Farin, *Computer Science And Scientific Computing--Curves And Surfaces For Computer Aided Geometric Design--A Practical Guide*, 1988, Academic Press, 334p.
- [19] P.Burger et D. Gillies, *Interactive Computer Graphics--Functional, Procedural And Device-Level Methods*, 1989, Addison-Wesley, 504p.
- [20] D.F.Rogers et J.A.Adams, *Mathematical Elements For Computer Graphics*, 1976, Mcgraw-Hill, 239p.
- [21] R.Sedgewick, *Algorithms*, 1988, Addison-Wesley, 657p.
- [22] R.Z.Morawski, L.Lemire, M.Ben Slima et A.Barwicz, "The Use Of Splines for Static Calibration of Electronic Measuring Systems - A Comparative Study," *Proc. IEEE Instrum. & Meas. Technology Conf.* (Atlanta, 14-16 Mai 1991).
- [23] "International Vocabulary of Basic and General Terms in Metrology", *National Standard of Canada*, 1987.
- [24] G.Farin, *Courbes et surfaces pour la CGAO--conception géométrique assistée par ordinateur*, 1992, MASSON, 399p.
- [25] *The Random House Dictionary of the English Language*, Second Edition, 1987, Unabridged, 2478p.
- [26] J.A.Simpson et E.S.C.Weiner, *The Oxford English Dictionary*, Second Edition, Vol.XVI, 1989, Clarendon Press.