

## Synthèse d'images scanners à partir de résultats de simulations numériques

**M**AINTEANT que nous avons un modèle EDP qui reproduit bien les aspects constatés en clinique, interrogeons-nous sur la manière de présenter ses résultats. La simulation numérique fournit l'évolution des densités  $N(t, x)$ ,  $P(t, x)$  et  $S(t, x)$  (toutes comprises entre 0 et 1). Les données cliniques quant à elles fournissent des images en niveaux de gris. L'enjeu ici, consiste à traiter les résultats numériques de sorte à produire des images en niveaux de gris comparable à un scanner. Rappelons que le niveau de gris d'un scanner traduit l'absorption des tissus aux rayons X. D'un point de vue biologique, on ne peut pas relier à une population un niveau de gris car les populations ne sont pas isolées, et aucun moyen ne permet d'en déceler leur proportions dans chaque voxel du scanner. Le lien entre densité de populations et absorption aux rayons X n'est donc pas connu. Ainsi le choix a été fait de créer une image scanner de synthèse, à partir de l'évolution des populations données par la simulation numérique, en considérant l'interpolation

$$\tau_N N + \tau_P P + \tau_S S, \quad (1.1)$$

où les paramètres  $\tau_N, \tau_P$  et  $\tau_S$  sont les niveaux de gris que l'on associe aux populations  $N, P$  et  $S$ . N'ayant que peu d'indication sur la nature de ces paramètres, nous proposons ici de les optimiser de sorte à ce que les images de synthèse se rapprochent le plus possible des images scanners.

### 1.1 Présentation de l'approche

Pour un patient donné, on considère  $n$  instants auxquels on possède des images scanners (aux temps  $t_i, i \in \{1, \dots, n\}$ ). Sur ces  $n$  images, on propose d'optimiser les coefficients (niveaux de gris) de l'interpolation  $\tau_N N + \tau_P P + \tau_S S$  où  $N, P$  et  $S$  sont les populations définies dans le modèle présenté précédemment. Sur l'ensemble de ces images, on fait correspondre le niveau de gris

moyen des images numériques à celui des scanners, ce qui s'écrit :

$$\begin{aligned} \frac{1}{\mathcal{A}(Z_1(t_i))} \left( \tau_N \int_{Z_1(t_i)} N(t_i, x) \, dx + \tau_P \int_{Z_1(t_i)} P(t_i, x) \, dx + \tau_S \int_{Z_1(t_i)} S(t_i, x) \, dx \right) \\ = \frac{1}{\mathcal{A}(Z_2(t_i))} \int_{Z_2(t_i)} s(t_i, x, z_0) \, dx \quad i \in \{1, \dots, n\} \end{aligned} \quad (1.2)$$

où :

- $\mathcal{A}(Z)$  est l'aire de la zone  $Z$ .
- $Z_1(t_i)$  est la zone correspondant à la tumeur dans les simulations numériques au temps  $t_i$ . Elle est définie par un seuillage sur  $S$ .
- $Z_2(t_i)$  est la zone tumorale sur le scanner réalisé au temps  $t_i$ . Cette zone a été définie par contourage manuel à l'aide du logiciel OsiriX (*cf.* contourage des tumeurs en Annexe ?? pages ?? et ??).
- $z_0$  est la coupe que l'on choisit d'étudier dans les scanners. Cette coupe est approximativement la même au cours du temps de sorte à suivre l'évolution d'une même section du foie.
- $s(t_i, x, z_0)$  est la valeur du niveau de gris du pixel en position  $x$  sur la coupe  $z_0$  du scanner effectué au temps  $t_i$ .

En utilisant la discrétisation, aussi bien sur les simulations numériques que sur les scanners, on obtient :

$$\begin{aligned} \frac{1}{\mathcal{N}(Z_1(t_i))} \left( \tau_N \sum_{x \in Z_1(t_i)} N(t_i, x) + \tau_P \sum_{x \in Z_1(t_i)} P(t_i, x) + \tau_S \sum_{x \in Z_1(t_i)} S(t_i, x) \right) \\ = \frac{1}{\mathcal{N}(Z_2(t_i))} \sum_{x \in Z_2(t_i)} s(t_i, x, z_0) \quad i \in \{1, \dots, n\} \end{aligned} \quad (1.3)$$

où  $\mathcal{N}(Z)$  désigne le nombre de pixels contenus dans la zone  $Z$ . On a donc un système linéaire de 3 inconnues à  $n$  équations que l'on peut réécrire :

$$A\tau = B, \quad (1.4)$$

avec  $\tau = {}^t(\tau_N, \tau_P, \tau_S)$ ,  $A$  matrice de taille  $n \times 3$  et  $B$  vecteur colonne de taille  $n$ .

Pour ne pas se limiter au cas  $n = 3$  qui clôt le système, on le résout par la minimisation suivante :

$$\min_{\tau} J(\tau) \quad \text{avec} \quad J(\tau) = \frac{\|A\tau - B\|_{\ell^2}^2}{\|B\|_{\ell^2}^2} + \mathcal{P}(\tau), \quad (1.5)$$

où  $\mathcal{P}$  pénalise la fonction coût  $J$  lorsque l'une des composantes de  $\tau$  est en dehors de l'intervalle  $[0; 255]$ . Une pénalisation en créneau sera considérée ici

$$\mathcal{P}(\tau) = 1e7 \times \mathbf{1}(\tau \notin [0; 255]^3). \quad (1.6)$$



## 1.2 Optimisation sur 3 paramètres

La résolution de l'équation (1.5) fournit le  $\tau$  optimal. Il y a cependant plusieurs manières de calculer cet optimum. On peut faire varier :

- le nombre d'images considérées
- les moments considérés
- l'algorithme d'optimisation lui-même

Dans tous les cas, on ne considérera pas le premier scanner (numéro 0) car la condition initiale du modèle EDP (donnée par les équations (??) et (??) page ??) n'est pas prise de sorte à respecter la répartition des niveaux de gris du scanner. Evitons donc d'inclure dans l'optimisation une erreur de base qui serait incompressible. On regardera des situations avec seulement 2 images (problème sous-déterminé) ou 3 images (problème fermé) ou plus (problème sur-déterminé).

En ce qui concerne les algorithmes d'optimisations utilisés, nous en choisissons quatre :

- SLSQP (Sequential Least Squares Programming) [?] : Méthodes des moindres carrés
- GC [?] : Gradient Conjugué
- Nelder-Mead [?, ?, ?] : Algorithme basé sur une méthode du Simplex.
- BFGS (Broyden, Fletcher, Goldfarb, and Shanno) [?] : Méthode quasi newtonnienne basée également sur une approximation de la dérivée.

Ces algorithmes seront initialisés avec les paramètres  $\tau_N, \tau_P$  et  $\tau_S$  considérés dans la section ?? page ?. Lorsqu'ils sont nécessaires, les gradients (voire les hessiennes) sont approchées par les algorithmes eux-mêmes. Outre un vecteur de paramètre initial et la fonction coût à optimiser (1.5), aucune information supplémentaire n'est fournie à ces différents algorithmes. Le code de calcul permettant de réaliser ces optimisations a été développé en Python et s'appuie sur la librairie SciPy [?].

La Table 1.1 synthétise une partie des résultats d'optimisation obtenus sur les différents tests qui ont été réalisés. On remarque que plus le nombre d'images considérées est grand, plus l'erreur à convergence a tendance à être grande. Ce comportement est attendu et ne pose pas de problème tant que l'erreur reste acceptable (de l'ordre de quelques pourcents). Augmenter le nombre d'images considérées s'avère utile pour rendre les optima moins sensibles aux perturbations éventuelles qu'il y a sur les données (bruit, marge d'erreur de segmentation manuelle, etc.).

On peut de plus remarquer que selon les images et les algorithmes choisis les résultats sont assez variables. Les moyennes des optima trouvés selon l'algorithme sont présentées sur la dernière ligne de la Table 1.1. Seul l'algorithme SLSQP se démarque des autres qui ont :

- une valeur moyenne de  $\tau_P$  supérieure à  $\tau_S$  alors qu'on s'attendrait plutôt à l'inverse. Des images ont été reconstituées avec des valeurs de  $\tau_S$  aux

alentours de 140 et 150 (on pourra consulter la Figure 1.1 pour se donner une idée de l'intensité de gris que cela représente). Il apparaît que la couleur du tissu sain n'est pas bonne : il est beaucoup trop foncé.

- pour environ un tiers des combinaisons, une valeur de  $\tau_P$  proche de celle de  $\tau_S$ , ce qui ne facilite pas du tout le contraste du tissu proliférant avec le tissu sain.

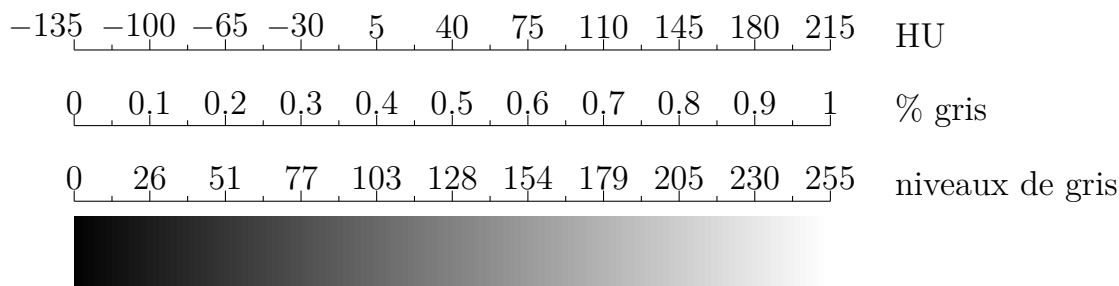


FIGURE 1.1 – Correspondance entre l'échelle de Hounsfield (tomodensitométrie en HU) et les niveaux de gris – Les bornes sont ici fixées à -135 HU et 215 HU.

L'algorithme Nelder-Mead semble le moins bon algorithme, au vu notamment des résultats fournis pour les combinaisons [1,9,11], [2,3,4,5] et [3,5,7] qui font apparaître une valeur de  $\tau_N$  de 255, ce qui correspond à du blanc. Non seulement cette valeur est grande, mais en plus elle atteint la valeur limite autorisée. Notons que si l'on retire ces 3 valeurs de la moyenne présentée sur la dernière ligne du tableau, alors celle-ci s'approche des moyennes des algorithmes GC et BFGS.

Bien que paraissant le meilleur algorithme, l'optimisation réalisée par la méthode SLSQP sur 3 paramètres n'est pas acceptable. Elle présente les mêmes défauts : le niveau de gris moyen du tissu proliférant est trop proche encore de celui du tissu sain et l'intensité de gris  $\tau_S$  du tissu sain semble sous-estimée.

Pour expliquer l'erreur commise sur  $\tau_S$ , on pourrait penser à une large variabilité permise sur ce paramètre due à la présence très minoritaire de tissu sain dans la tumeur. Ainsi  $\tau_S$  serait très peu influent dans le calcul de l'erreur (1.5). Cependant après avoir examiné de plus près les valeurs des intégrales de l'équation (1.2) (*i.e.* les coefficients de la matrice  $A$ ), il y a toujours au moins 10% de cellules saines. La plupart du temps, elles sont réparties sur le pourtour de la tumeur, dans la zone de transition sur laquelle il y a un mélange de tissu sain et de tissu tumoral. Ceci écarte donc l'hypothèse avancée. On pourrait alors avancer des variations dans les données (bruit dans les images, erreur sur le contourage, variation du temps d'acquisition

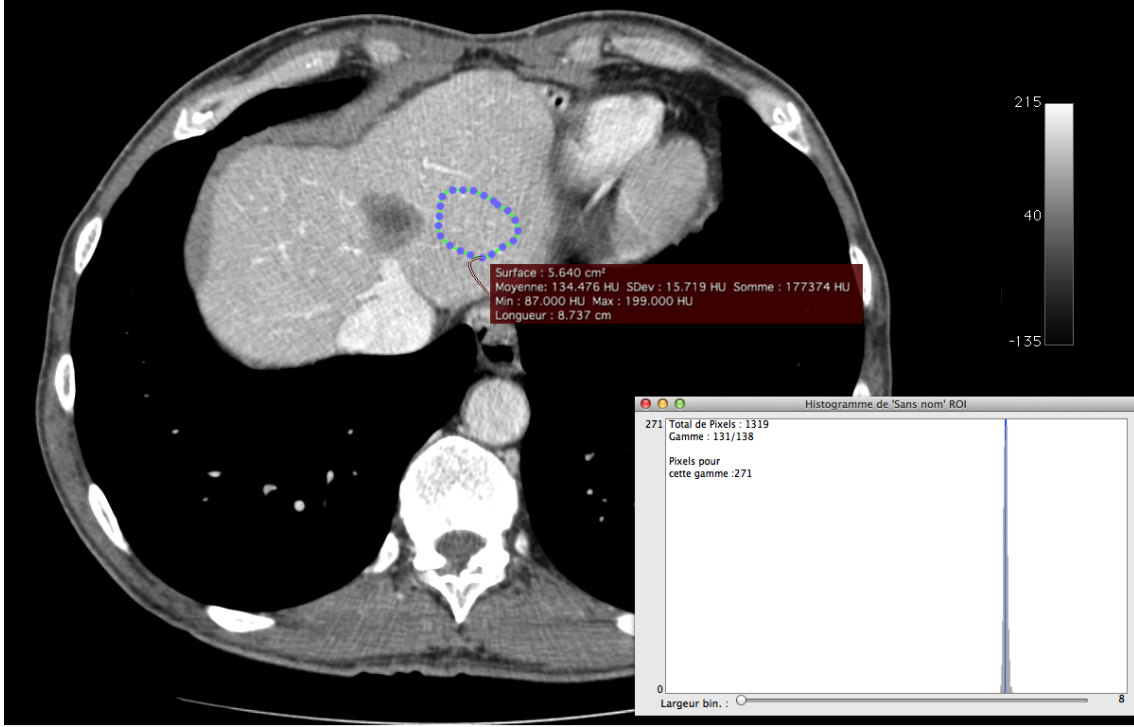


FIGURE 1.2 – Contourage d'une zone saine réalisé à l'aide du logiciel OsiriX – Moyenne de la valeur des pixels dans ce périmètre : 134,5 HU (avec une échelle HU de -135 à 215).

du scanner qui impacte les niveaux de gris, ... ) pour justifier cela, mais les impacts sont difficiles à mesurer. Il n'en reste pas moins que le niveau de gris  $\tau_S$  est mal estimé, et ce quel que soit l'algorithme d'optimisation utilisé.

### 1.3 Optimisation sur 2 paramètres, $\tau_S$ fixé

Pour essayer de palier aux problèmes rencontrés dans la section précédente, nous allons fixer  $\tau_S$  à une valeur de 197 (sur l'échelle des niveaux de gris de 0 à 255). Cette valeur a été fixée en réalisant un contourage d'une zone de tissu sain dans OsiriX (*cf.* Figure 1.2). La moyenne de ce contourage est de 134,5 HU. Le niveau de gris étant échelonné linéairement entre -135 et 215, on peut ainsi faire correspondre cette quantité en HU à un niveau de gris compris entre 0 et 255 (*cf.* Figure 1.1). Ainsi, selon l'échelle considérée ici, 134,5 HU est équivalent à un gris de 77% soit un gris de niveau 197.

Ainsi nous résolvons toujours l'équation (1.4), mais ici avec

$$\begin{aligned}
 A_{i,\dots,\tau} &= \frac{1}{\mathcal{N}(Z_1(t_i))} \left( \tau_N \sum_{x \in Z_1(t_i)} N(t_i, x) + \tau_P \sum_{x \in Z_1(t_i)} P(t_i, x) \right), \\
 B_i &= \frac{1}{\mathcal{N}(Z_2(t_i))} \sum_{x \in Z_2(t_i)} s(t_i, x, z_0) - \frac{\bar{\tau}_S}{\mathcal{N}(Z_1(t_i))} \sum_{x \in Z_1(t_i)} S(t_i, x),
 \end{aligned} \tag{1.7}$$

Scanners choisis pour l'optimisation	Algorithmes d'optimisation							
	SLSQP		GC		Nelder-Mead		BFGS	
	$\tau_N$ ,	$\tau_P$	$\tau_N$ ,	$\tau_P$	$\tau_N$ ,	$\tau_P$	$\tau_N$ ,	$\tau_P$
[1, 2]	38.64, 145.56 Err : 8.0e-07		38.64, 145.56 Err : 2.7e-11		38.64, 145.56 Err : 8.6e-08		38.64, 145.56 Err : 3.4e-11	
[1, 2, 3]	33.65, 145.78 Err : 1.3e-02		33.6, 145.8 Err : 1.3e-02		33.63, 145.79 Err : 1.3e-02		33.67, 145.78 Err : 1.3e-02	
[1, 2, 3, 4]	27.62, 145.69 Err : 3.2e-02		26.63, 146.05 Err : 3.2e-02		26.53, 146.08 Err : 3.2e-02		26.51, 146.08 Err : 3.2e-02	
[1, 2, 3, 4, 5]	25.93, 144.01 Err : 5.1e-02		19.97, 146.34 Err : 4.9e-02		19.82, 146.4 Err : 4.9e-02		19.77, 146.42 Err : 4.9e-02	
[1, 3]	27.72, 146.09 Err : 5.5e-07		27.42, 146.12 Err : 5.3e-04		27.72, 146.09 Err : 1.9e-08		27.72, 146.09 Err : 2.3e-11	
[1, 3, 5]	27.55, 142.69 Err : 5.0e-02		12.47, 146.8 Err : 4.2e-02		12.35, 146.84 Err : 4.2e-02		12.34, 146.84 Err : 4.2e-02	
[1, 2, 5]	28.23, 144.23 Err : 5.6e-02		20.87, 146.33 Err : 5.4e-02		20.71, 146.38 Err : 5.4e-02		20.68, 146.39 Err : 5.4e-02	
[1, 2, 3, 5]	27.26, 144.83 Err : 5.0e-02		23.32, 146.22 Err : 5.0e-02		23.17, 146.27 Err : 5.0e-02		23.16, 146.27 Err : 5.0e-02	
[1, 2, 3, 7]	25.65, 141.8 Err : 1.3e-01		11.26, 146.64 Err : 1.3e-01		10.86, 146.77 Err : 1.3e-01		10.78, 146.81 Err : 1.3e-01	
[1, 2, 9]	28.58, 134.69 Err : 2.9e-01		34.13, 133.49 Err : 2.9e-01		35.21, 133.25 Err : 2.9e-01		35.33, 133.23 Err : 2.9e-01	
[1, 3, 9, 11]	29.62, 133.93 Err : 2.5e-01		19.26, 135.52 Err : 2.5e-01		17.99, 135.72 Err : 2.5e-01		17.55, 135.83 Err : 2.5e-01	
Moyenne :	29.13, 142.66		24.32, 144.08		24.24, 144.1		24.19, 144.12	
Ecart-type :	3.63, 4.14		8.37, 4.55		8.66, 4.58		8.73, 4.57	

TABLE 1.2 – Tableau récapitulatif des optimisations réalisées sur 2 niveaux de gris,  $\tau_S$  fixé à 197, avec un créneau comme pénalisation de l'intervalle – cas favorables

avec  $i \in \{1, \dots, n\}$ , où  $A_{i,\dots}$  désigne la  $i$ -ième ligne de la matrice  $A$  et où  $\bar{\tau}_S$  est fixé à 197.

Les premiers essais ont été réalisés avec la même fonction coût (1.5) et avec la même pénalisation (1.6) que pour l'optimisation sur 3 paramètres. L'ensemble des résultats d'optimisation de  $\tau_N$  et de  $\tau_P$  avec  $\tau_S$  fixé à 197 est fourni dans les Tables 1.2 et 1.3.

La Table 1.2 présente les cas correctement convergés. Ici les niveaux de gris moyens fournis sont conformes aux attentes dans le sens où l'on a  $\tau_N < \tau_P < \tau_S$ . Une nouvelle fois, l'algorithme basé sur la méthode des moindres carrés se démarque des 3 autres qui fournissent des optima très similaires. Contrairement aux résultats fournis par l'optimisation sur 3 paramètres, les variations sur les optima sont beaucoup moins importantes en fonction des images choisies, comme en atteste le faible écart-type, inférieur à 10 niveaux de gris (*cf.* dernière ligne de la Table 1.2), voire inférieur à 5 niveaux de gris pour l'algorithme SLSQP sur les 256 qu'offre l'échelle. Les résultats obtenus ici sont donc tout à fait satisfaisant. Les niveaux de gris que l'on pourra considérer comme optimaux seront ainsi donné par la moyenne de tous ces cas de figures,

CHAPITRE 1. Synthèse d’images scanners à partir de résultats de simulations numériques

Scanners choisis pour l’optimisation	Algorithme d’optimisation			
	SLSQP	GC	Nelder-Mead	BFGS
	$\tau_N, \tau_P$	$\tau_N, \tau_P$	$\tau_N, \tau_P$	$\tau_N, \tau_P$
[2, 3]	25.49, 150.89 Err : 1.9e-02	31.85, 146.9 Err : 1.9e-02	255.0, 6.31 Err : 1.4e-02	226.63, 22.8 Err : 1.6e-02
[2, 3, 4]	23.64, 147.84 Err : 4.4e-02	32.94, 142.01 Err : 4.3e-02	255.0, 2.76 Err : 3.6e-02	220.28, 27.1 Err : 4.0e-02
[2, 3, 5]	22.75, 146.5 Err : 6.9e-02	27.06, 143.79 Err : 6.9e-02	255.0, 0.39 Err : 6.6e-02	223.15, 18.18 Err : 6.7e-02
[2, 3, 4, 5]	21.9, 145.05 Err : 6.5e-02	27.6, 141.48 Err : 6.5e-02	253.36, 0.0 Err : 6.0e-02	222.72, 21.59 Err : 6.3e-02
[2, 5, 7]	15.96, 135.58 Err : 2.0e-01	24.3, 130.33 Err : 2.0e-01	231.68, 0.0 Err : 2.0e-01	219.66, 12.28 Err : 2.0e-01
[1, 7, 11]	0.0, 139.22 Err : 1.1e-01	0.3, 139.4 Err : 1.1e-01	0.0, 139.65 Err : 1.0e-01	0.11, 131.97 Err : 1.2e-01
[1, 9, 11]	31.17, 131.88 Err : 2.7e-01	-0.01, 134.01 Err : 3.3e+02	0.0, 134.11 Err : 2.6e-01	2.14, 140.56 Err : 2.7e-01
[1, 2, 7, 11]	28.41, 138.98 Err : 1.3e-01	6.75, 143.27 Err : 1.2e-01	6.3, 143.36 Err : 1.2e-01	6.13, 143.4 Err : 1.2e-01
[1, 3, 7, 11]	27.94, 137.75 Err : 1.2e-01	0.03, 142.99 Err : 1.1e-01	0.0, 143.06 Err : 1.1e-01	0.09, 145.57 Err : 1.1e-01
[3, 5, 7]	13.92, 132.24 Err : 1.9e-01	14.03, 132.18 Err : 1.9e-01	225.52, 0.0 Err : 1.9e-01	13.91, 132.23 Err : 1.9e-01
[1, 3, 5, 7]	23.41, 137.16 Err : 1.4e-01	-0.04, 144.54 Err : 1.4e+04	0.0, 144.78 Err : 1.2e-01	1.2, 148.93 Err : 1.3e-01
[2, 3, 5, 7]	18.39, 139.46 Err : 1.7e-01	23.89, 136.02 Err : 1.7e-01	240.58, 0.0 Err : 1.7e-01	216.95, 18.63 Err : 1.7e-01
[1, 2, 5, 7]	24.17, 138.77 Err : 1.4e-01	-0.03, 146.6 Err : 9.2e+03	0.0, 146.65 Err : 1.3e-01	0.01, 144.07 Err : 1.3e-01

TABLE 1.3 – Tableau récapitulatif des optimisations réalisées sur 2 niveaux de gris,  $\tau_S$  fixé à 197, avec un créneau comme pénalisation de l’intervalle – cas défavorables.

à savoir aux environs de :

$$\tau_N = 25, \quad \tau_P = 143, \quad \text{avec } \tau_S = 197. \quad (1.8)$$

Notons tout de même que dans certaines configurations, présentées dans la Table 1.3, les algorithmes tendent vers un jeu de paramètres optimal qui s’approche du bord 0 ou du bord 255 (*i.e.* non convergence de l’algorithme d’optimisation) ou bien tel que  $\tau_N \gg \tau_S$  ce qui est aberrant. Ces erreurs pourraient être dues notamment au fait que la pénalisation choisie (1.6) présente une discontinuité. Les algorithmes de descente fonctionnant sur une approximation du gradient peuvent ainsi être perturbés par cette discontinuité aux abords des bornes autorisées. Dans l’annexe ?? page ??, l’enquête menée sur les fonctions de pénalisation est présentée. Des pénalisations plus régulières ont été testées (parabole tronquée et parabole tronquée régularisée) mais n’améliorent que très peu le résultat final. Ceci nous amène à penser que ce n’est donc pas la régularité de la pénalisation qui est à mettre en cause mais les données elles-mêmes. Certaines combinaisons d’images fourniraient donc



de mauvais résultats, d'autres de bons.

**Pourquoi y-a-t-il des cas convergents et d'autres non ?**

Scanners choisis pour l'optimisation	Conditionnement matrice 3x3	Scanners choisis pour l'optimisation	Conditionnement matrice 2x2
[2, 3, 4]	1.5e+07	[2, 3]	1.2e+05
[2, 3, 5]	2.8e+05	[1, 2]	1.6e+01
[2, 5, 7]	3.1e+06	[1, 3]	1.8e+01
[1, 7, 11]	1.8e+03		
[1, 9, 11]	1.6e+07		
[3, 5, 7]	4.7e+06		
[1, 2, 3]	6.7e+02		
[1, 3, 5]	1.5e+02		
[1, 2, 5]	6.9e+01		
[1, 2, 9]	2.2e+01		

TABLE 1.4 – Conditionnement (rapport de la plus grande valeur propre sur la plus petite) des systèmes considérés lors de la synthèse d'images scanners. – Partie supérieure des tableaux : mauvais conditionnement / Partie inférieure : conditionnement acceptable.

Pour une combinaison de 2 ou 3 images, on s'attend à ce que l'optimum soit la solution exacte du système linéaire, et ce indépendamment de l'algorithme choisi. Dans la mesure où ce n'est pas le cas, nous devons nous interroger sur le conditionnement des systèmes que l'on résout. S'ils ne sont pas bien conditionnés, alors une petite perturbation des données entraîne un très grand écart sur la solution du système. Bien que nous ne résolvions pas directement le système, mais que nous effectuions une minimisation, ces problèmes de sensibilité aux données n'ont aucune raison de ne pas se reporter. La Table 1.4 présente les conditionnements (rapport de la plus grande valeur propre sur la plus petite) associés aux différentes combinaisons d'images que nous avons examinées. Pour les cas ayant bien convergé (partie inférieure des tableaux), le conditionnement reste raisonnable (excepté peut-être pour le cas [1,2,3]). Maintenant, si on examine les cas ayant mal convergé, on constate des conditionnements très élevés (dans la partie supérieure des tableaux, les conditionnements dépassent  $10^3$ ). Le conditionnement semble donc expliquer pourquoi certaines configurations avec 2 ou 3 images convergent bien, et d'autres non. Pour les configurations avec plus d'images, un examen des conditionnements des sous-systèmes carrés permet d'arriver à la même conclusion.

## 1.4 Conclusion

Des cas mal conditionnés ont été identifiés. Après élimination de ceux-ci, on considère les optima fournis par les cas biens conditionnés restants. Les optima fournis sont relativement proches les uns des autres même en faisant varier la méthode d'optimisation ou bien les images considérées comme en atteste la Table 1.2. Dans la suite de ce manuscrit, on fixera alors les niveaux de gris  $\tau_N = 25$ ,  $\tau_P = 143$  et  $\tau_S = 197$  qui sont les niveaux de gris moyens identifiés comme optimaux.