

Revue des Sciences et Sciences de l'Ingénieur

<u>_</u>

ISSN 2170-0737

Journal homepage : <u>http://www.RSSI.lagh-univ.dz</u>

Simulation de la diffusion anisotrope en traitement d'images par la méthode des éléments finis

S. Y. IDRISSI, S. BELFKIH, S. NAJAH, P. MONTESINOS

Laboratoire GRETIC – Faculté des sciences et techniques Fès Saïs B.P. 2202 – Route d'Imouzzer – FES – MAROC

**Corresponding author: sidiyassine12@yahoo.fr

Résumé - Dans cet article, nous proposons une nouvelle mise en œuvre numérique de la méthode de Malik et Perona pour la restauration des images en niveau de gris à base d'équations aux dérivées partielles (EDP) par la méthode des éléments finis(MEF). Tout d'abord, nous présentons le formalisme de base de la diffusion anisotrope. Ensuite, après avoir rappelé les grandes lignes de la MEF que nous appliquerons par la suite sur le modèle de Malik et Perona, nous comparons qualitativement et quantitativement les résultats obtenus par notre méthode de résolution numérique avec d'autres résultats issus de résolutions explicites et semi-implicite. Plusieurs tests appliqués sur des images synthétiques et réelles sont présentés afin de bien illustrer l'efficacité de notre approche.

Mots Clés - Diffusion Anisotrope, Equations aux Dérivées Partielles, Eléments Finis, Restaurations d'Images

I. Introduction

L'application des équations aux dérivées partielles en traitement d'images a attiré depuis quelques années l'attention de plusieurs chercheurs en vision par ordinateur [1] [2]. Ceci est dû surtout au formalisme mathématique qui encadre toute approche à base d'EDP et qui permet de donner une bonne interprétation et justification des résultats obtenus par ces méthodes. contrairement aux méthodes classiques et heuristiques en traitement d'images [3]. Ces équations ont été notamment étudiées dans le cadre de l'analyse multi-échelle [4] et dans le cadre des problèmes de restauration d'images basés sur le principe de la diffusion anisotrope [5].

La diffusion intervient en prétraitement, afin de supprimer les perturbations locales du signal. Il devient alors possible dans un second temps d'effectuer par exemple, une recherche des contours sans être gêné par le bruit. La diffusion isotrope rend les contours flous et homogénéise l'image. Malik et Perona [6] ont proposé un modèle de diffusion anisotrope. La motivation essentielle de l'utilisation de ce genre de modèle est la construction d'un opérateur de diffusion dépendant des propriétés locales de l'image. Ils ont proposé un filtre permettant d'atténuer la diffusion dans les régions à fort gradient (Diffusion minimale au niveau des contours) et de la maintenir dans les zones à faible gradient.

La performance de ces méthodes, à base d'EDP, en traitement d'images dépend en grande partie de la résolution numérique et de la mise en œuvre de ces EDP. Le principe des schémas numérique les plus représentatifs se repose sur la discrétisation des dérivées du premiers et second ordre $(D_x, D_y, D_{xx}, D_{yy}, D_{xy})$. Pour assurer une meilleure stabilité, plusieurs méthodes proposent des schémas explicites qui consistent à déterminer la solution approchée à l'instant n+1 à partir de celle à l'instant n. Bien que la programmation est simple et peu coûteuse en utilisant un schéma numérique explicite, elle devient prohibitive au cours du temps.

Dans ce papier, nous proposons une nouvelle mise en œuvre numérique de la méthode de Malik et Perona pour la restauration des images en niveau de gris à base d'équations aux dérivées partielles (EDP) par la méthode des éléments finis(MEF).

La suite de la présente étude est organisée de la façon suivante: dans la section 2 nous

rappelons brièvement la diffusion anisotrope et le modèle de Malik et Perona; dans la section 3 nous décrivons et mettons en œuvre la méthode des éléments finis pour ce modèle; Dans la section 4 nous présentons la résolution explicite et semi implicite de l'équation de Malik et Perona. Les résultats et la comparaison des différentes méthodes sont présentés dans la section 5. La section 6 conclut le papier.

II. Equation Aux Dérivées Partielles En Traitement D'images

1.Diffusion anisotrope par EDP

L'idée de base de la diffusion anisotrope en restauration d'images est de filtrer (clarifier) l'image en privilégiant une direction de diffusion (étalage du bruit) par rapport à une autre, selon la géométrie locale de chaque pixel. Nous présenterons brièvement une approche variationnelle afin d'aboutir à une formulation unificatrice de la diffusion anisotrope et à des méthodes de restauration d'images à base d'EDP.

L'objectif de cette approche est de retrouver une image originale I à partir de l'image dégradée observée I_0 . Le modèle de dégradation reliant I à I_0 le plus couramment utilisé est un modèle linéaire de la forme:

$$I_0 = PI + \upsilon \tag{1}$$

Où:

- *P* est un operateur de dégradation linéaire, en général identité ou un operateur de convolution.

- v Un bruit blanc gaussien.

On cherche *I* parmi les minima de la fonctionnelle énergie définie par:

$$E(I) = \frac{1}{2} \|I - PI_0\| + \lambda \int_{\Omega} \phi(\|\nabla I\|) d\Omega$$
 (2)

Le premier terme de l'énergie E(I) est un terme d'attache aux données et le second est un terme de régularisation dépendant de la fonction ϕ . Le paramètre λ permet de régler l'influence de chaque terme de l'énergie.

L'équation d'Euler-Lagrange associée à la minimisation de *E* s'écrit sous la forme:

$$P^{*}(I - PI_{0}) + \lambda Div(\phi'(\|\nabla I\|) \frac{\nabla I}{\|\nabla I\|})$$
(3)

Qu'on peut aussi écrire sous la forme:

$$P^{*}(I - PI_{0}) + \lambda(\phi(\|\nabla I\|)I_{\xi\xi} + \phi(\|\nabla I\|)\frac{\nabla I}{\|\nabla I\|}I_{\eta\eta} = 0$$
(4)

Avec:

- P^* est l'opérateur adjoint de P.
- Div est l'opérateur divergence.
- Le paramètre λ est supérieur à 0.

Dans le cas où $\lambda > 0$, l'équation peut s'écrire sous la forme:

$$\frac{\partial I}{\partial t} = C_{\xi} I_{\xi\xi} + C_{\eta\eta} I_{\eta\eta} \tag{5}$$

Avec:

$$C_{\xi} = \phi''(\|\nabla I\|) C_{\eta} = \phi'(\|\nabla I\|) \frac{\nabla I}{\|\nabla I\|}$$
$$\eta = \frac{\nabla I}{\|\nabla I\|}, \ \xi \perp \eta . \tag{6}$$

On remarque que le choix de la direction de diffusion dépend de la fonction de régularisation ϕ . Par exemple si on choisit $\phi(x) = x^2$, alors l'équation précédente correspond à l'équation classique de la chaleur et traduit une diffusion isotrope, qui tout en diminuant le bruit rend cependant l'image plus flou.

Parmi les modèles de restauration les plus représentatifs basés sur le principe de la diffusion anisotrope, on trouve le Scale Space de Malik et Perona [6], le modèle limite proposé par L. Alvarez [5] et le filtre de choc pour les images flous [7].

2. Diffusion anisotrope de Malik et Perona

Comme nous l'avons dit en introduction, Perona et Malik [6] ont proposé un modèle de diffusion anisotrope pour résoudre les problèmes issus de la diffusion isotrope. Ils ont proposé un filtre permettant d'atténuer la diffusion dans les régions à fort gradient et de la maintenir dans les zones à faible gradient. Le scale space qui formalise cette idée est donné par:

$$\begin{cases} \frac{\partial I}{\partial t}(t, x, y) = div(c(\|\nabla I\|) \nabla I) \\ I(0, x, y) = I_0(x, y) \\ \frac{\partial I}{\partial n}(t, x, y) = 0 \end{cases}$$
(7)

Quand c = 1, on retrouve l'équation de la chaleur. On impose souvent $\lim_{t\to+\infty} c(t) = 0$ et c(0) = 1. Ainsi, dans les régions de faible gradient, l'équation agit essentiellement comme l'EDP de la chaleur, et dans les régions de fort gradient, la régularisation est stoppée ce qui permet de préserver les bords. Dans la suite nous

prenons
$$c(t) = \exp\left(-\frac{t^2}{k}\right)$$
où $k \ge 0$ est un

paramètre qui contrôle la diffusion.

III. Mise En Œuvre de l'EDP de Malik et Perona par la Méthode des Eléments Finis

1. Rappel sur la méthode des éléments finis

La méthode des éléments finis consiste à résoudre des équations elliptiques ou paraboliques. Elle a l'avantage, par rapport à la méthode des différences finies, de pouvoir traiter sans difficultés supplémentaires toute géométrie ainsi que d'augmenter la précision des résultats au prix d'efforts de programmation raisonnables. C'est la méthode des éléments finis qui est systématiquement utilisée pour le calcul des structures et de très nombreux codes d'éléments finis ont été développés.

Il s'agit de mettre en place, à l'aide de la formulation variationnelle, un algorithme discret mathématique permettant de rechercher une solution approchée.

En effet, en présence d'une fonction définie sur un ouvert Ω de IR^p p = (1, 2, 3)à valeurs dans IR d'une part et un maillage de l'ouvert Ω d'autre part, la méthode des éléments finis permet la construction d'une fonction approchée, qui ne dépend plus que d'un nombre fini de degrés de libertés. Ainsi au lieu de considérer la solution de l'équation aux dérivées partielles dans un espace de dimension infinie, on cherche la solution de ce même problème dans un espace de dimension finie. Toutefois pour que le problème initial soit bien posé dans ce nouvel espace, on doit en transformer la formulation et on fait alors recours à la formulation variationnelle (ou faible). L'avantage de la méthode des éléments finis vient du fait que toute fonction dans l'espace d'approximation peut se décomposer de façon très simple sur une base canonique [8] [9] [10].

2. Résolution numérique du modèle de Malik et Perona par la MEF:

Contrairement à la méthode des différences finies utilisées d'ailleurs par Malik et Perona, où les opérateurs différentiels sont approchés à l'aide d'une formule de différences, la méthode des éléments finis qui jusqu'à présent n'a jamais été adoptée pour le modèle de Malik et Perona et que nous appliquons dans le présent travail, c'est l'image résultat qui est approchée.

2.1. Formulation variationnelle :

Soit φ une fonction test, on a:

$$\int_{\Omega} \frac{\partial I}{\partial t} \varphi = \int_{\Omega} div(c(\|\nabla I\|) \nabla I) \varphi$$
(8)

$$\int_{\Omega} div(c(\|\nabla I\|)\nabla I)\varphi = \int_{\Omega} \frac{\partial}{\partial x}(c(\|\nabla I\|)\frac{\partial I}{\partial x})\varphi + \int_{\Omega} \frac{\partial}{\partial y}(c(\|\nabla I\|)\frac{\partial I}{\partial y})\varphi$$
(9)

En utilisant la formule de Green:

$$\int_{\Omega} \frac{\partial u}{\partial x_{i}} (\vec{x}) v(\vec{x}) d\Omega = -\int_{\Omega} u(\vec{x}) \frac{\partial v}{\partial x_{i}} (\vec{x}) d\Omega + \int_{\Gamma} u(\vec{x}) v(\vec{x}) \eta(\vec{x}) (10)$$

On obtient

$$\int_{\Omega} \frac{\partial}{\partial x} (\alpha \|\nabla I\|) \frac{\partial I}{\partial x}) \varphi + \int_{\Omega} \frac{\partial}{\partial y} (\alpha \|\nabla I\|) \frac{\partial I}{\partial y}) \varphi$$

= $-\int_{\Omega} \alpha \|\nabla I\| (\nabla I \cdot \nabla \varphi) + \int_{\Gamma} \varphi \alpha (\|\nabla I\|) (\frac{\partial I}{\partial x} n_x + \frac{\partial I}{\partial y} n_y)$ (11)

Comme $\frac{\partial I}{\partial n} = 0$ on a la formulation ionnelle suivante:

variationnelle suivante:

$$\begin{cases} \int_{\Omega} \frac{\partial I}{\partial t} \varphi = -\int_{\Omega} c(\|\nabla I\|) (\nabla I \cdot \nabla \varphi) \\ I(0, x, y) = I_0(x, y) \end{cases}$$
(12)

2.2 Approximation numérique

La construction des espaces d'approximations se fait en définissant d'abord ce qu'est un élément fini, puis en définissant des règles qui permettent d'assembler des éléments finis d'un même type, de façon à approcher des fonctions d'un ouvert Ω à valeurs réelles, Nous renvoyons le lecteur à [8] [9] [10] [11] [12] aux définitions et propriétés qui suivent.

2.3 Définitions et propriétés

On se donne

1. Un compact K de IR^n , connexe et d'intérieur non vide.

2. Un ensemble $\sum = \{a_1, ..., a_N\}$ de *N* points distincts (*N* degrés de liberté).

3. Un espace P de dimension finie N de fonctions définies sur K à valeurs réelles.

On dit que l'ensemble \sum est P-unisolvant si et seulement, étant donné N scalaires quelconques α_j 1 \leq j \leq N, il existe une fonction pde l'espace P et une seule telle que

$$p(a_j) = \alpha_j \quad 1 \le j \le N.$$

Lorsque l'ensemble Σ est P-unisolvant, le triplet (K, Σ, P) est appelé élément fini de Lagrange.

Etant donné un élément fini (K, Σ, P) , il existe donc N fonctions φ_i de P qui satisfont $\varphi_i(a_j) = \delta_{ij}$, ses N fonctions sont appelées fonctions de base de l'élément fini.

Si on dispose de la base de *P* constituée des fonctions de base, alors le caractère unisolvant est automatiquement vérifié.

Deux éléments finis de Lagrange (K, \sum, P) et $(\hat{K}, \hat{\Sigma}, \hat{P})$ sont dits affinement équivalents s'il existe une application affine bijective F de \hat{K} sur K vérifiant:

1)
$$P = \left\{ p : K \to IR; p \circ F \in \hat{P} \right\}.$$

 $2)\sum = F(\hat{\Sigma}).$

Réciproquement l'image d'un élément fini par une application bijective continue vérifiant 1 et 2 est un élément fini.

Cette notion d'équivalence est très importante, car on peut générer à partir d'un élément fini une famille d'éléments finis qui lui sont affinement équivalents. Dans ce qui suit nous prenons comme élément fini de référence:

K : le triangle de sommet (0,0), (1,0), (0,1).

$$\Sigma = \{(0,0), (1,0), (0,1)\}.$$

$$P = \{p(x, y) = a + bx + cy, a, b, c \in IR\}.$$

Soit (K, Σ, P) l'élément fini de référence, on peut facilement calculer ses trois fonctions de base grâce à la relation suivante: $\varphi_i(S^j) = \delta_{ii}$. En effet on trouve:

$$\varphi_1(x, y) = 1 - x - y .$$

$$\varphi_2(x, y) = x .$$

$$\varphi_3(x,y)=y$$

Ainsi, nous pouvons déterminer facilement les fonctions de base d'un élément fini $(\hat{K}, \hat{\Sigma}, \hat{P})$ affinement équivalent à l'élément de référence (K, Σ, P) . En effet on utilise la propriété (2) de la définition précédente $(\Sigma = F(\hat{\Sigma}))$.

Maintenant que nous avons mis en place tous les outils des éléments finis pour l'élément de référence, il reste à définir des règles pour assembler des éléments de cette famille et définir ainsi un espace d'approximation de fonctions. Nous avons besoin pour cela définir ce qu'est un maillage.

2.4 Maillage

Soit Ω un ouvert à frontière polyédrique et Γ un recouvrement de Ω par des triangles. Nous dirons que Γ est un maillage de Ω , si pour deux triangles quelconques et différents K_1 et K_2 de Γ , l'intersection $K_1 \cap K_2$ est soit vide, soit un sommet commun, soit une arête commune.



Fig.1. Maillage représentant une image rectangulaire

En pratique, le maillage d'un domaine consiste à déterminer les tableaux suivants :

Coordonnés: une matrice de trois colonnes définies comme suit :

Colonne 1: contient les numéros des noeuds.

Colonne 2: contient les abscisses de chaque nœud dans la colonne 1.

Colonne 3: contient les ordonnées de chaque nœud dans la colonne 1.

Elément: contient les numéros des éléments finis (ici triangles) avec les numéros de degré de liberté de chaque élément.

Bord: contient les numéros des nœuds sur la frontière du domaine.

2.5 Construction de la solution approchée

L'espace d'approximation que nous considérons est:

$$V_h = \left\{ v_h \in C_0(\Omega), \ (v_h)_{/K} \in P, \ \forall K \in \Gamma \right\}$$

Nous introduisons les fonctions ϕ_i $1 \le j \le \mathbb{N}$ (\mathbb{N} est le nombre total de nœud), définies par: $\phi_i \in V_h$ $\phi_i(S^j) = \delta_{ij}$, D'après la propriété d'unisolvance, les fonctions ϕ_i sont définies de façon unique puisque dans chaque triangle K, la valeur de ϕ_i est prescrite en chacun des trois sommets. On constate immédiatement qu'il s'agit d'une base de V_h et on a :

$$(\phi_i)_{/K_j} = \varphi_k^{K_j}$$
 $k=1$ ou 2 ou 3, $(\phi_i)_{/K_j} \equiv 0$ si
 $S^i \notin K_j$.

On cherche alors I_h (solution approchée de l'équation de Malik et Perona) sous forme d'une combinaison linéaire des éléments de cette base, i.e sous la forme:



Fig.2. Fonction de base ϕ_i

Comme $\phi_i(S^j) = \delta_{ij}$, on a $\alpha_j(0)$ qui égale le niveau de gris de l'image au

sommet S^{j} . En pregnant comme fonction test dans la formulation variationnelle précédente une fonction de base ϕ_{i} on a :

$$\frac{\partial}{\partial t} \int_{\Omega} \sum_{j \in N} \alpha_j(t) \phi_j \phi_i = \frac{\partial}{\partial t} \sum_{j \in N} (\int_{\Omega} \phi_j \phi_i) \alpha_j(t)$$

$$= \frac{\partial}{\partial t} \sum_{j \in N} a_{ij}(t) \alpha_j(t) \quad \forall i \ge 1$$
(13)

Avec:

$$a_{ij} = \int_{\Omega} \phi_j \phi_i = \int_{\substack{k \in T_h \\ k \in T_h}} \phi_j \phi_i = \sum_{k \in T_h} \int_{K_k} \phi_j \phi_i \qquad (14)$$

On pose alors $A = (a_{ij})_{1 \le i,j \le m}$, Pour l'approximation numérique des intégrales on utilise la formule de quadrature de Gauss-Légendre:

$$\int_{K_j} h(x, y) dx dy \simeq \frac{aire(K_j)}{3} \sum_{i=1}^3 h(\hat{z}_i) \qquad (15)$$

Avec $(\hat{z}_i)_{1 \le i \le 3}$ Milieux des arêtes et $aire(K) = |\det(\begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_2 & y_2 \end{pmatrix})| \cdot$

Cette formule est exacte pour les polynômes de degré ≤ 2 .

L'assemblage de la matrice se résume à:

Boucle sur les éléments T_h k=1 à ne

Boucle sur les sommets i de l'élément K

Boucle sur les sommets j de l'élément *K*

Assemblage de la matrice A: $A_{ij} = A_{ij} + A_{ij}(K)$

Fin de boucle j.

Fin de boucle i.

Fin de boucle k.



Fig.3. Structure de la matrice creuse

La matrice résultante de cette méthode contient un très grand nombre de zéros, il est bien sûr hors de question dans ces conditions de stocker n termes de la matrice, pensons simplement à une image de 100^2 pixels, la matrice résultante aurait donc 10^8 termes, alors qu'il y aurait en effet moins de 70 000 termes non nuls (moins de 0.1%), on conçoit donc qu'il conviendrait d'optimiser le stockage de telles matrices au cours de l'assemblage.

2.6 Stockage de matrice creuse et résolution

a) Stockage de matrice creuse:

Pour profiter de la structure creuse de la matrice *A* (Réduction de la place mémoire et la réduction du nombre d'opérations : on n'effectuera pas les opérations portant sur les zéros) nous allons présenter une technique de stockage connue sous le nom CSR format (compressed sparce row format) [13] [14]. C'est l'une des techniques les plus connues et ce schéma est représenté par trois vecteurs:

-Le premier *AA* contient les éléments non nuls de la matrice *A* stockés dans l'ordre ligne par ligne; la dimension du vecteur *AA* est nz.

-Le deuxième JA (d'entiers) contient les indices colonnes correspondants aux éléments A dans AA, il est de taille nz.

-Le troisième vecteur IA (d'entiers) contient le pointeur de commencement de chaque ligne dans AA (et aussi dans JA), avec la dernière case de IA égale à nz+1.

Exemple:

	4	0	0	2	0]
	5	3	0	1	0	
4 =	8	0	6	7	10	
	0	0	9	12	0	
	0	0	0	0	11	

Cette matrice sera stockée sous la forme CSR comme suit:

Tableau 1 - Vecteurs	représentant	la	matrice	A
sous la forme CSR				

A A	4	2	5	3	1	8	6	7	1 0	9	1 2	1 1
J A	1	4	1	2	4	1	3	4	5	3	4	5

I A	1	3	6	1 0	1 2	1 3
--------	---	---	---	--------	--------	--------

b) Résolution

Produit d'une matrice par un vecteur *Y=A.X* :

Initialisation du vecteur Y par n zéros

For i=1 à n $K_1 = IA(i)$, $K_2 = IA(i+1)$ $Y(i) = \sum_{j=K_1}^{j=K_2} AA(j).X(JA(j))$

Fin (i)

Méthode du Gradient conjugué Pré conditionné:

Soit $A \in IR^{n \times n}$ une matrice symétrique définie positive et Y un vecteur quelconque. On veut résoudre le système linaire Ax = b, pour cela considérons la fonctionnelle quadratique $E: IR^n \to IR$ définie par :

$$E(x) = 1/2 \langle Ax, x \rangle - \langle b, x \rangle$$

Dont le gradient vaut $\nabla E(x) = (Ax - b)$. Le minimum de cette fonctionnelle est atteint pour le point \overline{x} qui annule $\nabla E(x)$, donc vérifiant $A\overline{x} = b$.

La méthode du gradient conjugué [13][15] fait partie des méthodes de descente, qui ont comme principe commun la recherche de \overline{x} suivant le procède itératif :

$$x^0$$
 donné, $x^{i+1} = x^i + \rho^i d^i$.

Avec $d^i \in IR^n$ et $\rho^i \in IR$ le pas de descente.

Le choix du paramètre de descente est simple, car une fois d^i et x^i fixé, on peut calculer facilement un pas de descente optimal

$$\rho^{i} = -\frac{\left\langle Ax^{i} - b, d^{i} \right\rangle}{\left\langle Ad^{i}, d^{i} \right\rangle}$$

Qui réalise le minimum de la fonctionnelle *E* dans la direction choisie.

Quant à la direction de descente, on choisit $d^i = h^i$, avec les directions de descente conjuguées par rapport à la matrice A, c'est-à-

dire
$$\langle Ah^i, h^j \rangle = 0$$
, $i \neq j$. On peut pré

conditionner cette méthode par une matrice C symétrique définie positive, la matrice C doit être la plus simple et la plus creuse possible ce qui impose comme choix les plus faciles:

-C = I, la matrice identité (algorithme sans pré conditionnement).

 $-C = D^{-1}$, l'inverse de la partie diagonale de la matrice A.

Pour obtenir à la fin l'algorithme suivant:

$$x_0 \in IR^n$$
, ε , C étant donnés.

$$G^{0} = Ax^{0} - b$$

$$H^{0} = -CG^{0}$$

$$Pour \quad i=0 \text{ à } n$$

$$x^{i+1} = x^{i} + \rho H^{i}$$

$$G^{i+1} = G^{i} + \rho A H^{i}$$

$$\gamma = \frac{\left\langle G^{i+1}, G^{i+1} \right\rangle_{C}}{\left\langle G^{i}, G^{i} \right\rangle_{C}}$$

$$H^{i+1} = -CG^{i+1} + \gamma H^{i}$$
Si $\left\langle G^{i+1}, G^{i+1} \right\rangle \leq \varepsilon$ stop.

On remarque bien que pour manipuler cet algorithme pour les matrices stockées sous CSR format on a seulement besoin de savoir définir le produit d'une matrice par un vecteur, chose déjà faite supra.

2.7 Adaptation de l'assemblage de la matrice avec le schéma de stockage

Le principe de remplissage de la matrice A sous la forme définie précédemment consiste en un premier temps à remplir le tableau IA en remarquant que a_{ij} est non nul uniquement si $q^i q^j$ est une arête de la triangulation. La construction du tableau devient alors particulièrement simple, les points à l'intérieur du rectangle ont sept sommets communs, les bords ont cinq sommets communs et les coins en ont trois ou quatre.

La construction de AA et JA se fait d'une manière simultanée en parcourant la triangulation. Le calcul de AA(k) n'est pas fait en une fois : il est fait en additionnant les contributions élémentaires de chaque triangle, ce qui signifie que pour chaque couple (i,j) il faut retrouver l'indice k de stockage de a_{ij} dans le tableau AA.

2.8 Calcul du deuxième terme de l'équation

Comme *I* s'écrit sous forme d'une combinaison linéaire de fonctions de base elles mêmes linéaires on a:

$$I_{/K_i} = a_i(t)x + b_i(t)y + c_i(t)$$
$$\|\nabla I\|^2 = a_i^2(t) + b_i^2(t).$$

On détermine $a_i(t)$, $b_i(t)$, $c_i(t)$ en fonction de α_i de la façon suivante :

On note par T_i le tableau qui représente le triangle K_i (élément)

- e_x^i : Abscisse sommet de K_i
- e_{y}^{i} : Ordonnée sommet de K_{i}

Posons
$$P = \begin{pmatrix} e_x^i(1) & e_y^i(1) & 1 \\ e_x^i(2) & e_y^i(2) & 1 \\ e_x^i(3) & e_y^i(3) & 1 \end{pmatrix}$$
 on a alors

$$\begin{pmatrix} a_i(t) \\ b_i(t) \\ c_i(t) \end{pmatrix} = P^{-1} \begin{pmatrix} \alpha_1^i(t) \\ \alpha_2^i(t) \\ \alpha_3^i(t) \end{pmatrix}$$

:

Finalement $\|\nabla I\|$ ne dépend que de α .

Le principe de l'assemblage de la deuxième matrice se fait de la même manière que la première sauf qu'ici elle dépend de α_j . Nous obtenons ainsi le système différentiel suivant:

$$\begin{cases} \frac{d}{dt} A\alpha = g(\alpha) \\ \alpha(0) = \alpha_0 \end{cases}$$
(16)

Où g est une fonction qui dépend de la fonction c (terme non linéaire de l'équation de Malik Perona).

2.9 Résolution système différentielle

Les méthodes de Runge-Kutta [16], d'ordre 2 ou 4, sont très couramment utilisées pour la résolution des équations différentielles ordinaires. Elles ont l'avantage d'être simples à programmer et d'être assez stables pour les fonctions courantes de la physique. Sur le plan de l'analyse numérique [17], elles ont surtout l'immense avantage de ne pas nécessiter autre chose que la connaissance des valeurs initiales. Elles ont quand même un inconvénient, surtout la méthode Runge-Kutta d'ordre 4, qui est assez consommatrice en temps de calcul.

Soit
$$X: IR \to IR^n$$
 et

 $f: IR^n \times IR \rightarrow IR^n$, on veut résoudre le système différentiel suivant:

$$\begin{cases} \frac{dX(t)}{dt} = f(X(t), t) & t \ge t_0 \\ X(t) = X_0 & t = t_0 \end{cases}$$
(17)

La méthode Runge-Kutta utilise plusieurs points intermédiaires pour calculer la valeur de X_{i+1} à partir de la valeur de X_i :

$$f_{0} = f(X_{0}, t_{0})$$

$$f_{1} = f(X_{0} + \frac{h}{2}f_{0}, t_{0} + \frac{h}{2})$$

$$f_{2} = f(X_{0} + \frac{h}{2}f_{1}, t_{0} + \frac{h}{2})$$

$$f_{3} = f(X_{0} + hf_{2}, t_{0} + h)$$

$$X(t_{0} + h) = X_{0} + \frac{h}{6}(f_{0} + 2f_{1} + 2f_{2} + f_{3})$$
(18)

IV. Présentation des Schémas de Résolution par Différences Finies et Semi-implicites de la Méthode de Malik et Perona:

1.Résolution explicite de la mise en œuvre de la Méhode de Malik et perona :

Malik et Perona [6] proposent la discrétisation explicite suivante:

$$u_{i,j}^{t+\Delta t} = u_{i,j}^{t} + \Delta t (c_{E_{i,j}}^{t} \nabla_{E} u_{i,j}^{t} + c_{W_{i,j}}^{t} \nabla_{W} u_{i,j}^{t} + c_{N_{i,j}}^{t} \nabla_{N} u_{i,j}^{t} + c_{S_{i,j}}^{t} \nabla_{S} u_{i,j}^{t}) (19)$$

Où *N*, *S*, *E*, *W* représentent les directions spatiales, nord, sud, est, ouest. Le symbole ∇ dénote le gradient spatial dans la direction indiquée par l'indice et les cœfficients C sont définis par :

$$c_{N_{i,j}}^{t} = c(\left|\nabla_{N}u_{i,j}^{t}\right|), c_{W_{i,j}}^{t} = c(\left|\nabla_{W}u_{i,j}^{t}\right|),$$

$$c_{E_{i,j}}^{t} = c(\left|\nabla_{E}u_{i,j}^{t}\right|), c_{S_{i,j}}^{t} = c(\left|\nabla_{S}u_{i,j}^{t}\right|)$$
(20)

2. Résolution semi-implicite de la mise en œuvre de la Méhode de Malik et Perona:

La facilité d'implémentation des schémas explicites est payée en général par un pas d'intégration très petit pour ne pas perturber la stabilité numérique, ce qui introduit un grand nombre d'itérations avant d'aboutir à la convergence [18] [19]. Weikert [20] a introduit un schéma semi implicite d'intégration nommé Additive Operator Splitting (AOS) avec une indépendant stabilité absolue. du pas d'intégration. Pour pouvoir utiliser ce schéma, il faut que cette EDP soit écrite sur le modèle suivant:

$$\partial_t u = div(z(\|\nabla u\|)\nabla u) = \sum_{l=1}^m (\frac{\partial}{\partial x_l} z(\|\nabla u\|) \frac{\partial u}{\partial x_l}) (21)$$

Où *l* est l'indice indiquant la dimension jusqu'à l'indice maximal. Pour une image 2D ($x_1 = x$, $x_2 = y$), Le schéma semi implicite s'écrit alors:

$$u^{k+1} = \frac{1}{m} \sum_{l=1}^{m} \left[I - (m\Delta t) A_l(u^k) \right]^{-1} u^k \quad (22)$$

Avec
$$A_l(u^k) = \frac{\partial}{\partial x_l} \left(z(\|\nabla u\|) \frac{\partial}{\partial x_l} \right)$$
, k le numéro

d'itération en cours et $u_0 = u^0$. Pour obtenir la solution du schéma on résout un système d'équations linaires par l'algorithme de Thomas [5].

V. Résultats

Pour la comparaison, nous allons utiliser le critère quantitatif « rapport signal sur bruit (SNR) » qui s'exprime en décibels et défini par:

$$SNR = 10.\log_{10}\left(\frac{|u|^2}{|u-v|^2}\right)$$
(23)

Avec u une image de référence et v l'image après traitement. Nous précisons aussi que le SNR est calculé sur des images centrées sur la moyenne des niveaux de gris.

Le graphique suivant montre les résultats de la comparaison de la variation du rapport signal/ bruit pour les différents schémas numériques



Fig.4. Comparaison de la variation du rapport signal à bruit pour les différents schémas numériques.

La variation des valeurs du rapport signal sur bruit obtenue par la méthode des éléments finis en fonction du nombre d'itérations décroît d'une manière stable et correcte: les plus grandes valeurs sont obtenues au début du processus (entre 10 et 20 itérations), par contre quand le nombre d'itérations croit, la variation se fait d'une façon très douce ce qui montre que notre algorithme devient de plus en plus stable.

En ce qui concerne la comparaison utilisant le SNR, de notre discrétisation par éléments finis, avec la discrétisation par différences finies de Malik et Perona et de la méthode semi implicite, il ressort que les trois méthodes se comportent presque de la même manière et donnent quasiment les mêmes résultats avec un écart correct au début du processus dû à leurs différences théoriques.

Les images qui suivent présentent les résultats obtenus sur des images synthétiques et réelles.



Fig.5. (a) Image synthétique fortement bruitée (512x512) (b) Résolution par l'équation d'éléments finis (100 itérations dt=0.1 et k=0.005), (c) Résolution par éléments finis (50 itérations dt=0.1 et k=0.1), (d) Solution par éléments finis (30 itérations dt=0.1 et k=0.2).

Nous remarquons (Fig. 5) que notre méthode supprime d'une façon efficace le bruit, et que le paramètre de diffusion k joue un rôle important sur la qualité de restauration.





Fig.6. (a) Image réelle bruitée (256x256), (b) solution obtenue par discrétisation de Malik-Perona 50 itération k=0.01, (c) solution obtenue par discrétisation semi implicite 50 itération k=0.01, (d) solution obtenue par éléments finis 50 itération k=0.01.

Nous remarquons (Fig.6) que notre méthode préserve plus de détails de l'image en effet on voit bien que les doits du caméraman sont dégradés dans la résolution explicite (Fig. 6 (b)) et semi implicite (Fig. 7 (c)) alors qu'il ne l'est pas pour la résolution éléments finis (Fig. 7 (d)).





Fig.7. a) Image réelle bruitée (256x256), (b) solution obtenue par discrétisation de Malik-Perona 150 itération k=0.005, (c) solution obtenue par discrétisation semi implicite 150 itération k=0.005, (d) solution obtenue par éléments finis 150 itération k=0.005.

Il est clair que d'une part, notre méthode (Fig. 7 (d)) supprime plus de bruit comparé à la méthode explicite (Fig. 7 (b)), et d'autres part, elle laisse plus intacts les détails dans l'image par rapport à la méthode semi implicite (Fig. 7 (c)) qui rend de plus l'image floue (regarder la lettre w).

Notre méthode est très efficace pour les bruits de type gaussien, Toute fois pour les bruits impulsionnels, il faut appliquer en premier temps quelques itérations de l'équation de la chaleur (2 à 4 itération) afin de diminuer les fortes perturbations du bruit puis appliquer notre méthode.

Pour les images couleurs plusieurs normes du gradient existent dans la littérature [21], l'extension de notre méthode est envisageable avec un bon choix de la norme du gradient.

VI. Conclusion

Nous avons rappelé dans ce papier le principe de la diffusion anisotrope et la méthode des éléments finis. Nous avons proposé une nouvelle mise en œuvre numérique de l'EDP de Malik-Perona par la méthode des éléments finis. Les résultats obtenus sont très satisfaisant.

La comparaison de notre méthode avec les autres schémas numériques permet de dire que notre algorithme est stable, donne des meilleurs résultats.

VII. Références

- [1] R. Deriche, O. faugeras, les EDP en traitement des images et vision par ordinateur. Rapport techniques no. 2697, *institut national de recherche en informatique et automatique*, France, 1995.
- [2] L. Alvarez, F. Guichard, P.L. Lions, and J.M. Morel, Axioms and fundamental equations in image processing, *Archive for Rational Mechanics and Analysis*, vol. 123, no 3, (1993) 200–257.
- [3] Samir Befkih, L'application des équations aux dérivées partielles en traitement d'images, *thèse de doctorat de l'université de Montpelier*, France 2002.
- [4] L. Alvarez, F. Guichard, P.L. Lions, and J.M. Morel, Analyse multi-echelle de films. C. R. Acad. Sci, vol 315, no 1, (1993), 1145-1148.
- [5] L. Alvarez, P. Lions, and J. Morel, Image selective smoothing and edge detection by nonlinear diffusion, *SIAM Journal of Numerical analysis*, vol. 29, no. 3, (1992) 845-866.
- [6] P. Perona, J. Malik , Scale space and edge detection using anisotropic diffusion, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, (1990) 629-639.
- [7] S. Osher and L. Rudin, Feature oriented image enhancement using shock filter, *SIAM Journal for Numerical Analysis*, vol. 27, no.4, (1990) 919-940.
- [8] O. Pironneau, B. Lucquin, Introduction au calcul scientifique, *Masson*, 1997.
- [9] <u>Pavel Solín</u> Partial Differential Equations and the Finite Element Method, *Wiley*, 2005.
- [10] P.A. Raviart, J.M Thomas Introduction à l'analyse numérique des équations aux dérivées partielles, *Masson*, 1988.

- [11] O.C. Zienkiewicz, R. L. Taylor, *The Finite Element Method The Basis*, Vol. 1, Elsevier 2000.
- [12] V. Girault et P. A. Raviart, Finite element methods for Navier-Stokes equations: Theory and algorithms, *Springer Series in Computational Mathematics*, vol. 5, 1986.
- [13] Y. Saad Iterative Methods for Sparse linear systems, *Society for Industrial and Applied Mathematics*, 2003.
- [14] I. S. Duff, A. M. Erisman, J. K. Reid, Direct Methods for Sparse Matrices, Oxford University Press, 1986.
- [15] P. Lascauw et R. Theodor, analyse numérique matricielle appliqué à l'art de l'ingénieur, *Dunod*, 2004.
- [16] J. P. Demailly Analyse Numérique et équations différentielles, EDP sciences, 1990.
- [17] M. Sibony et J. Mardon, Approximations et équations différentielles, *Hermann*, 1988.
- [18] J. Sethian, Level Set Methods, *Cambridge University Press*, 1996.
- [19] S.Belfkih, un nouveau schéma numérique pour la mise en œuvre des équations aux dérivées partielles en traitement d'images, *Proceedings of 3rd International conference sciences of electronic ,technologies of information and telecommunication*, 2005.
- [20] J. Weickert, B. M. Ter Haar Romeny, M. Viergever, Efficient and reliable schemes for nonlinear diffusion filtering, *IEEE Transactions on Image Processing*, vol. 7 no.3, (1998) 398–410.
- [21] A. Trémeau, C. Fernandez-Maloigne, P. Bonton, Image numérique couleur, *Sciences sup*, 2004.