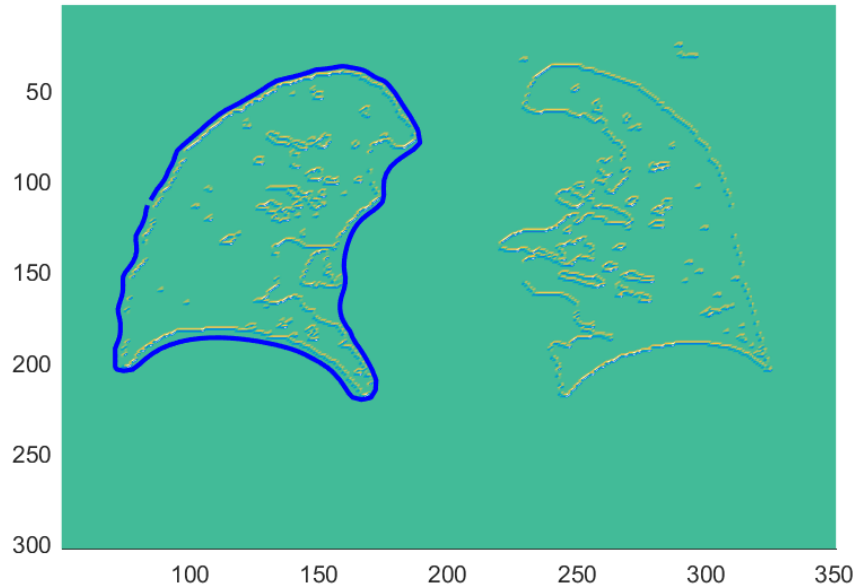


Segmentation d'images médicales par contours actifs

Mathieu Amendola

Tuteurs : Fabien Pierre et Pierre-Frédéric Villard



Rapport de projet 3A

8 Février 2018

Table des matières

Remerciements	3
1 Introduction	4
2 Résolution théorique en 2D	6
2.a) Paramétrage du problème	6
2.a).1 Présentation	6
2.a).2 Energie interne	6
2.a).3 Energie extérieure	6
2.b) Minimisation de l'énergie totale	7
2.c) Discrétisation spatiale	7
2.c).1 Différences finies	7
2.c).2 Application au problème	8
2.d) Discrétisation temporelle	9
2.e) Explication de f	9
2.f) Force <i>ballon</i>	9
2.g) Seuillage	11
3 Résolution numérique en 2D	13
3.a) Principe	13
3.b) Simulations	13
3.c) Impact et nécessité de chacune des forces	16
3.d) Définition de l'erreur	20
3.e) Paramètres de régularité	21
3.f) Force <i>ballon</i>	23
3.g) Vitesse de convergence	23
3.h) Flou gaussien	27
3.i) Application sur des images médicales	29
4 Contour actif en 3D	31
4.a) Analogies avec la résolution en 2D	31
4.b) Définition de la matrice A	31
4.c) Equation finale en 3D	33
4.d) Explication de f	33
5 Conclusion	34

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué à la réalisation de ce projet.

J'adresse en premier lieu mes remerciements à mes tuteurs de projet, **Fabien Pierre** et **Pierre-Frédéric Villard**. Ils m'ont guidé tout au long de ce projet en me prodiguant des conseils et en me proposant des pistes d'amélioration dans le but d'obtenir les résultats souhaités. Je les remercie également pour leur accueil au Loria et le partage de leurs connaissances.

Je tiens également à remercier **Sandie Ferrigno** pour m'avoir proposé ce sujet de projet, intéressant dans son contenu et ses applications.

1 Introduction

L'informatique et la simulation dans le domaine de la santé permettent à la fois de participer à la formation des chirurgiens et de planifier des opérations chirurgicales. En effet, une simulation permet de se placer dans des conditions très proches de la réalité. Ainsi, le chirurgien pourra s'entraîner pour être plus efficace sur une opération réelle. Le but de cette branche est donc de mettre en place des simulateurs chirurgicaux. Des entreprises telles que 3D Systems [1] ou Mentice [2] proposent déjà des simulateurs variés qui permettent aux chirurgiens de s'exercer. Des appareils permettant de simuler par exemple une embolisation [3] (procédure visant à prévenir ou stopper un saignement) ou une endoscopie [4] (examen d'une cavité inaccessible à l'oeil) existent donc déjà. D'autres travaux, spécialisés dans la simulation des muscles respiratoires, sont en cours, comme par exemple ceux menés par Tominec et al. [5].

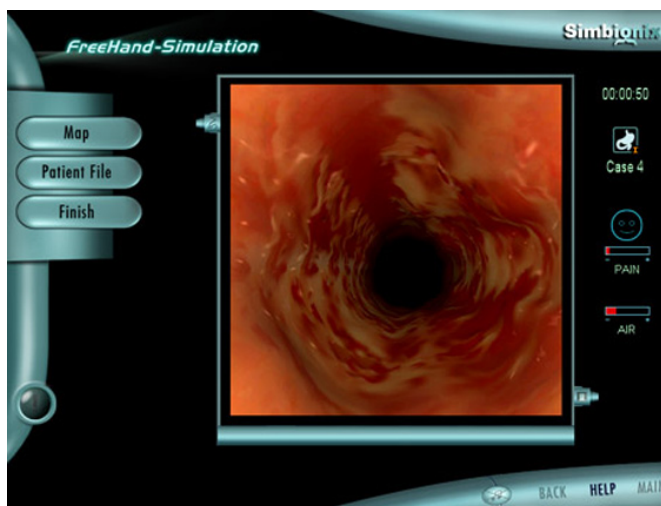


FIGURE 1 – Simulation d'une endoscopie

Alors se pose le problème de l'adaptation. En effet, pour être pertinente, la simulation doit s'adapter à chaque patient, à chacune des différences physiques et morphologiques des personnes qui vont être opérées. Pour répondre à cette problématique, la solution fournie par ce projet est la segmentation numérique et automatique des organes nécessaires sur des images médicales (obtenues par le biais de la radiologie). Avec cette segmentation, les organes du patient seront ainsi bien identifiés et permettront une préparation efficace à l'opération en adaptant la simulation. Nous nous intéresserons ici à la segmentation de l'appareil respiratoire. Plus spécifiquement, c'est le diaphragme, ce muscle qui permet à l'air d'arriver dans les voies respiratoires et également l'expiration passive, qui nous intéresse.

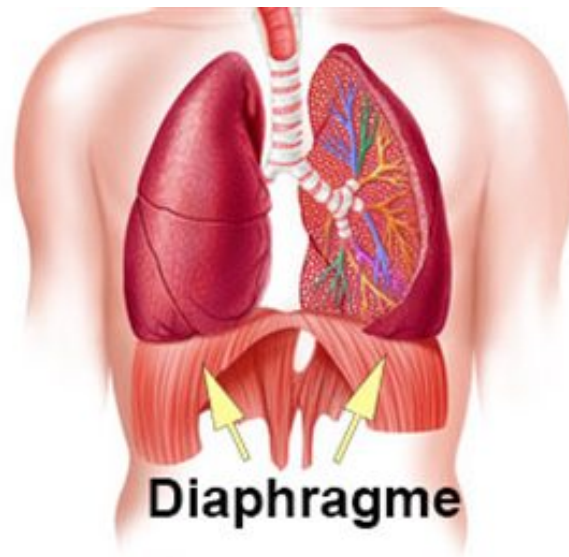


FIGURE 2 – Position du diaphragme dans le corps humain

La segmentation d'images consiste à traiter une image dans le but d'obtenir une carte binaire à partir de l'image initiale, composée d'une part de la forme que l'on souhaite obtenir, de l'autre du reste de l'image.

Nous utiliserons dans ce projet la méthode de segmentation par contour actif, aussi appelée *snake*. Le but de cette méthode de segmentation d'images est de modéliser une courbe qui épousera la forme de l'objet à segmenter.

Pour cela, nous allons commencer par faire une résolution dans un espace à deux dimensions. La résolution théorique nous donnera les bases pour généraliser ce problème en trois dimensions, tandis que la résolution numérique nous fournira des informations importantes sur le choix des paramètres.

Nous nous appuyerons pour cela sur trois articles traitant de la théorie du *snake* [6] et son application en 3D [7] [8].

2 Résolution théorique en 2D

2.a) Paramétrage du problème

2.a).1 Présentation

Le but du modèle de contour actif nommé *snake* va être de trouver le contour d'une image. Pour obtenir ce contour, l'objectif est de minimiser une somme de plusieurs énergies représentant les forces soumises au contour. On travaille pour l'instant dans un espace à deux dimensions.

Dans le modèle présenté par l'article De Michael Kass et Andrew Witkin [6], il y a deux forces qui agissent sur le contour :

- une force interne à minimiser pour lisser par morceaux le contour
- une force imposée par l'image à minimiser pour que le contour se rapproche de l'image

Notons E_{int} l'énergie interne, E_{ext} l'énergie extérieure.

On va représenter le *snake* à la position s par $v(s) = (x(s), y(s))$. x et y sont donc les vecteurs des coordonnées des points du snake.

La problématique est ainsi de discrétiser un chemin continu de \mathbb{R}^2 .

On commence par exprimer l'énergie totale à minimiser, notée E_s .

$$E_s = \int_0^1 E_{int}(s) + E_{ext}(s) ds$$

2.a).2 Energie interne

Dans tout le problème, les dérivées du vecteur position du *snake* seront définies coordonnée par coordonnée, avec la discrétisation que nous verrons plus tard.

L'énergie interne peut s'écrire $E_{int} = \frac{1}{2} (\alpha(s) || v'_s(s) ||^2 + \beta(s) || v''_s(s) ||^2)$, $\alpha(s)$ et $\beta(s)$ étant des poids variables qui permettent de contrôler la régularité du contour.

Les poids α et β seront considérés constants afin de simplifier le problème et de le discrétiser, ce qui permettra une application numérique.

2.a).3 Energie extérieure

Notons $E_{ext} = E_{con} + E_{im}$. Nous allons considérer que cette énergie est exclusivement représentée par le gradient de l'image. En effet, pour permettre au contour de se rapprocher de l'image, il faut le pousser vers les zones à fort gradient qui représentent le contour de l'image. Il faut donc maximiser le gradient. Cette énergie extérieure peut donc être représentée par le potentiel de l'image.

On peut ainsi écrire $E_{ext} = -P(v(s))$.

Remarque 2.1. Dans notre résolution numérique, on prendra $P(v(s)) = || \nabla I ||^2 (v(s))$.

2.b) Minimisation de l'énergie totale

On peut maintenant écrire le problème sous la forme :

$$\min_v \int_0^1 \frac{1}{2} (\alpha \|v'_s(s)\|^2 + \beta \|v''_s(s)\|^2) - P(v(s)) ds$$

Le but est de trouver le contour qui minimise cette équation. Pour la résoudre, on va appliquer le théorème d'Euler-Lagrange.

Théorème 2.2. *Euler Lagrange :*

$$\min_v \int_0^1 G(s, v, v', v'') ds \Leftrightarrow \frac{\partial G}{\partial v} - \frac{\partial}{\partial s} \frac{\partial G}{\partial v'} + \frac{\partial^2}{\partial s^2} \frac{\partial G}{\partial v''} = 0$$

Par analogie avec notre problème, on va écrire les différents termes de G.

Le terme en v est $-P(v(s))$.

Ainsi, $\frac{\partial G}{\partial v} = -\nabla(P(v(s)))$

Le terme en v' est $\frac{1}{2}\alpha \|v'_s(s)\|^2$.

Ainsi, $\frac{\partial}{\partial s} \frac{\partial G}{\partial v'} = \alpha v^{(2)}(s)$.

Le terme en v'' est $\frac{1}{2}\beta \|v''_s(s)\|^2$.

Ainsi, $\frac{\partial^2}{\partial s^2} \frac{\partial G}{\partial v''} = \beta v^{(4)}(s)$.

On obtient donc finalement l'équation suivante :

$$\alpha v^{(2)}(s) - \beta v^{(4)}(s) - \nabla(P(v(s))) = 0$$

On va maintenant discrétiser le problème dans le but de réaliser une application numérique.

On va ainsi prendre un contour s composé de n points indexés de 1 à n , représentés par les coordonnées v_1, \dots, v_n .

2.c) Discrétisation spatiale

2.c.1 Différences finies

On utilise les méthodes d'Euler implicite et explicite pour discrétiser les dérivées, en centrant la discrétisation d'un certain point k autour de ce point.

f représente dans cette section une fonction discrète quelconque.

Propriété 2.3. *Méthode d'Euler implicite :*

$$\frac{\partial f}{\partial x_k} \simeq f(x_k) - f(x_{k-1})$$

Propriété 2.4. *Méthode d'Euler explicite :*

$$\frac{\partial f}{\partial x_k} \simeq f(x_{k+1}) - f(x_k)$$

Appliquées à l'ordre 2, ces méthodes donnent :

$$\frac{\partial^2 f}{\partial x_k^2} \simeq f(x_{k+1}) - 2f(x_k) + f(x_{k-1})$$

A l'ordre 3, on a :

$$\frac{\partial^3 f}{\partial x_k^3} \simeq f(x_{k+2}) - 3f(x_{k+1}) + 3f(x_k) - f(x_{k-1})$$

Enfin, à l'ordre 4, on obtient :

$$\frac{\partial^4 f}{\partial x_k^4} \simeq f(x_{k+2}) - 4f(x_{k+1}) + 6f(x_k) - 4f(x_{k-1}) + f(x_{k-2})$$

2.c).2 Application au problème

On va donc pouvoir définir des matrices de passage pour passer du point au temps t à celui au temps $t+1$. Notons A_2 la matrice tridiagonale (dérivée seconde) et A_4 la matrice pentadiagonale (dérivée quatrième).

On rajoute des conditions au bord, nécessaires pour l'algorithme.

On s'attend à ce que le contour soit fermé, on peut donc définir $v_0 = v_n$.

$$A_2 = \begin{pmatrix} -2 & 1 & 0 & \dots & 0 & 1 \\ 1 & -2 & 1 & \dots & 0 & 0 \\ 0 & 1 & -2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -2 & 1 \\ 1 & 0 & 0 & \dots & 1 & -2 \end{pmatrix}$$

$$A_4 = \begin{pmatrix} 6 & -4 & 1 & \dots & 1 & -4 \\ -4 & 6 & -4 & \dots & 0 & 1 \\ 1 & -4 & 6 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 0 & \dots & 6 & -4 \\ -4 & 1 & 0 & \dots & -4 & 6 \end{pmatrix}$$

Pour s'adapter au problème, notons naturellement la matrice $A = -\alpha A_2 + \beta A_4$.

On note de plus $f(x, y) = \nabla(P(x, y))$.

Le problème peut désormais s'écrire :

$$\begin{cases} Ax + f(x, y) = 0 \\ Ay + f(x, y) = 0 \end{cases}$$

On va pouvoir encore simplifier l'équation. On va noter :

$$A = \begin{pmatrix} A & 0 \\ 0 & A \end{pmatrix} \quad V = \begin{pmatrix} x \\ y \end{pmatrix}$$

On arrive finalement à l'équation finale à résoudre :

$$AV + f(V) = 0$$

2.d) Discrétisation temporelle

L'objectif est d'obtenir une discrétisation temporelle pour avoir un algorithme itératif afin que le contour se rapproche petit à petit de l'image.

On va ainsi noter V_t le vecteur V au temps t .

On a ici un problème de point fixe.

$$AV + f(V) = 0 \iff AV + f(V) + V = V$$

Notons $g(V) = AV + f(V)$.

Notons γ le pas de temps de la discrétisation temporelle d'Euler.

Si on discrétise par le biais d'une méthode d'Euler implicite, on obtient $g(V_n) = \frac{1}{\gamma}(V_n - V_{n-1})$.

Si on discrétise par le biais d'une méthode d'Euler explicite, on obtient $g(V_{n-1}) = \frac{1}{\gamma}(V_n - V_{n-1})$.

On va choisir d'utiliser la méthode implicite pour AV , et la méthode explicite pour $f(V)$.

On obtient finalement le problème suivant :

$$AV_t + f(V_{t-1}) = -\frac{1}{\gamma}(V_t - V_{t-1})$$

En inversant la matrice $A + \gamma I_d$ (presque pentadiagonale), on obtient finalement V_t en fonction de V_{t-1} .

$$V_t = (\gamma A + I_d)^{-1} (V_{t-1} - \gamma f(V_{t-1}))$$

2.e) Explication de f

Nous avons besoin de discrétiser le gradient pour pouvoir appliquer la formule de f à notre problème. Nous allons pour cela utiliser la formule suivante. Notons N le nombre de points et g une fonction discrète quelconque à 2 dimensions [9].

$$\begin{cases} \frac{\partial g_{x,y}}{\partial x} \simeq g_{x+1,y} - g_{x,y} & \text{si } x < N \\ 0 & \text{si } x = N \end{cases}$$

De même, en y , nous avons :

$$\begin{cases} \frac{\partial g_{x,y}}{\partial y} \simeq g_{x,y+1} - g_{x,y} & \text{si } y < N \\ 0 & \text{si } y = N \end{cases}$$

Nous allons utiliser cette discrétisation pour calculer numériquement le gradient. Nous allons commencer par l'appliquer directement à l'image. On calcule ensuite point par point des 2 matrices obtenues. Enfin, on applique de nouveau à la matrice obtenue la discrétisation selon x et y pour obtenir la fonction f .

2.f) Force *ballon*

Cependant, comme on peut le voir sur la figure ci-dessous, cette résolution théorique présente des limites. En effet, sur les parties concaves de la figure, les forces de lissage vont être plus fortes que la force du gradient. Ainsi, le *snake* ainsi construit n'épousera pas les parties concaves des formes voulues.

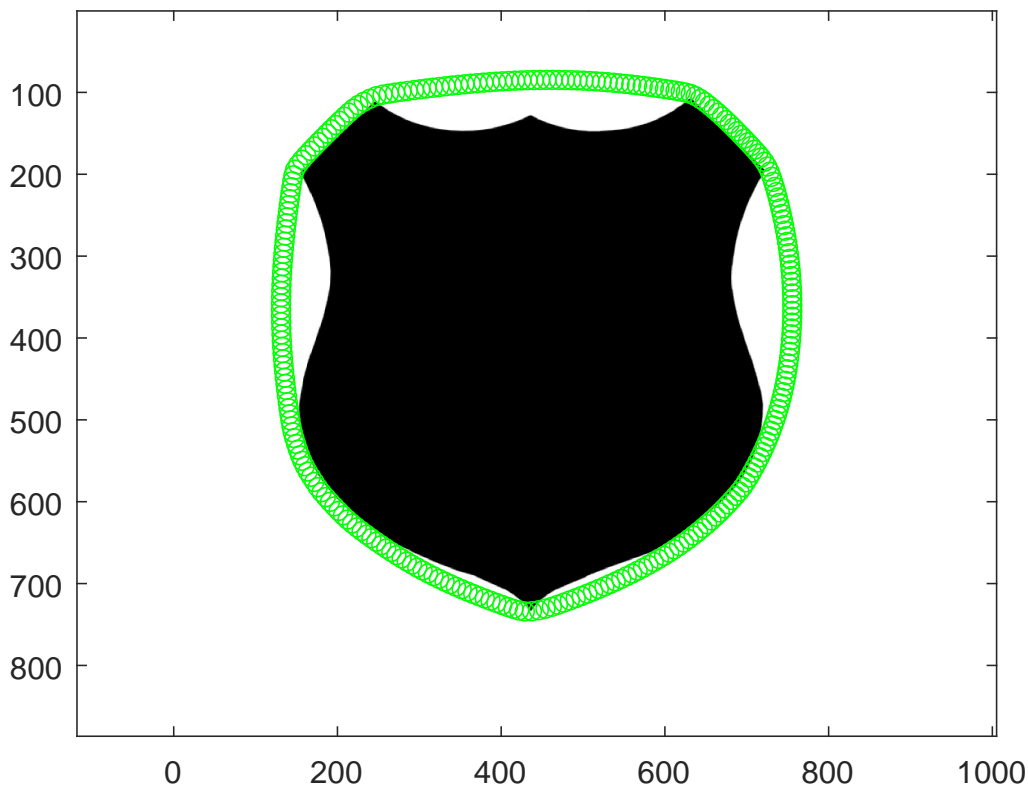


FIGURE 3 – Résultat final sur un exemple simple

On va donc améliorer la fonction f définie précédemment. On va pour cela rajouter une force, que l'on pourra appeler force *ballon* en s'appuyant sur les travaux de Cohen [10], qui permettra au contour de se contracter naturellement, même s'il n'est pas attiré par un gradient. Cela permettra au contour actif de continuer à se rapprocher de la forme voulue, en s'opposant aux forces de régularité.

Cette force *ballon* va être construite en s'appuyant sur la normale du *snake*. On va donc commencer par calculer la normale en chaque point. Pour cela, on va calculer la normale du vecteur entre le point précédent et le point suivant du contour actif. Si ce vecteur a pour coordonnées $\begin{pmatrix} x \\ y \end{pmatrix}$, on choisit naturellement le vecteur $\begin{pmatrix} -y \\ x \end{pmatrix}$ pour avoir un contour qui se contracte, $\begin{pmatrix} y \\ -x \end{pmatrix}$ pour un contour qui se dilate.

Il suffit ensuite de calculer cette normale pour chaque point du cercle, ce qui nous donne une force $F_B = \begin{pmatrix} B_x \\ B_y \end{pmatrix}$. En écrivant par la suite $f(x, y) = \nabla(P(x, y)) + k_B F_B(x, y)$, on peut avoir une résolution numérique plus juste. Le k_B est le coefficient de la force *ballon*, très important à quantifier pour obtenir le résultat souhaité.

Remarque 2.5. On avait précédemment défini f comme les forces extérieures appliquées au contour actif. Cependant, la force *ballon* est une force interne. La fonction f est donc la somme d'une force externe et d'une force interne.

2.g) Seuillage

Dans certains cas, comme par exemple celui de l'imagerie médicale, certaines formes peuvent attirer le contour actif sans que cela soit voulu. Cela peut entraîner une perturbation et engendrer une erreur dans les résultats obtenus.

Dans l'image ci-dessous, on voudrait segmenter le poumon gauche. Cependant, les côtes sur le bord gauche du poumon (formes circulaires de couleur blanche sur l'image) peuvent attirer le *snake* à cause de leur gradient.

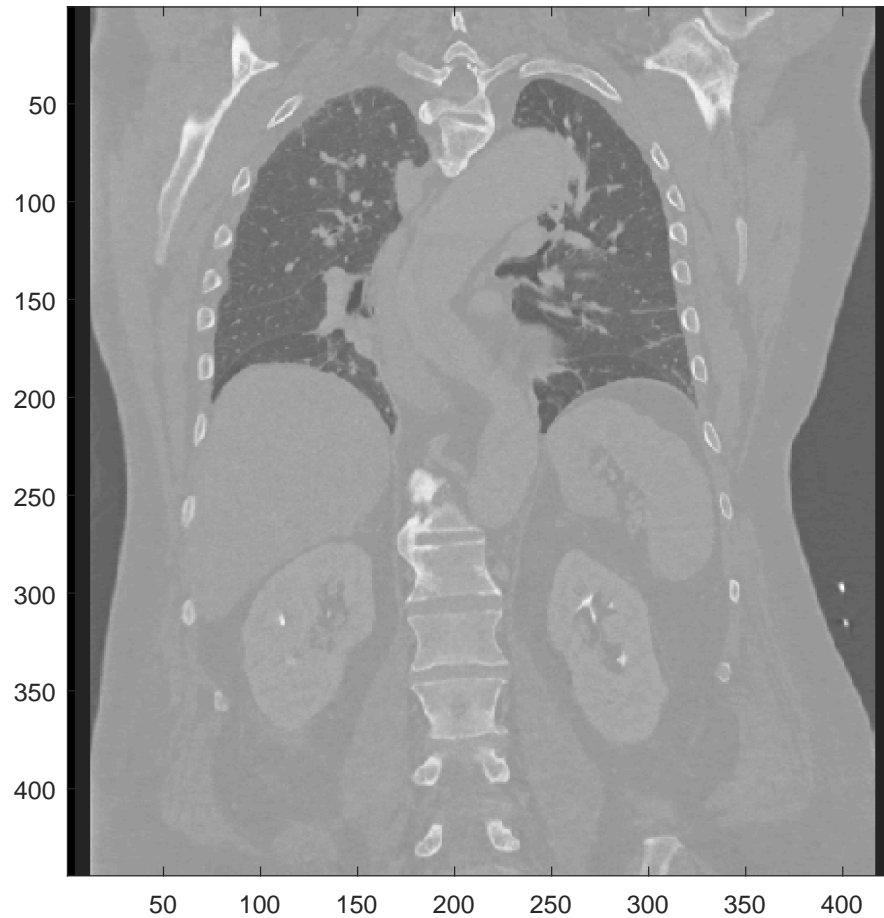


FIGURE 4 – Image médicale

Pour résoudre ce problème, on va effectuer un seuillage sur l'image de départ. Cela consiste à créer une image secondaire binaire sur laquelle s'effectuera l'algorithme. Pour cela, on va utiliser un paramètre λ , appelé paramètre de seuillage. Pour chaque pixel de l'image de départ dont la valeur est supérieure à ce paramètre, le pixel associé de l'image secondaire prendra la valeur maximale (255). Inversement, si la valeur du pixel de l'image initiale est inférieure à λ , le pixel de la nouvelle image sera égal à 0. On obtient dans cet exemple l'image ci-dessous, plus facile à segmenter.

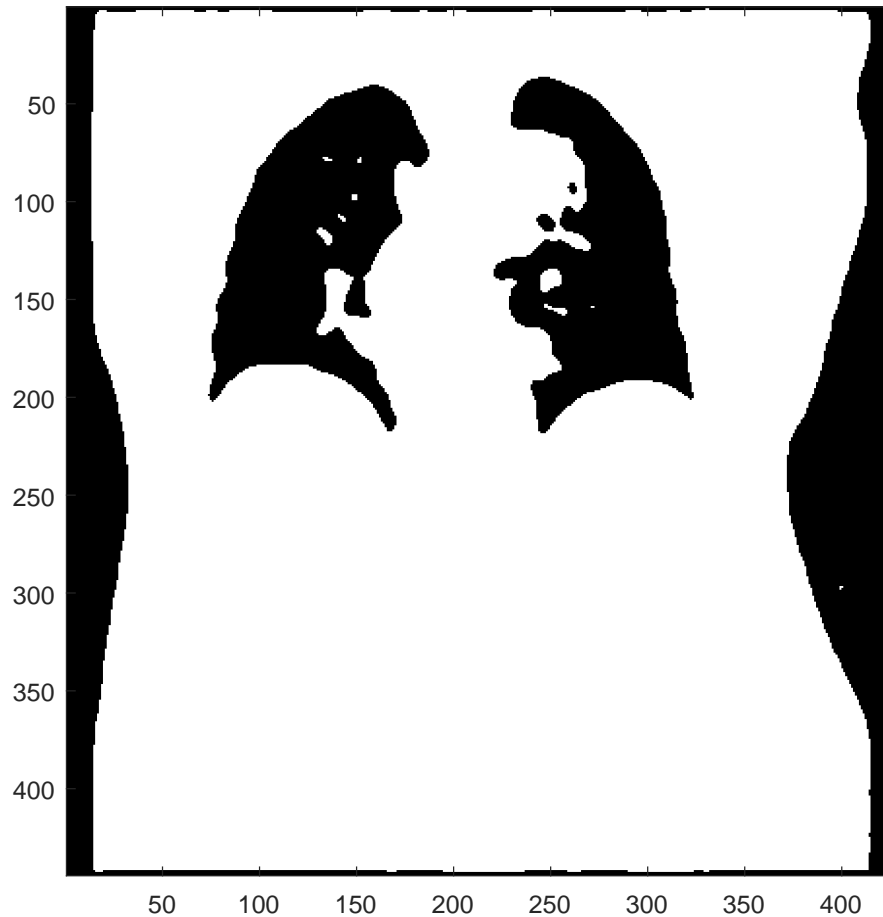


FIGURE 5 – Image médicale après seuillage

3 Résolution numérique en 2D

3.a) Principe

Nous allons appliquer les résultats que l'on a énoncé dans la partie théorique. Dans un premier temps, il faut commencer par importer l'image sur laquelle nous allons travailler. Comme ce sont des images de test qui ne sont *a priori* pas faites pour cela, nous les transformons en image en niveaux de gris. Nous appliquons ensuite un flou gaussien à l'image afin que la courbe puisse approcher plus facilement les contours.

Il faut ensuite créer la matrice A et les gradients vertical et horizontal dans le but de calculer les forces à minimiser. Il ne reste ensuite qu'à appliquer la formule d'itération. Nous utilisons un *while* afin d'avoir deux conditions d'arrêt, qui sont un nombre d'itérations maximal et une erreur minimale définie dans la suite.

Le programme prend en entrée les paramètres α , β , γ , σ (paramètre du flou gaussien), k_B , λ , les coordonnées du contour initial et le nom de l'image. Il renvoie en sortie le nombre d'itérations et l'erreur finale.

3.b) Simulations

On commence par initialiser le contour, comme illustré sur la figure ci-dessous. Pour cela, on utilise une fonction qui permet de choisir des points en cliquant sur la figure, puis une fonction qui va subdiviser les segments ainsi créés en un nombre de points qui dépend de la longueur en pixels du segment, dans le but d'avoir des espacements réguliers entre les points.

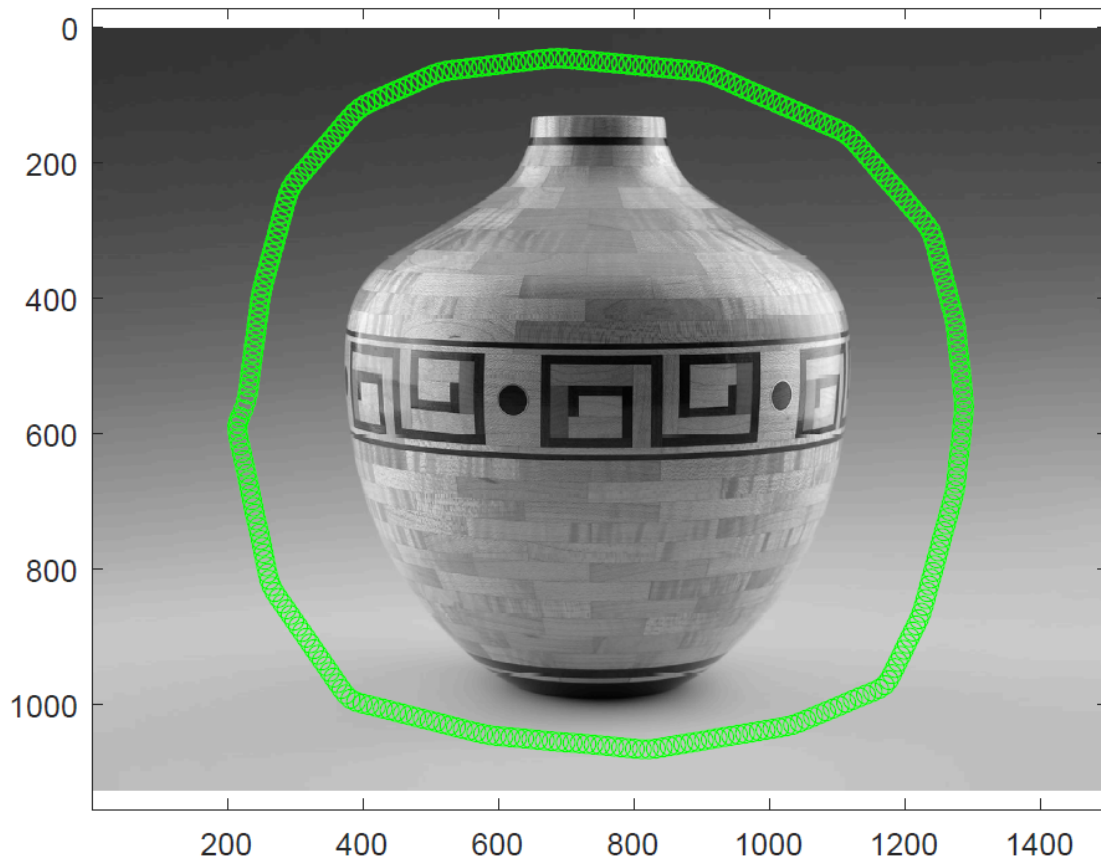


FIGURE 6 – Exemple d'initialisation

On peut voir que les points verts forment le contour initial. Les points sont situés à des intervalles réguliers, et sont assez nombreux pour que le contour puisse épouser la forme voulue. On peut également remarquer que le contour n'est absolument pas lisse et est très loin de la forme finale voulue.

On va prendre trois images en niveaux de gris pour tester les paramètres et la pertinence du modèle. On choisit pour cela les photos d'une pièce (forme simple, mais proche au niveau de la teinte de gris de l'arrière-plan), un vase (forme lisse qui présente cependant des parties concaves) et un chat (forme complexe dont la couleur est en plus proche de l'arrière-plan).

Les résultats du contour actif, obtenus en choisissant minutieusement les paramètres, sont montrés ci-dessous.

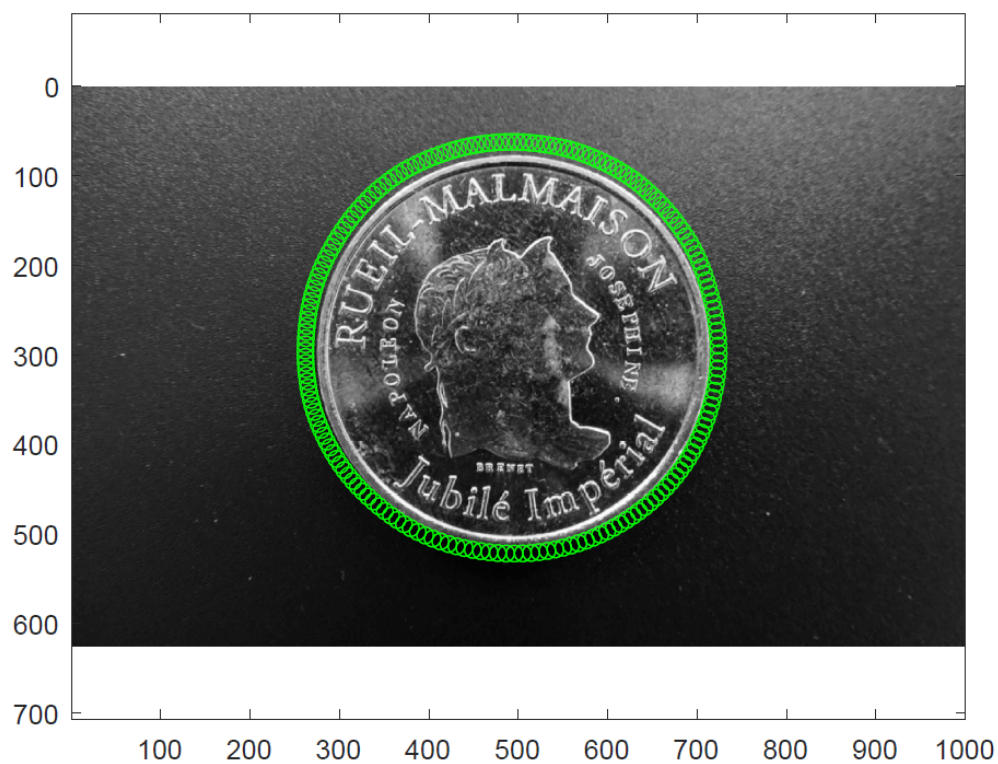


FIGURE 7 – Contour actif d'une pièce

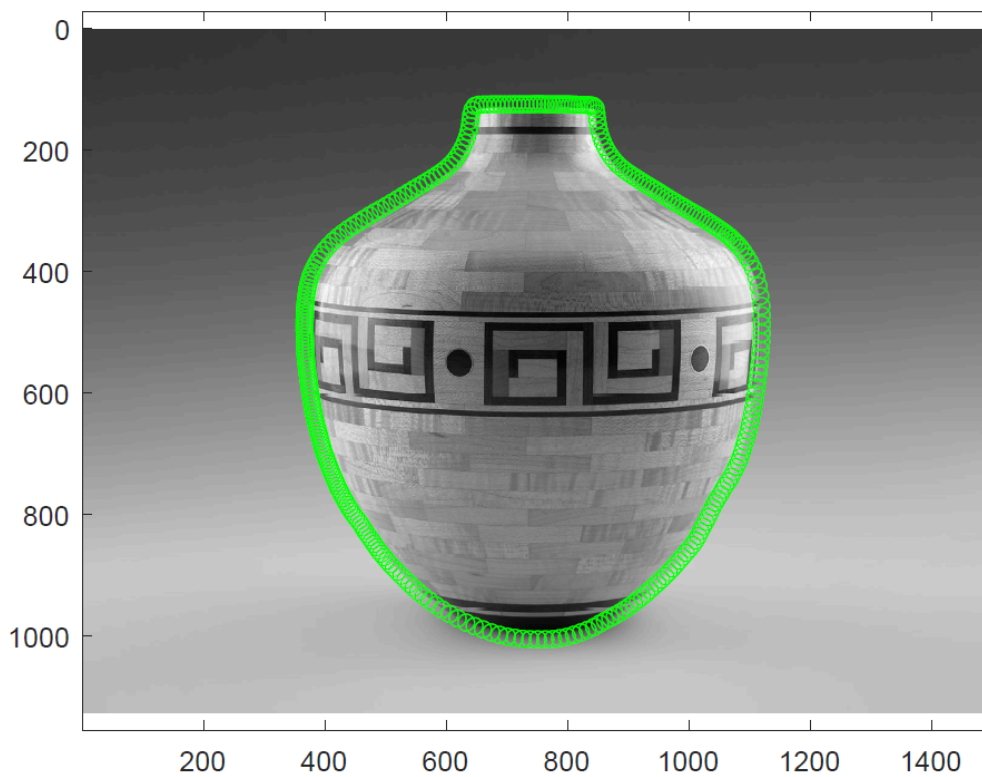


FIGURE 8 – Contour actif d'un vase

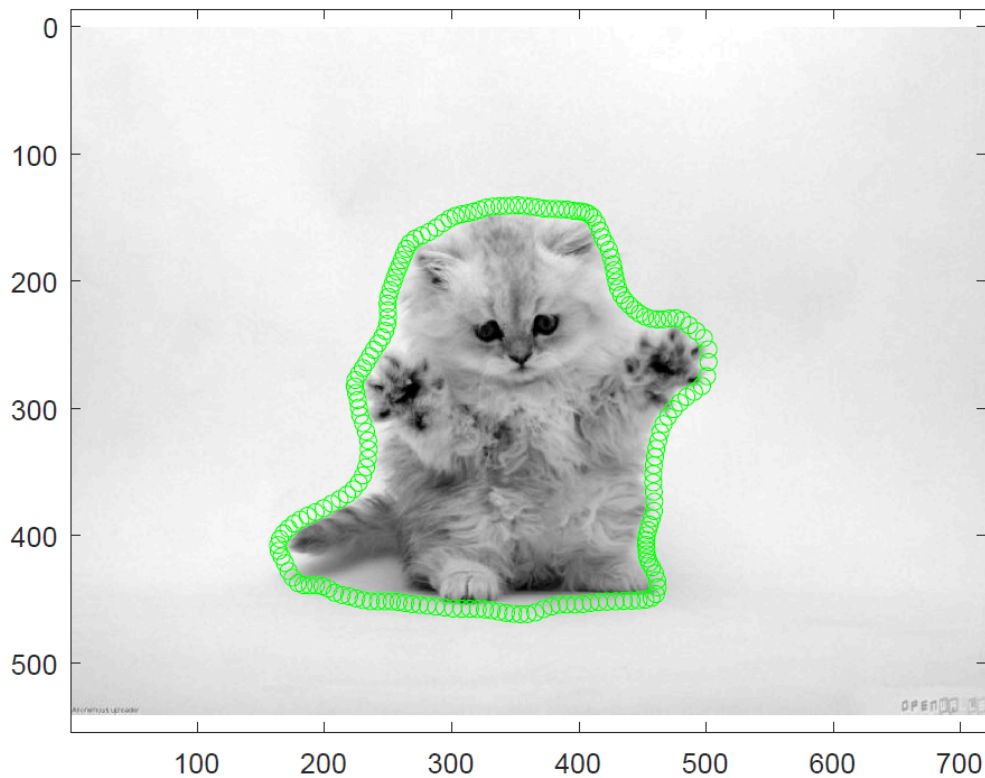


FIGURE 9 – Contour actif d'un chat

Pour la pièce et le vase, on peut ainsi voir que le contour actif épouse parfaitement les formes voulues. En ce qui concerne le chat, le *snake* n'est pas loin de la forme de base mais présente des imperfections. Cela s'explique à la fois par la complexité de la forme, mais aussi par les teintes de gris qui sont proches de l'arrière-plan. Ainsi, l'ombre de la queue est par exemple considérée comme faisant partie du chat. Pour atténuer ce problème, on pourrait penser à augmenter σ , le paramètre du flou gaussien, mais cela peut aussi faire que le contour sera attiré par des zones très contrastées, comme les yeux, au lieu du contour. On remarquera également que, pour la pièce notamment, il y a un petit écart entre le vrai bord de la pièce et le contour actif. Cela est dû est au flou gaussien, qui est absolument nécessaire mais qui a aussi le défaut de rendre moins clair le bord. Il induit donc une légère différence entre le *snake* et la forme voulue.

3.c) Impact et nécessité de chacune des forces

On va regarder sur un exemple très simpliste l'impact des différentes forces pour les comprendre, vérifier qu'elles remplissent bien le rôle voulu et démontrer leur nécessité.

On prend donc pour cela une image binaire initialisée comme ci-dessous.

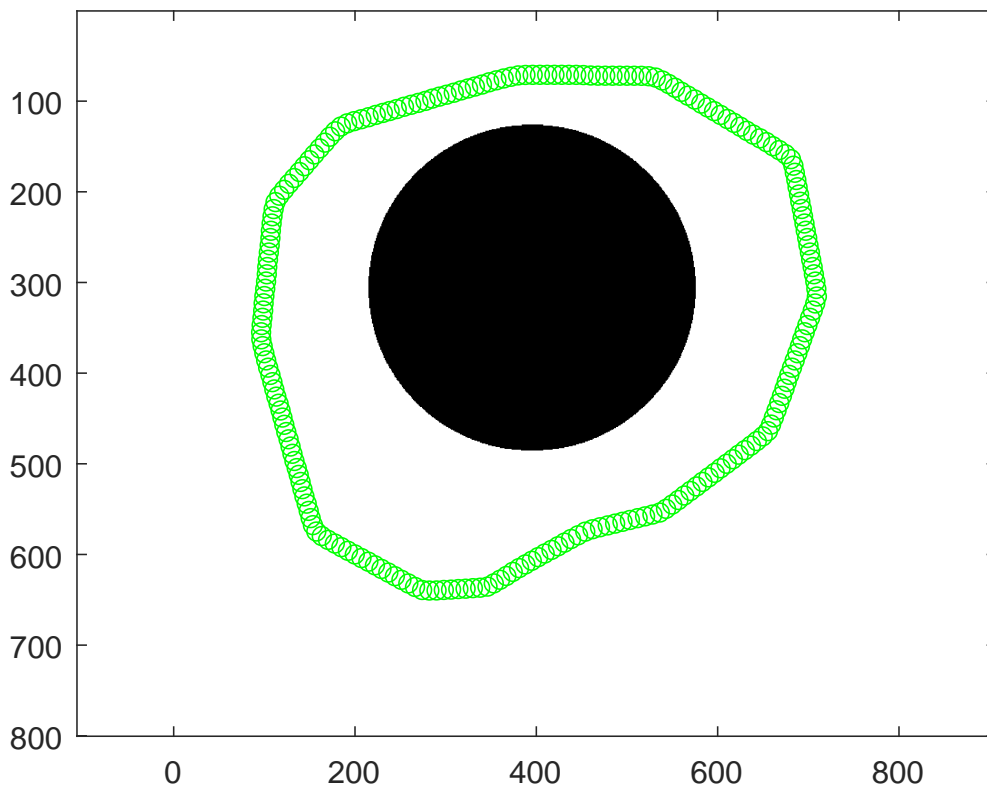


FIGURE 10 – Initialisation du contour actif

On s'intéresse dans un premier temps aux forces de régularité. On s'aperçoit que le cercle noir n'a alors pas d'influence sur le contour. Cependant, par rapport à l'initialisation, on obtient un *snake* qui prend une forme beaucoup plus lisse, comme voulu pour stabiliser le contour. On a ici limité le nombre d'itérations mais, si on laisse l'algorithme tourner pendant longtemps, on obtiendra un cercle, la forme la plus lisse possible.

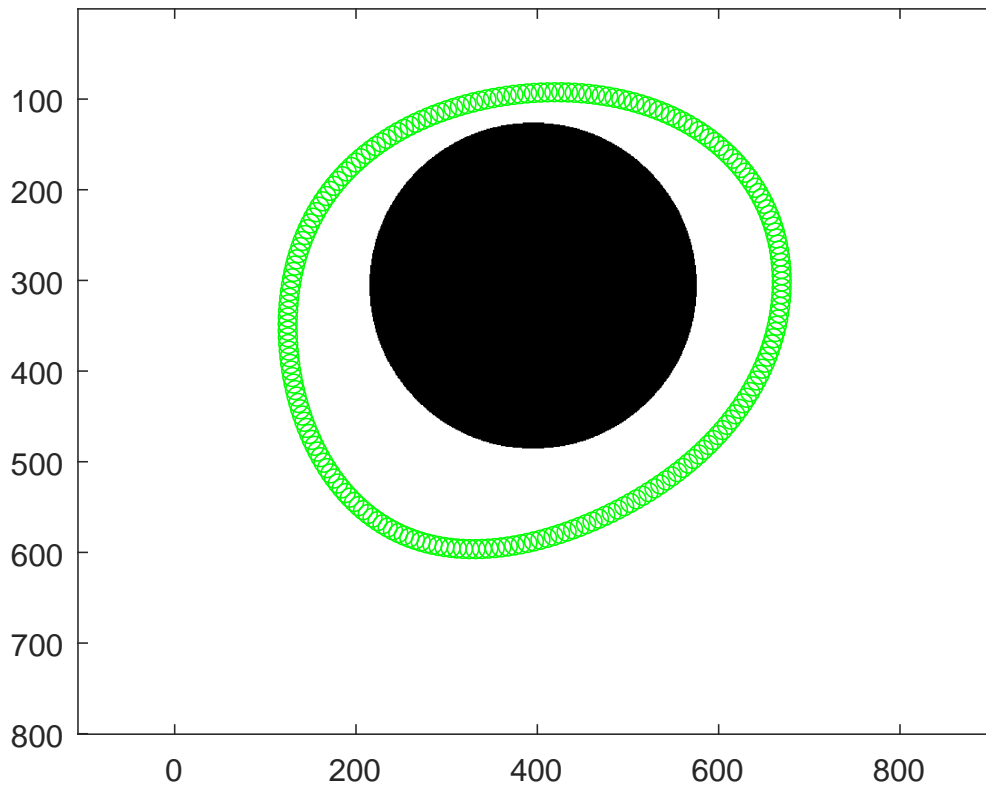


FIGURE 11 – Forces de régularité

On va maintenant rajouter la force *ballon*. Le contour va alors se contracter, sans toutefois s'arrêter sur les contours du cercle. C'est cette force qui permettra au *snake* de se rapprocher de la forme à épouser, notamment sur ses parties concaves. De même, on a ici limité le nombre d'itérations. Sinon, le contour actif se serait écrasé et aurait convergé vers un point.

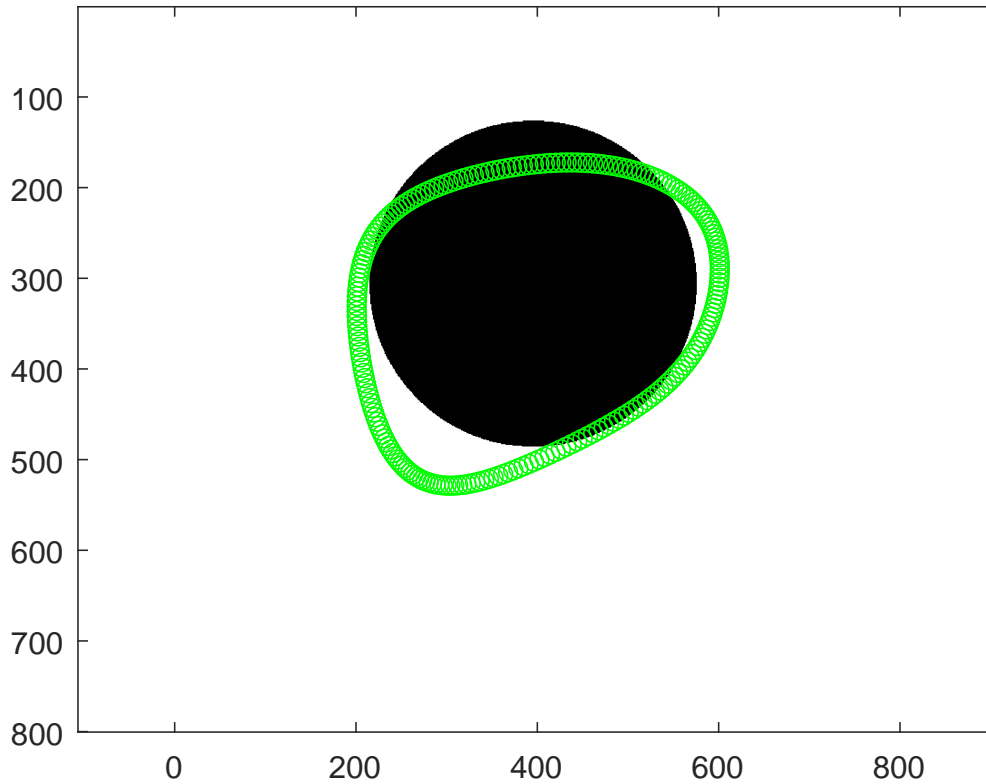


FIGURE 12 – Force *ballon*

Enfin, en rajoutant le potentiel de l'image, on voit que le contour actif va être attiré sur les bords de la forme. Sur cette zone, elle est donc plus forte que la force *ballon* et permet au *snake* de s'arrêter sur les bords.

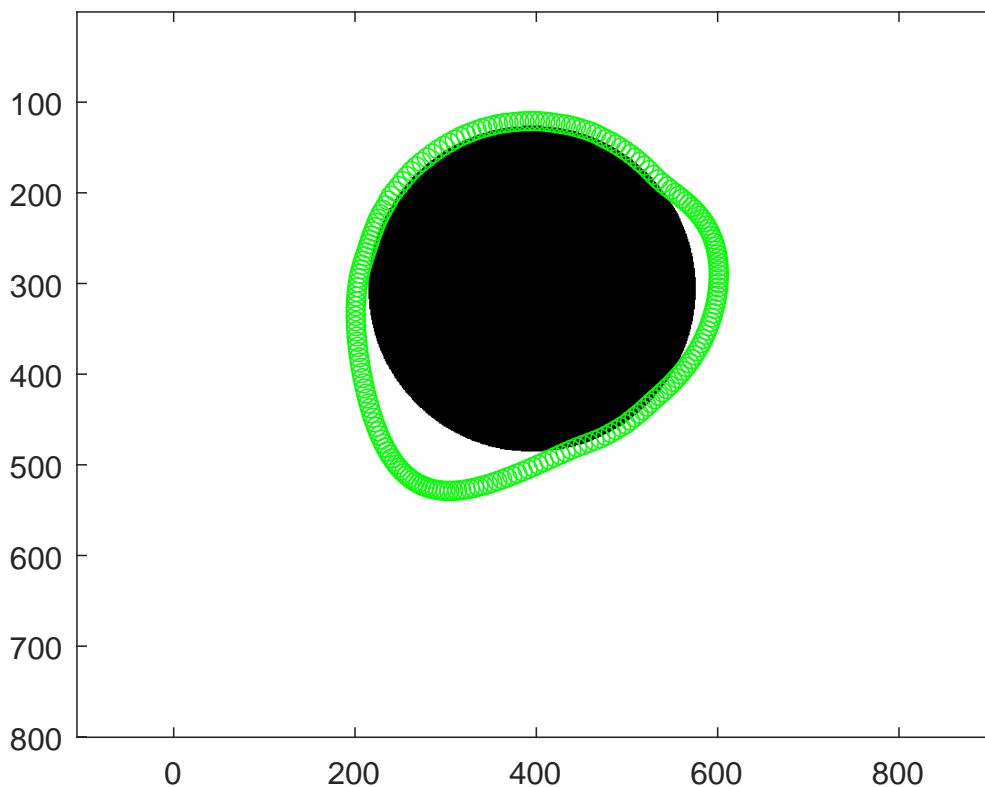


FIGURE 13 – Potentiel de l'image

3.d) Définition de l'erreur

Pour pouvoir analyser les paramètres, il est important de définir la notion d'erreur. En effet, dans le but de confronter les différents résultats obtenus en faisant varier les paramètres, il nous faut un élément de comparaison.

Pour cela, on a choisi de regarder la différence entre la forme à épouser et l'intérieur du contour actif. On va donc calculer la différence entre les aires de ces deux formes. Pour réaliser ce calcul, nous allons utiliser la fonction *poly2mask* de *Matlab* qui va renvoyer une matrice de la taille de l'image initiale. Si un pixel est à l'intérieur du *snake*, la valeur associée dans la matrice sera égale à 1. Sinon, elle vaudra 0. On va ensuite diviser par 255 les valeurs de l'image initiale pour avoir également des valeurs égales à 0 (éléments de l'image de couleur blanche) et 1 (éléments de l'image de couleur noire). Pour simplifier le calcul, on prendra donc une forme noire sur un fond blanc. Pour avoir l'erreur finale, il suffit de calculer la norme entre les deux matrices. Cette erreur nous satisfait car plus le contour actif va être proche de l'image, plus l'erreur sera faible.

Cependant, pour des raisons techniques, cette méthode n'est applicable simplement que sur des images binaires. C'est pourquoi nous nous intéresserons, pour la variation des paramètres α , β et k_B , à des images binaires.

3.e) Paramètres de régularité

Pour tester les différentes valeurs de α et β et voir leur influence sur la convergence du contour actif, on va faire varier ces paramètres sur 2 images binaires, un cercle noir et l'écusson vu précédemment. A une valeur fixée d'un des 2 paramètres, on va faire varier l'autre dans le domaine des valeurs intéressantes, repérées grâce à plusieurs tests effectués en amont. On représentera les courbes sur une échelle semi-logarithmique, plus pertinente ici qu'une échelle classique.

On va commencer par faire varier le paramètre α .

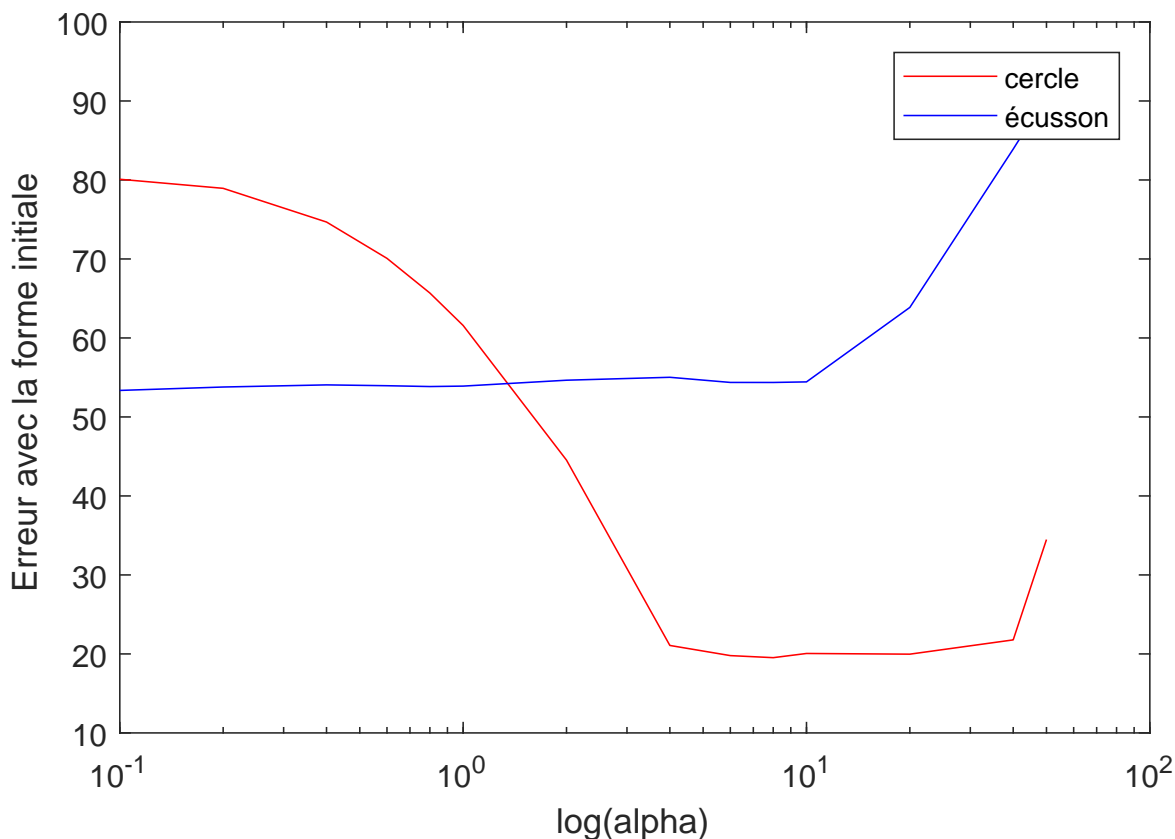


FIGURE 14 – Variation du paramètre α

Le but est de choisir α le plus élevé possible pour assurer la convergence en ayant un contour actif stable, mais tout en minimisant l'erreur avec la forme à épouser. En effet, si α est trop élevé, la force associée va contrebalancer les autres forces et ainsi empêcher la convergence vers la forme voulue. Ainsi, au vu de l'allure des courbes (croissance exponentielle à partir d'un certain α), il est pertinent d'utiliser la méthode du *coude*. Cela nous donnerait ainsi $\alpha = 4$ pour le cercle et $\alpha = 1$ pour l'écusson. Il est de plus cohérent que α soit plus petit pour l'écusson, qui est une forme moins lisse que le cercle.

On s'intéresse maintenant au paramètre β . On normalise les résultats pour pouvoir les comparer.

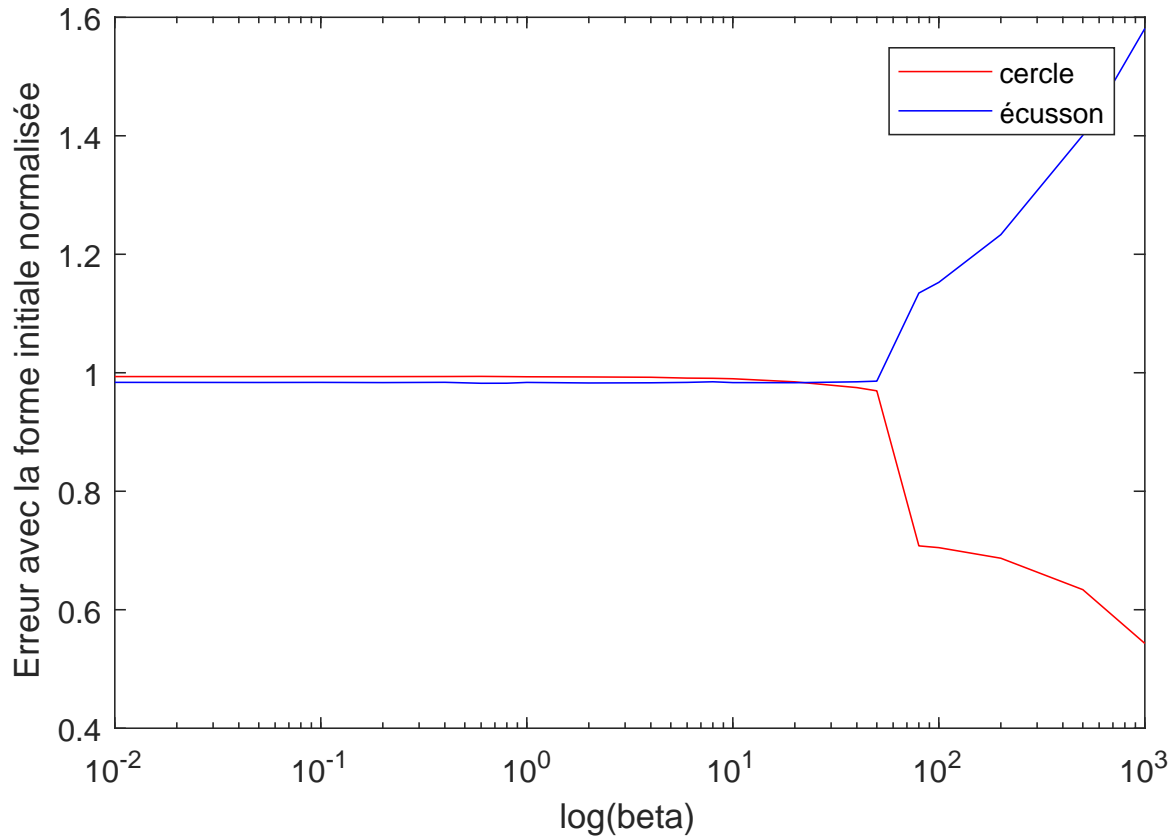


FIGURE 15 – Variation du paramètre β

Pour le paramètre β , les courbes sont relativement constantes sur un domaine très large. Cependant, à partir de $\beta = 50$, l'évolution de β a un plus fort impact. Pour une forme simple et lisse comme le cercle, cela tend à faire baisser l'erreur. Cependant, sur une forme moins régulière, cela fait augmenter l'erreur. On va donc choisir β dans le domaine sur lequel les courbes sont à peu près constantes.

3.f) Force *ballon*

On va effectuer le même test que pour les paramètres de régularité sur k_B . Cela nous donne le graphe suivant :

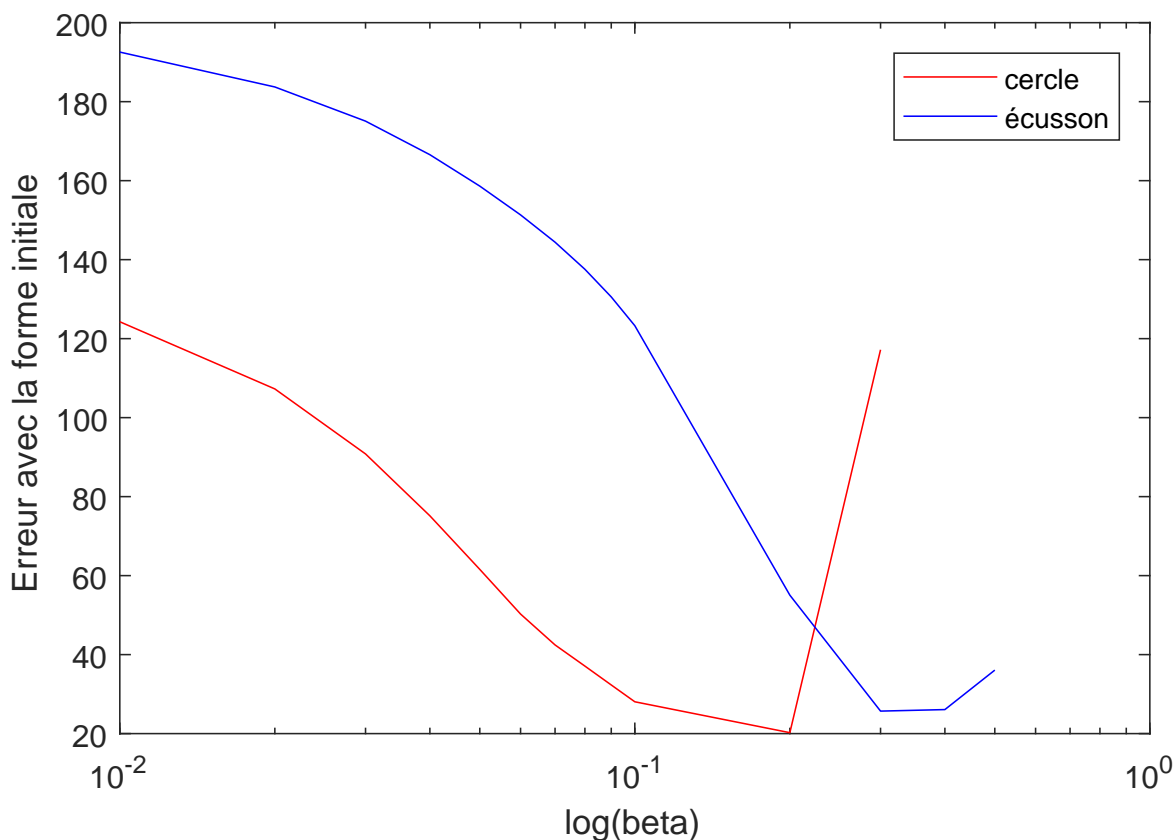


FIGURE 16 – Variation du paramètre k_B

On obtient des résultats similaires à ceux du paramètre α . On doit avoir k_B assez grand pour avoir une force qui fait converger le contour actif assez rapidement, mais pas trop importante pour avoir une convergence vers la forme voulue. On applique donc la même méthode et on trouve $k_B = 0.1$ pour le cercle et $k_B = 0.3$ pour l'écusson. Cependant, le choix de ce paramètre est beaucoup plus fin que celui de α . En effet, si il est trop petit, la force *ballon* sera trop faible pour que le contour actif prenne la bonne force, et, s'il est trop élevé, il n'y a plus de convergence. Le domaine intéressant étant restreint, il faut veiller à choisir le bon paramètre.

3.g) Vitesse de convergence

Le paramètre γ dépend sensiblement de la forme à épouser et est à déterminer avec soin pour ne pas avoir un programme trop lent, mais pour néanmoins assurer la convergence.

Pour évaluer ce paramètre, on va regarder l'erreur moyenne entre V_t et V_{t+1} en fonction de t . Cette erreur moyenne sera calculée en sommant les erreurs point par point entre V_t et V_{t+1} . Cette mesure est intéressante car elle permet de voir la convergence du contour vers sa forme finale. On va donc observer une asymptote horizontale à partir d'un certain t .

En reprenant exemples de la pièce, du vase et du chat, on obtient les graphes ci-dessous avec les différentes valeurs prises par γ dans la légende.

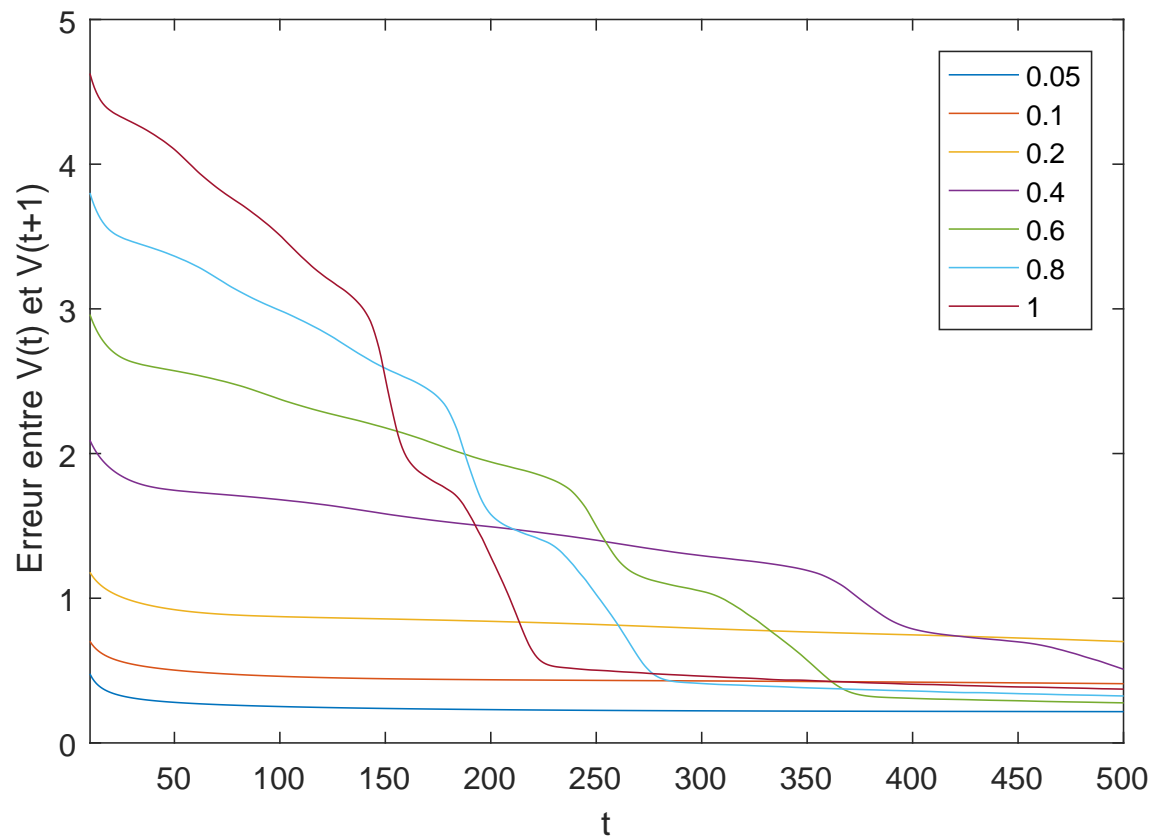


FIGURE 17 – Vitesse de convergence pour la pièce

L'exemple de la pièce est le plus parlant. On constate dans un premier temps que plus γ est élevé, plus la pente de la courbe est importante. C'est un résultat cohérent puisque la vitesse de convergence augmente quand γ augmente. De plus, à partir d'un certain t , on observe bien l'asymptote. Cela correspond au moment à partir duquel le contour actif est très proche de la forme à épouser, et à partir duquel il ne va presque plus évoluer. De plus, pour cet exemple de forme simple, on remarque que le résultat est sensiblement le même (à partir d'un certain t) quel que soit γ .

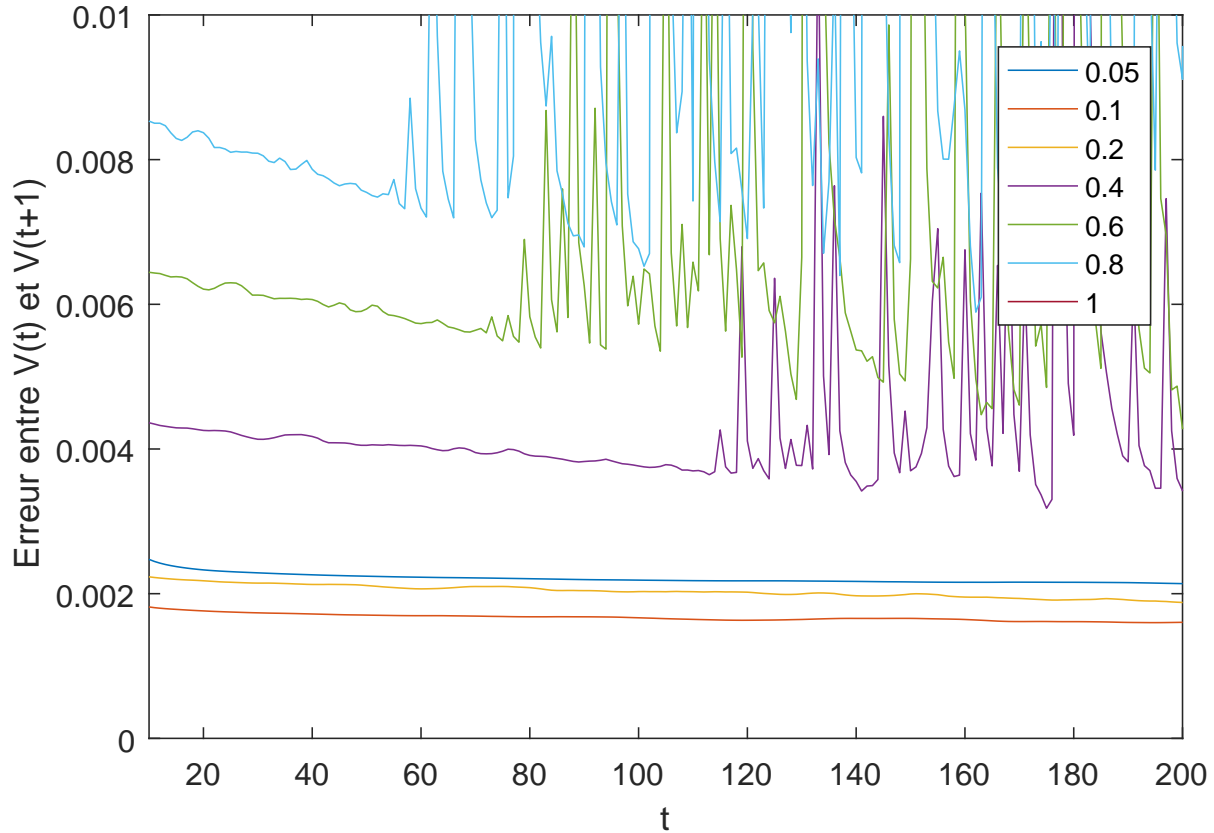


FIGURE 18 – Vitesse de convergence pour le vase

Pour l'exemple du vase, on peut faire le même constat en ce qui concerne la pente. Néanmoins, ici, le choix du γ est très important. En effet, pour $\gamma \leq 0.2$, on a une convergence lente mais stable. Pour un γ plus élevé, cependant, la convergence n'est pas stable à partir d'un certain t . Cela est sûrement dû aux coins du vase (au niveau de l'ouverture), qui font bouger le contour actif une fois qu'il est proche de la forme à épouser. Pour $\gamma = 1$, le contour actif explose dès le départ et ne converge absolument pas vers le vase.

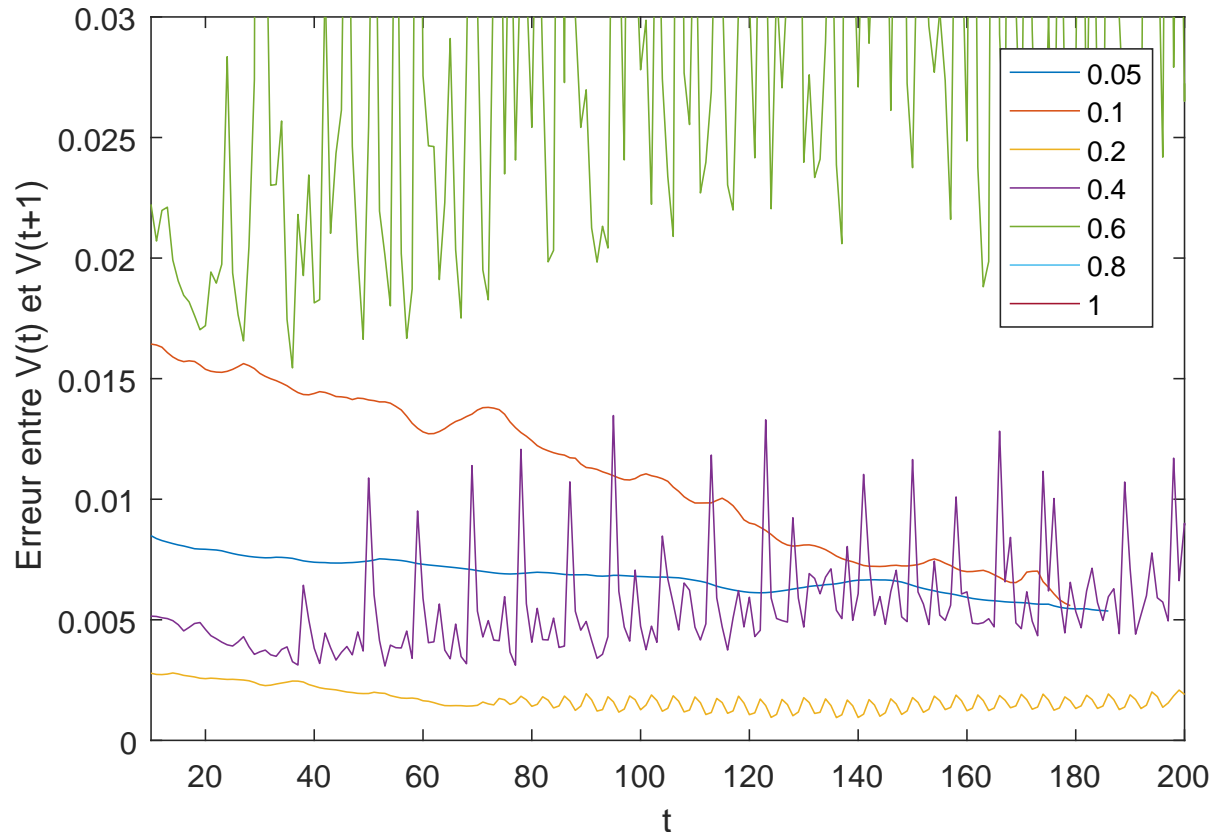


FIGURE 19 – Vitesse de convergence pour le chat

Pour la forme la plus complexe, à savoir le chat, l'effet du γ est encore plus significatif. Pour $\gamma = 0.2$, on remarque déjà que le *snake* n'est pas complètement stable. Pour $\gamma = 0.4$, la stabilité est encore moins assurée. Au-delà, il n'y a plus convergence du contour actif.

Le choix du γ est donc capital pour avoir la convergence du *snake*. Un γ trop petit ralentit la convergence, tandis qu'un γ trop élevé n'assure pas la convergence.

3.h) Flou gaussien

Le paramètre σ lié au flou gaussien est à déterminer en fonction de la forme à épouser. En effet, on a besoin du flou gaussien pour que le contour de la forme à épouser attire le *snake* vers lui, mais il ne faut pas qu'il soit trop élevé pour ne pas être attiré par un élément se trouvant à l'intérieur de la forme. On va prendre l'exemple du chat pour illustrer le choix du σ .

Si on prend $\sigma = 1$, on obtient un résultat proche de celui obtenu précédemment, comme illustré ci-dessous.

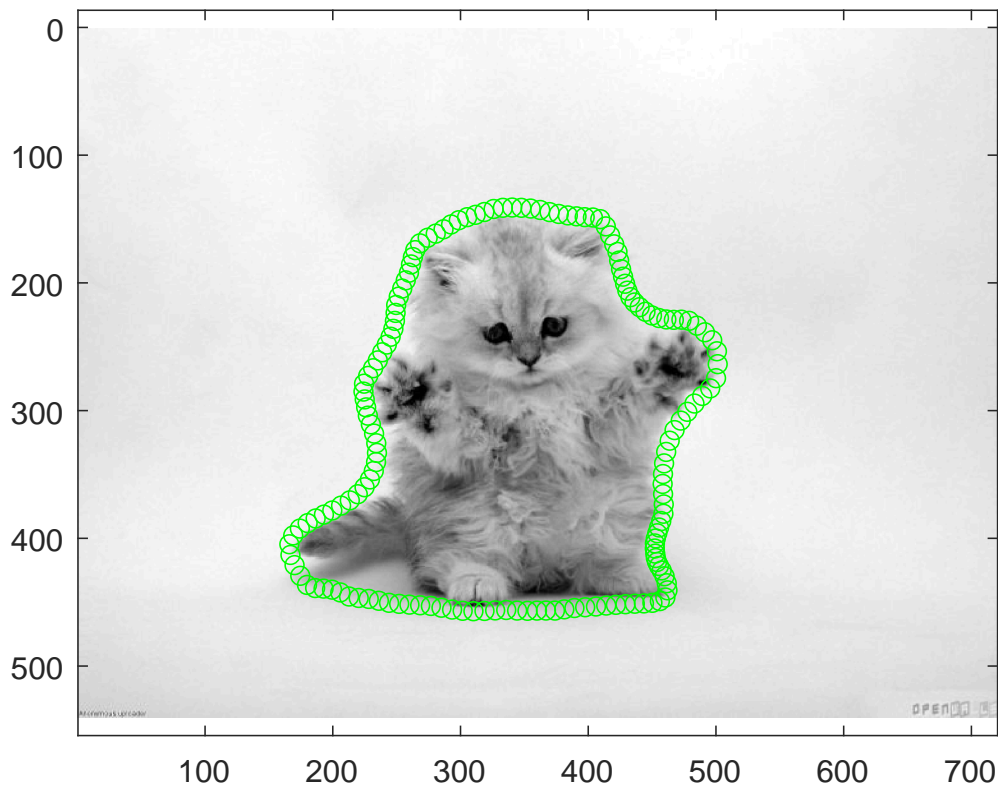


FIGURE 20 – Segmentation d'un chat pour $\sigma = 1$

Cependant, en prenant $\sigma = 3$, on comprend l'influence de ce paramètre. En effet, le contour actif va s'approcher plus précisément de la queue qui, grâce au flou gaussien, va plus attirer le contour que son ombre. Néanmoins, de la même manière, le *snake* va ignorer le bord du chat au niveau de l'oreille car, à cause du flou gaussien, il va être attiré par des zones à fort gradient comme les yeux.

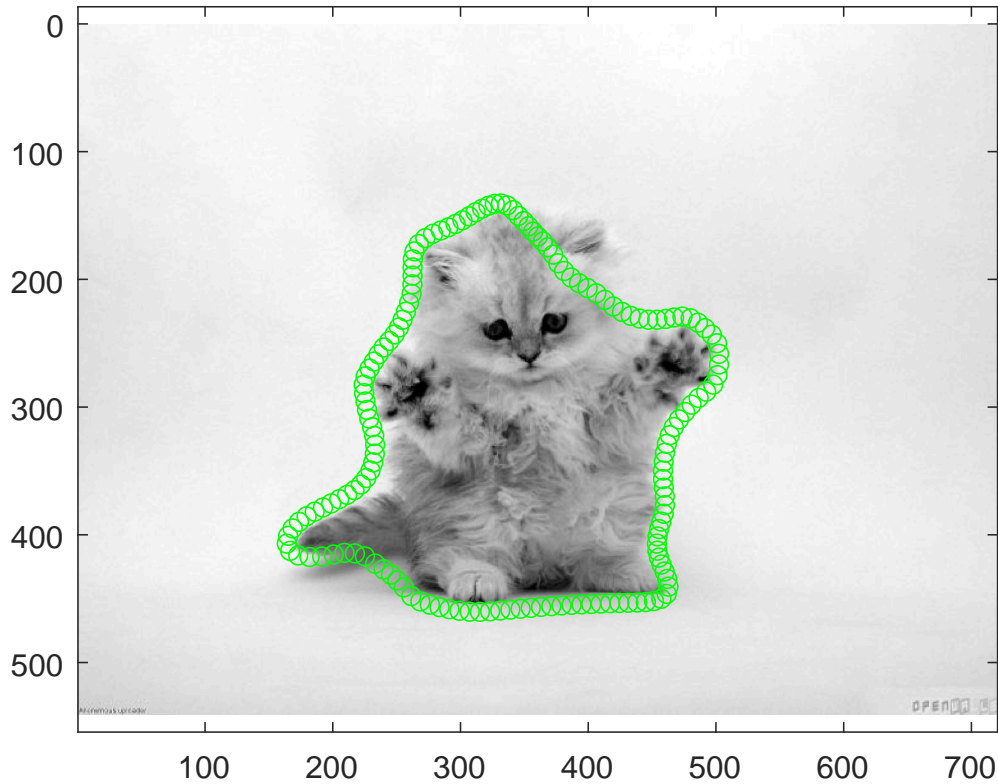


FIGURE 21 – Segmentation d'un chat pour $\sigma = 3$

Le choix de ce paramètre doit donc s'adapter à la forme à épouser.

3.i) Application sur des images médicales

On va reprendre ici l'image médicale vue précédemment. Le but va être de segmenter les poumons, la forme la plus compliquée de cette image.

Pour cela, on va commencer par appliquer un seuillage à l'image. On va ensuite appliquer l'algorithme de segmentation.

Pour le poumon droit, on choisit les paramètres suivant (obtenus de manière empirique) :

- $\lambda = 80$
- $\beta = 1$
- $\alpha = 0.8$
- $\gamma = 0.02$
- $\sigma = 1$
- $k_B = 0.2$

On obtient le résultat ci-dessous.

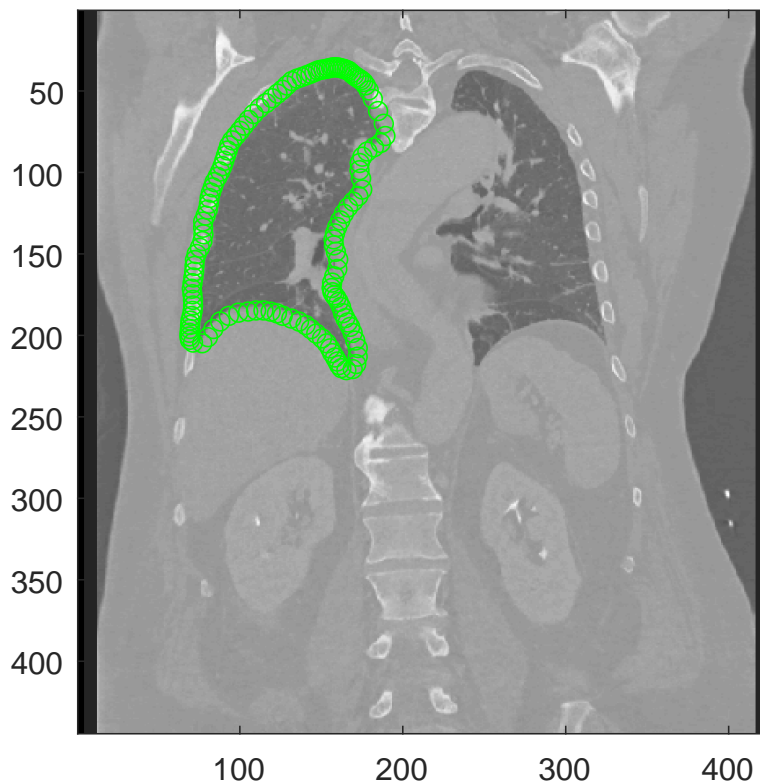


FIGURE 22 – Segmentation du poumon droit

Pour le poumon gauche, on prend les mêmes paramètres avec $\alpha = 1.5$, ce qui nous donne la segmentation suivante :

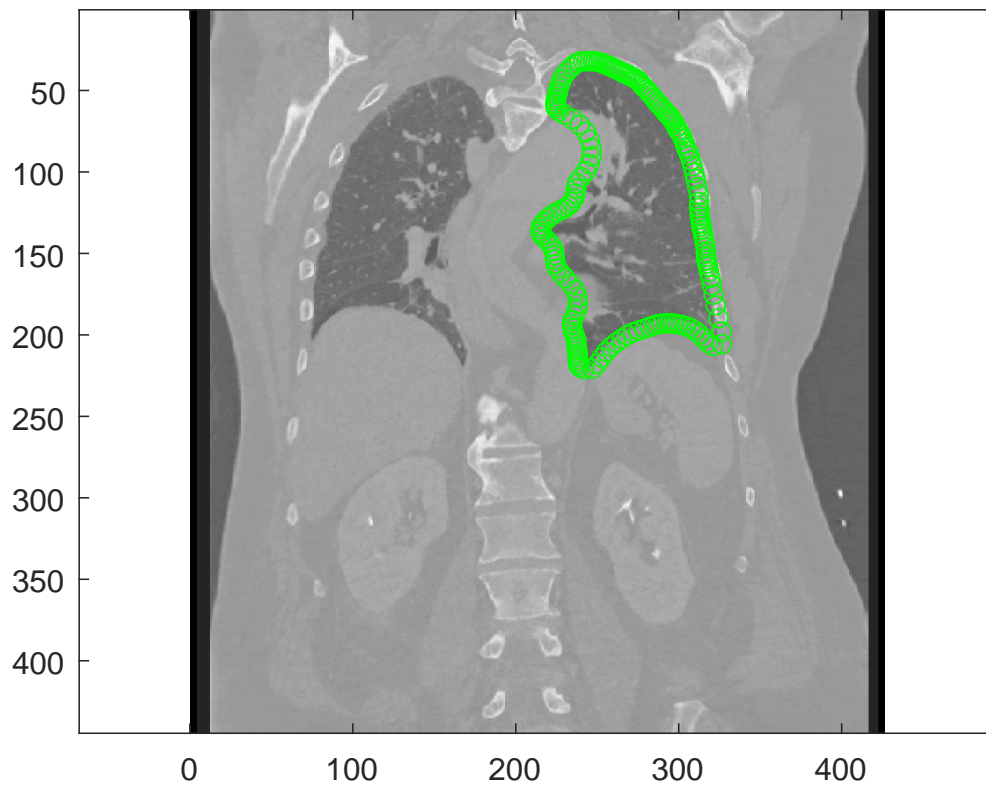


FIGURE 23 – Segmentation du poumon gauche

La segmentation des poumons est satisfaisante. La limite principale du contour actif est au niveau du coeur, le *snake* ayant du mal à épouser parfaitement la cavité à l'intérieur du poumon.

4 Contour actif en 3D

4.a) Analogies avec la résolution en 2D

Le paramétrage du problème en 3D est le même que celui du problème en 2D.

On notera naturellement $v(s) = (x(s), y(s), z(s))$ avec x , y et z les coordonnées des points formant le *snake*.

Le but est de minimiser l'énergie totale $E_s = \int_0^1 E_{int}(s) + E_{im}(s) ds$.

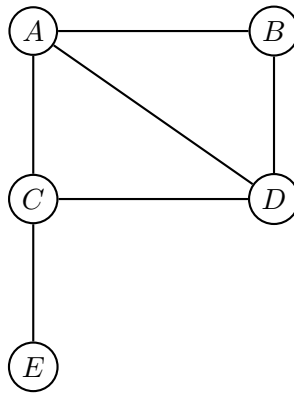
Le problème principal va être d'expliciter la matrice A . En effet, la formule des différences finies est parfaitement adaptée pour un chemin, sur lequel chaque point i a deux voisins situés en $i + 1^{me}$ et $i - 1^{me}$ place. Ici, le maillage 3D sera fait de telle sorte que chaque sommet aura plusieurs voisins, ce qui va modifier la matrice A .

4.b) Définition de la matrice A

On suppose dans cette partie qu'on dispose de la matrice d'adjacence du *snake*, que l'on peut effectivement obtenir numériquement.

Définition 4.1. Pour un maillage à n sommets, la matrice d'adjacence du maillage est une matrice M $n \times n$ dont l'élément M_{ij} est égal à 1 si le sommet i est relié au sommet j , 0 sinon.

Pour mieux illustrer cette définition, on va prendre l'exemple d'un graphe simple :



La matrice d'adjacence liée à ce graphe est donc la matrice

$$A_2 = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

On va adapter les formules des différences finies vues précédemment. Pour cela, on va noter v_i le nombre de voisins du sommet i .

A l'ordre 2, on a ainsi :

$$\begin{cases} a_{ij} = -v_i & \text{si } i = j \\ a_{ij} = 1 & \text{si } i \text{ voisin de } j \\ a_{ij} = 0 & \text{sinon} \end{cases}$$

A l'ordre 4, la formule est plus compliquée. En effet, les voisins des voisins d'un point rentrent également en compte. Pour expliquer de façon plus pertinente et plus adaptée à la résolution que l'on veut en faire, on va expliquer en pseudo-code la création de la matrice A_4 .

Le but est de donner à l'élément diagonal son nombre de voisins, à ses voisins la valeur -4 et aux voisins de ses voisins la valeur 1.

Algorithme 1 Matrice A_4

```

A ←  $\begin{pmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{pmatrix}$ 
pour  $i$  allant de 1 à  $n$  faire
   $A_{ii} \leftarrow 1/3v_i$ 
  pour  $j$  allant de 1 à  $n$  faire
    si  $i$  voisin de  $j$  alors
       $A_{ij} \leftarrow -4$ 
      pour  $l$  allant de 1 à  $n$  faire
        si ( $j$  voisin de  $l$ ) et ( $l \neq i$ ) alors
           $A_{il} \leftarrow 1$ 
        fin si
      fin pour
    fin pour
  fin si
fin pour
fin pour

```

On va reprendre l'exemple précédent pour voir la construction des matrices A_2 et A_4 . On a ainsi :

$$A_2 = \begin{pmatrix} -3 & 1 & 1 & 1 & 0 \\ 1 & -3 & 0 & 1 & 1 \\ 1 & 0 & -2 & 1 & 0 \\ 1 & 1 & 1 & -3 & 0 \\ 0 & 1 & 0 & 0 & -1 \end{pmatrix}$$

$$A_4 = \begin{pmatrix} 9 & -3 & -3 & -2 & 1 \\ -3 & 9 & 2 & -3 & -4 \\ -3 & 2 & 6 & -3 & 0 \\ -2 & -3 & -3 & 9 & 1 \\ 1 & -4 & 0 & 1 & 3 \end{pmatrix}$$

On a donc maintenant $A = -\alpha A_2 + \beta A_4$.

4.c) Equation finale en 3D

En notant $V = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ et en raisonnant de manière similaire que pour le problème en 2D, on se ramène à l'équation

$$V_t = (\gamma A + I_d)^{-1} (V_{t-1} - \gamma f(V_{t-1}))$$

4.d) Explication de f

En utilisant la même formule que précédemment qui fonctionne aussi pour z , on discrétise l'image, on calcule la norme point par point, puis on discrétise à nouveau la matrice obtenue selon les 3 directions.

5 Conclusion

Nous avons donc présenté dans ce rapport les résultats théoriques et numériques pour le contour actif en 2D, ainsi que la généralisation du problème en 3D. Pour des images en deux dimensions, on peut constater que le *snake* épouse parfaitement les formes simples. Cependant, pour des formes plus complexes, on peut remarquer quelques imperfections. Pour effectuer cette segmentation, on s'appuie sur trois forces internes au contour actif, à savoir deux forces de régularité et la force *ballon*, et une force externe qui est le potentiel de l'image.

Le point clé de cette segmentation est le bon choix des paramètres. En effet, ils sont nombreux et dépendent les uns des autres. Il faut donc les choisir avec précision et en prenant en compte les spécificités de la forme à épouser. Il faut ainsi faire varier les paramètres de régularité, la vitesse de convergence, le flou gaussien, l'intensité de la force *ballon* et le paramètre de seuilage. Pour avoir des valeurs pertinentes, il faudra bien considérer toutes les observations réalisées pour déterminer leur influence.

Dans la suite de ce projet, nous nous intéresserons à l'application numérique en 3D. L'algorithme sera facile à implémenter en partant de l'algorithme déjà construit. La principale difficulté sera la modélisation en 3D et l'initialisation du contour actif. En effet, on ne pourra pas le construire de la même manière que dans le problème 2D. Une des solutions pourrait être de l'initialiser sous forme de sphère, sur laquelle on ferait un maillage qui nous donnerait la matrice d'adjacence. On pourra alors segmenter le diaphragme en 3D.

Références

- [1] <http://symbionix.com/>.
- [2] <http://mentice.com/>.
- [3] <http://www.mentice.com/prostatic-artery-embolization>.
- [4] <http://symbionix.com/simulators/gi-mentor/gi-mentor-library-of-modules/gi-mentor-upper-gi-endoscopy/>.
- [5] http://www.it.uu.se/research/scientific_computing/project/rbf/biomech.
- [6] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes : Active contour models. 1987.
- [7] Heather Cody Hazlett Rachel Gimpel Smith Sean Ho James C. Gee Paul A. Yushkevich, Joseph Piven and Guido Gerig. User-guided 3d active contour segmentation of anatomical structures : Significantly improved efficiency and reliability. 2006.
- [8] Arcot Sowmya Banafsheh Pazokifard and Daniel Moses. Automatic 3d modelling of human diaphragm from lung mdct images. 2016.
- [9] Bresson X. and Chan T.F. Fast dual minimization of the vectorial total variation norm and applications to colorimage processing. 2008.
- [10] Laurent D. Cohen. Note on active contour models and balloons. 1991.