

Optimisation de forme de composants véhicules par la méthode "level-set"

Rapport d'étape du PAr 102

Thomas Gaudalet

Encadrants Scientifiques :
Frédéric Gillot - Grégory Vial

Conseiller Projet :
Vincent Fridrici

Résumé

Ce document se veut être une introduction à l'optimisation de forme, et plus particulièrement à l'optimisation topologique au travers de la méthode des Level-Set, développée par les mathématiciens J.A. Sethian et S. Osher. Il a pour ambition d'apporter au lecteur une explication claire du principe de cette méthode ainsi que de proposer une implémentation détaillée de l'algorithme d'optimisation. Enfin, il illustre l'emploi de la méthode dans le cas simple en deux dimensions d'une doublure de capot de voiture. Cet exemple sera traité à l'aide de deux logiciels : Matlab et Salome-Meca, ce dernier au travers de son solveur, le Code Aster.

Table des matières

1	Introduction	4
2	Optimisation de forme	4
3	Méthode Level-Set	5
3.1	Principe	5
3.2	Résolution numérique	6
3.2.1	Position du problème	6
3.2.2	Gestion des chocs	10
3.2.3	Algorithme	12
3.2.4	Fonction distance signée	14
4	Application de la méthode	16
4.1	Définition du problème d'optimisation	17
4.2	Résolution numérique	19
5	Bilan	21
	Références	23

1 Introduction

Le progrès dans l'industrie est intimement lié aux progrès dans différents domaines sous-jacents. En effet, la découverte d'un nouveau matériau ou l'amélioration des caractéristiques d'un matériau par un nouveau traitement, permettent d'améliorer les structures actuelles. De même, un progrès des techniques de production va permettre d'améliorer le produit final tant au niveau de sa fiabilité que de sa finition.

Depuis quelques années, l'industrie s'intéresse à un nouveau moyen d'améliorer les qualités et caractéristiques des produits. Il s'agit de l'optimisation de la forme et de la géométrie des composants d'un produit donné, de sorte qu'il soit le mieux adapté possible aux contraintes imposées par la législation et son environnement de fonctionnement, en un mot son cahier des charges.

De nombreuses méthodes sont étudiées afin d'atteindre cet objectif dont notamment l'optimisation de forme. On citera plus particulièrement l'optimisation topologique dont l'objectif est de déterminer la répartition de matière idéale dans un volume donné soumis à certaines contraintes. Dans ce projet nous nous intéresserons plus en détails à une méthode numérique développée par les mathématiciens J.A. Sethian et S. Osher durant les années 80. Il s'agit de la méthode "level-set", qui est devenu par la suite populaire dans plusieurs domaines tels que le traitement d'image ou l'optimisation. Nous nous intéresserons à l'implémentation numérique de cette méthode et à son application à un objet mécanique simple en deux dimensions.

2 Optimisation de forme

Dans un premier temps, il est nécessaire de poser clairement les bases et les objectifs de tout problème d'optimisation de forme.

Pour tout problème de ce type on se donne un *modèle* qui va permettre d'évaluer le comportement mécanique d'une structure donnée. L'objectif sera de minimiser, ou maximiser, un *critère* fixé (notons qu'il peut y avoir plusieurs *critères*) au sein d'un *ensemble admissible* $O_{admissible}$ de variables d'optimisations qui tiendra compte d'éventuelles contraintes que l'on impose aux variables. Un tel problème disposera donc de trois données : un *modèle*, un *critère* (ou plusieurs) et un *ensemble admissible*.

Mathématiquement, cela revient à déterminer :

$$\min(J(\omega), \omega \in O_{admissible})$$

où J représente les critères et le modèle retenus.

La notion de topologie ayant une importance toute particulière dans certaines méthodes, il est important de la définir avant d'aller plus loin. On considère que deux géométries, deux objets possède la même topologie si il est possible de passer de l'une à l'autre par une déformation continue. Ainsi, en dimension 2, la topologie d'un objet est définie par le nombre de composantes connexes du bord, c'est-à-dire son nombre de trous.

La figure ci-dessous représente deux topologies différentes.

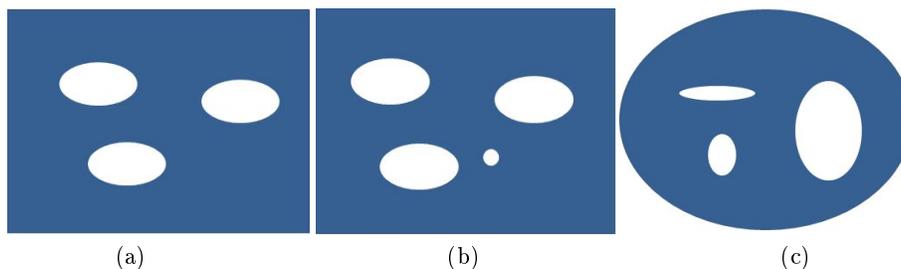


FIGURE 1 – Deux topologies, (b) et (c) ont la même topologie, (a) possède un trou en moins donc est de topologie différente.

On distingue trois types d'optimisation de forme.

- l'optimisation de forme paramétrique, les formes sont alors paramétrées par un nombre réduit de variable, ce qui naturellement réduit la variété de formes possibles.
- l'optimisation de forme géométrique, cela consiste à, partant d'un objet initial, faire évoluer la frontière de la forme initiale tout en conservant sa topologie.
- l'optimisation de forme topologique, on recherche la forme optimale sans aucune restrictions, quitte à changer la topologie.

La méthode Level-set étudiée dans ce projet est catégorisée parmi les méthodes d'optimisation topologique.

3 Méthode Level-Set

L'objectif d'une méthode d'optimisation de forme est de chercher à adapter la forme d'un objet à son environnement et aux différentes contraintes auxquelles il est soumis. Ainsi ces méthodes nécessitent une définition précise des conditions d'emploi et de sollicitation des objets considérés.

Nous détaillerons dans un premier temps le principe de la méthode étudiée au cours de ce projet avant de s'intéresser à la position et à la résolution du problème.

3.1 Principe

Pour simplifier le propos, nous nous plaçons dans un espace de dimension deux.

Soit un objet quelconque donné, on se place dans un ensemble $\Omega \in \mathbb{R}^2$ adapté à l'objet. On se munit de la norme euclidienne du plan, $\| (x, y) \| = \sqrt{x^2 + y^2}$, et on note Γ_t la courbe fermée définie par la surface de l'objet. On définit alors une fonction ϕ de $\Omega \times \mathbb{R}$ dans \mathbb{R} de telle manière qu'elle vérifie $\forall t \in \mathbb{R}$:

$$\begin{cases} \phi(x, t) > 0 & \text{pour } x \text{ à l'extérieur de } \Gamma_t \\ \phi(x, t) = 0 & \text{pour } x \in \Gamma_t \\ \phi(x, t) < 0 & \text{pour } x \text{ à l'intérieur de } \Gamma_t \end{cases}$$

Ceci implique que la surface de l'objet peut à tout moment être définie par :

$$\Gamma_t = \{ x \in \Omega / \phi(x, t) = 0 \}$$

La méthode level-set consiste alors à faire évoluer au cours du temps cette fonction ϕ de manière à respecter des contraintes données. Dans notre cas il s'agira de contraintes physiques pouvant être liées à différents facteurs comme les caractéristiques des matériaux ou les contraintes liées à l'environnement. L'influence de ces contraintes sur la surface considérée est représentée à travers le champ de vitesse F , orienté selon la normale extérieure en tout point appartenant à Γ_t , comme représenté en pointillé sur la figure 2. La seule connaissance de F pour les points de Γ_t n'est cependant pas suffisante, il est nécessaire d'étendre sa définition à tout l'espace pour la résolution numérique. Nous reviendrons sur ce point dans la dernière partie.

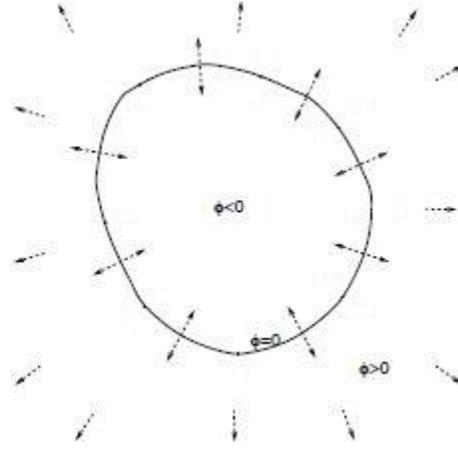


FIGURE 2 – Définition de la fonction ϕ , les flèches représentent le champ F (image issue de [2])

Considérons maintenant un point de la courbe Γ_t dont la position évolue avec le temps $x(t)$ par l'influence de F . A chaque instant t le point considéré appartient à Γ_t et vérifie donc :

$$\phi(x(t), t) = 0 \quad (1)$$

Qui donne par dérivation temporelle puis par application de la règle de dérivation composée :

$$\frac{\partial \phi}{\partial t} + \nabla \phi(x(t), t) \cdot x'(t) = 0 \quad (2)$$

Puisque la vitesse F est définie comme la vitesse selon la normale extérieure n à la courbe, définie $\frac{\nabla \phi}{|\nabla \phi|}$, nous avons : $x'(t) \cdot n = F$.

Ce qui, injecté dans l'équation précédente, fournit l'équation level-set définie par Osher et Sethian :

$$\frac{\partial \phi}{\partial t} + F|\nabla \phi| = 0 \quad (3)$$

L'évolution de la forme de l'objet initialement considéré est entièrement définie par la courbe de niveau zéro de la fonction ϕ solution de cette équation. Le problème se résume donc à la définition de la fonction initiale $\phi(x, t = 0)$, la définition du champ de vitesse F et à la résolution de l'équation précédente. Il est primordial de déterminer F le plus précisément possible car l'évolution de la forme de l'objet sera directement lié aux contraintes représentées par F .

3.2 Résolution numérique

Dans toute cette étude, nous conserverons les notations de la partie précédente et nous choisirons pour exemple, afin d'illustrer le propos, celui de l'évolution d'un feu de forêt. En partant d'un foyer circulaire de rayon 1 à l'origine du repère de l'espace, nous nous intéresserons à l'évolution du feu dans l'espace en considérant plusieurs facteurs d'évolution tels que l'environnement, le vent et son orientation. On introduit, par ailleurs, la condition d'irréversibilité selon laquelle une zone brûlée reste brûlée. C'est-à-dire que la courbe à un instant t englobe celle à l'instant s avec $s < t$.

3.2.1 Position du problème

L'étude numérique consiste en la définition d'un algorithme efficace pour résoudre l'équation level-set précédemment définie.

Il est également nécessaire de définir la fonction initiale $\phi(x, 0)$. Il nous apparaît pertinent de la définir par la fonction de la distance signée à Γ_t .

C'est à dire que nous définissons $\phi(x, 0)$ de la manière suivante :

$$\forall x \in \Omega, \phi(x, 0) = \phi_0(x) = \begin{cases} \inf\{\|x - z\|, z \in \Gamma\} & \text{si } x \text{ est à l'extérieur de } \Gamma_t \\ 0 & \text{si } x \in \Gamma_t \\ -\inf\{\|x - z\|, z \in \Gamma\} & \text{si } x \text{ est à l'intérieur de } \Gamma_t \end{cases}$$

Dans notre exemple la fonction initiale correspondra donc à :

$$\phi(x, y, 0) = x^2 + y^2 - 1 = \phi_0(x, y)$$

Nous nous intéresserons plus loin à un algorithme permettant de définir la fonction de la distance signée à partir d'une fonction ψ ayant pour seule caractéristiques de s'annuler uniquement sur Γ_t et d'être positive à l'extérieur de la courbe fermée Γ_t , respectivement négative à l'intérieur.

La résolution numérique de l'équation passe naturellement par des différences finies. Nous avons en premier lieu choisi un quadrillage classique carré défini entièrement par la donnée du pas d'espace h . On définit par ailleurs le pas de temps par Δt . On notera $\phi_{i,j}^n$ l'approximation de la fonction ϕ au noeud (i, j) après n itérations.

Dans un premier temps, il est pertinent de s'intéresser aux schémas classiques des différences finies, c'est-à-dire les schémas *progressif*, *centré* et *rétrograde*. Ces schémas donnent les approximations de la dérivée partielle spatiale selon x selon les formules suivantes :

$$\begin{aligned} \text{schéma progressif} : \frac{\partial \phi}{\partial x} &\approx \frac{\phi(x+h, y, t) - \phi(x, y, t)}{h} = D^{+x} \phi \\ \text{schéma centré} : \frac{\partial \phi}{\partial x} &\approx \frac{\phi(x+h, y, t) - \phi(x-h, y, t)}{h} = D^{0x} \phi \\ \text{schéma rétrograde} : \frac{\partial \phi}{\partial x} &\approx \frac{\phi(x, y, t) - \phi(x-h, y, t)}{h} = D^{-x} \phi \end{aligned}$$

Il est aisé d'étendre ces formules aux dérivées partielles selon y ou t . A noter que dans un premier temps nous nous limitons au premier ordre, nous considérerons plus loin l'utilisation d'un ordre supérieur dans les différences finies.

En se basant sur ces données, la résolution de l'équation level-set au premier ordre selon le premier des ces schémas donne l'algorithme suivant :

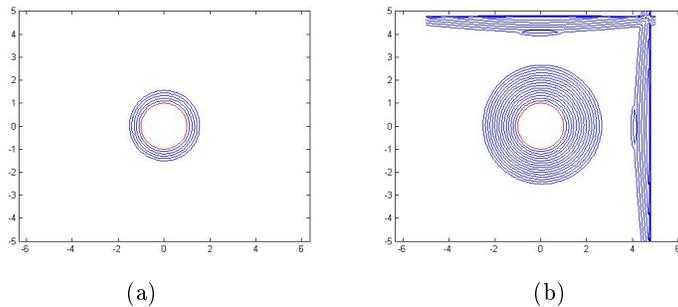
$$\phi_{i,j}^{n+1} = \phi_{i,j}^n - \Delta t F \sqrt{(D^{+x} \phi_{i,j}^n)^2 + (D^{+y} \phi_{i,j}^n)^2} \quad (4)$$

$$\text{où } D^{+x} \phi_{i,j}^n = \frac{\phi_{i+1,j}^n - \phi_{i,j}^n}{h} \text{ et } D^{+y} \phi_{i,j}^n = \frac{\phi_{i,j+1}^n - \phi_{i,j}^n}{h}.$$

L'algorithme est initialisé par la condition $\forall (i, j) \in \mathbb{Z}, \phi_{i,j}^0 = \phi_0(x_i, y_j)$.

On considère dans un premier temps que F est constante égale à 1, ce qui dans notre exemple correspondrait à un espacement et une répartition des arbres dans la forêt homogène.

Au cours des simulations sous Matlab, dont les résultats sont rassemblés dans la figure 3, nous avons fixé $\Omega = [-5, 5] \times [-5, 5]$, $h = 0.1$ et $\Delta t = 0.1$.



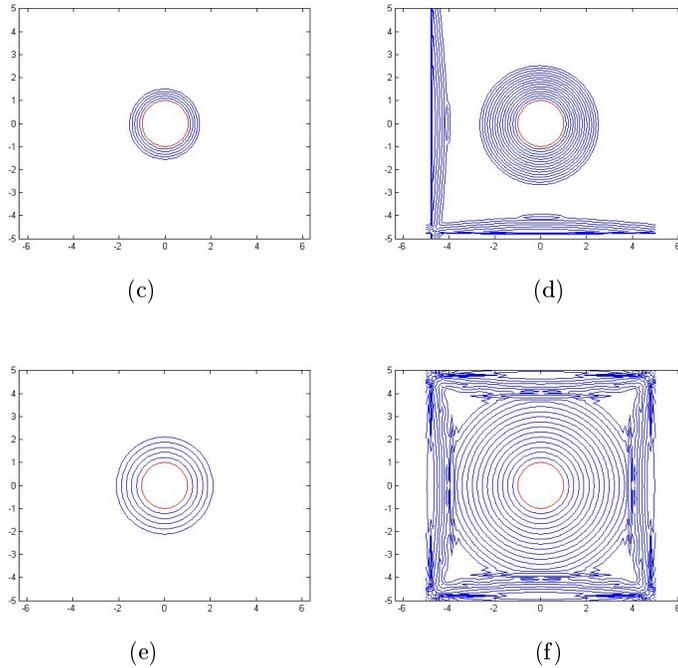


FIGURE 3 – Résultats de simulation sous Matlab. (a),(b) correspondent au schéma progressif, (c),(d) correspondent au schéma rétrograde et (e),(f) au schéma centré. (a), (c) et (e) sont les résultats après 5 itérations (b), (d) et (f) sont ceux obtenus après 15 itérations. Sur chaque graphique le cercle rouge correspond au foyer initial.

Nous observons que conformément à ce que nous pourrions attendre, l'interface, et donc le front des flammes, se propage vers l'extérieur à vitesse constante.

Nous remarquons par ailleurs qu'après un certain nombre d'itérations, des erreurs apparaissent aux bornes du domaine et elles se propagent vers l'intérieur. Il est intéressant de souligner que ces erreurs apparaissent sur les bords droit et haut pour le *schéma progressif* et sur les bords bas et gauche pour le *schéma rétrograde* c'est-à-dire dans le sens du schéma de différences finies. Ce constat nous permet tout d'abord d'écarter le *schéma centré*, celui-ci créant des erreurs sur l'ensemble des bords il n'est pas intéressant de le retenir. En outre, ce constat ne permet pas de départager les deux autres schémas. Nous en déduisons que l'algorithme devra jouer sur les deux schémas selon le sens dans lequel évolue l'interface et selon la localisation du point traité. Ce type de schéma numérique s'appelle un schéma *upwind*.

Pour mettre plus en évidence ce phénomène, nous choisissons de simuler l'effet sur le feu d'un vent soufflant dans une direction fixée et d'intensité constante. Pour cela, nous nous affranchissons de la condition d'irréversibilité énoncée plus haut afin d'analyser les deux schémas. Nous représentons le vent par le vecteur vitesse v , vecteur constant dans tout l'espace. Puisque le vecteur vitesse n'est plus orienté selon la normale extérieure à la courbe l'équation level-set est modifiée de la manière suivante :

$$\frac{\partial \phi}{\partial t} + v \cdot \nabla \phi = 0 \quad (5)$$

car on a alors $x'(t) = v$.

L'algorithme est remplacé par :

$$\phi_{i,j}^{n+1} = \phi_{i,j}^n - \Delta t v \cdot \begin{pmatrix} D^{+x} \phi_{i,j}^n \\ D^{+y} \phi_{i,j}^n \end{pmatrix} \quad (6)$$

Nous fixons dans un premier temps une vitesse v de norme 1 et orienté selon l'axe x , nous comprenons intuitivement que le feu va se décaler vers la droite avec le vent. Dans la simulation

numérique, dont les résultats sont donnés par la figure 4, nous avons conservés le même jeu de paramètre que précédemment.

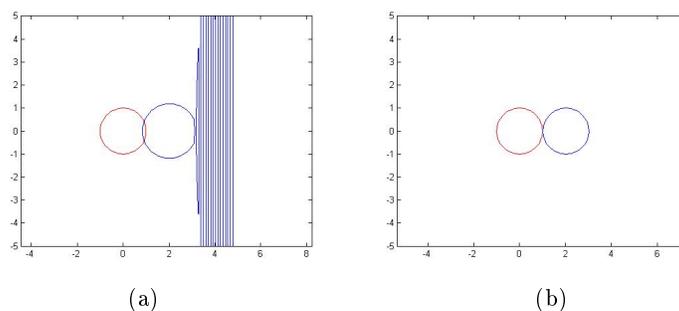


FIGURE 4 – Résultats de simulation sous Matlab après 20 itérations. (a) correspond au schéma progressif et (b) au schéma rétrograde. Les cercles rouge correspondent au foyer initial.

On remarque ainsi que la solution numérique fournie, dans ce cas particulier, par le schéma progressif est incorrecte. Non seulement elle est faussée par les effets de bords se propageant vers l’intérieur, mais on relève aussi une augmentation du rayon du cercle et que celui-ci ne s’est pas déplacé suffisamment loin sur la droite. En effet après 20 itérations, compte-tenu du pas de temps choisi, les cercles rouge et bleu devraient être tangents, ce qui, en revanche, est bien le cas pour la solution numérique fournie par le schéma rétrograde.

Nous comprenons bien par cet exemple l’importance du choix du schéma de différences finies. Il s’agit en chaque point et à chaque instant, donc numériquement à chaque itération, de tester le signe de la vitesse et de choisir, selon la localisation du point considéré, le schéma approprié.

Par ailleurs, différentes simulations, avec différent jeux de paramètres pour F définie comme précédemment, ont permis de faire apparaître l’influence du terme $\sigma = \frac{\Delta t}{h}$. On remarque que si $\sigma \in [0, 1[$, la solution numérique reste qualitativement correcte. Pour mettre ce phénomène en évidence on choisit de fixer $\Omega = [-5, 5] \times [-5, 5]$ et $\Delta t = 0.01$, et de faire varier h . Les résultats de ces simulations sont rassemblées dans la figure 5. Le schéma de différences finies utilisé dans ces simulations est le schéma progressif mais les résultats sont similaires pour le schéma rétrograde.

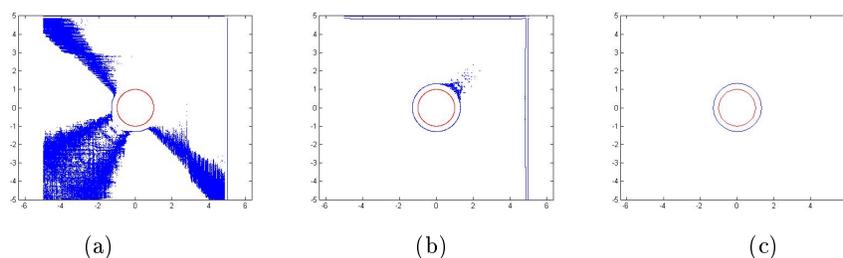


FIGURE 5 – Résultats des simulations après 30 itérations, le cercle rouge correspond au foyer initial. (a) $\sigma = 2$, (b) $\sigma = 1$ et (c) $\sigma = 0.1$

Cette condition se généralise, pour un schéma du premier ordre, par :

$$\Delta t \max_{\Omega} F \leq h$$

Cette condition impose, en fait, que le front ne doit pas se déplacer de plus d’un “carreau” du quadrillage à chaque itération. On appelle ce type de condition une *condition Courant-Friedrichs-Lévy* ou *condition CFL*.

3.2.2 Gestion des chocs

Il est possible, selon la forme de ϕ et de F , lors de la résolution numérique que des *chocs* apparaissent dans la solution.

Définissons la notion de choc.

Pour cela il est utile d'introduire les *caractéristiques* d'une équation aux dérivées partielles.

Soit le système suivant à une dimension suivant :

$$\begin{cases} \frac{\partial \phi}{\partial t}(x, t) + F \frac{\partial \phi}{\partial x}(x, t) = 0, & x \in \mathbb{R}, t > 0 \\ \phi(x, 0) = \phi_0(x), & x \in \mathbb{R} \end{cases} \quad (7)$$

Les caractéristiques de ce système correspondent à l'ensemble des fonctions

$$\begin{aligned} x &: \mathbb{R}_+^* \mapsto \mathbb{R} \\ t &\rightarrow x(t) \end{aligned}$$

qui vérifient $\phi(x(t), t) = \text{constante}$.

Si on choisit F constante, la condition $\frac{\partial \phi}{\partial t}(x(t), t) = 0$, donne le calcul suivant :

$$x'(t) \frac{\partial \phi}{\partial x}(x(t), t) + \frac{\partial \phi}{\partial t}(x(t), t) = 0$$

ce qui, compte-tenu de (7), donne :

$$[x'(t) - F] \frac{\partial \phi}{\partial x}(x(t), t) = 0$$

$\frac{\partial \phi}{\partial x}(x(t), t)$ n'étant pas identiquement nulle, nous obtenons :

$$x'(t) = F$$

Ce qui fournit les caractéristiques suivantes :

$$x(t) = F t + \text{constante}$$

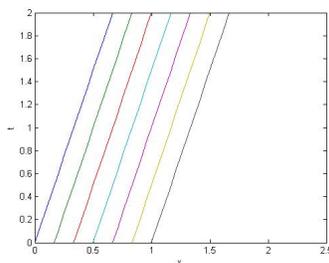


FIGURE 6 – Caractéristiques du système avec $F = \frac{1}{3}$

La solution unique du système est alors donnée par :

$$\phi(x, t) = \phi_0(x - F t)$$

Ceci se traduit par le fait que la donnée initiale ϕ_0 est transportée le long des caractéristiques sans être modifiée.

Considérons maintenant, un système non linéaire de la forme :

$$\begin{cases} \frac{\partial \phi}{\partial t}(x, t) + \frac{\partial f(\phi(x, t))}{\partial x} = 0, & x \in \mathbb{R}, t > 0 \\ \phi(x, 0) = \phi_0(x), & x \in \mathbb{R} \end{cases} \quad (8)$$

avec $f \in C^1(\mathbb{R}, \mathbb{R})$.

En reprenant le calcul effectué plus haut, nous obtenons, compte-tenu de (8) :

$$[x'(t) - f'(\phi(x(t), t))] \frac{\partial \phi}{\partial x}(x(t), t) = 0$$

Comme $\phi(x(t), t) = \text{constante} = \phi_0(x(0))$, cela donne :

$$x'(t) = f'(\phi_0(x(0)))$$

Les caractéristiques sont alors de la forme :

$$x(t) = f'(\phi_0(x(0)))t + x(0)$$

Si on fixe $f(\phi) = \frac{\phi^2}{2}$, et donc $f'(\phi) = \phi$, on obtient l'équation de Burgers :

$$\frac{\partial \phi}{\partial t}(x, t) + \phi(x, t) \frac{\partial \phi}{\partial x}(x, t) = 0, x \in \mathbb{R}, t > 0$$

Nous définissons alors :

$$\phi(x, 0) = \phi_0(x) = \begin{cases} -1 & \text{si } x < 0 \\ 1 & \text{si } x > 0 \end{cases}, x \in \mathbb{R}$$

Et on obtient les caractéristiques représentées ci-dessous pour le système :

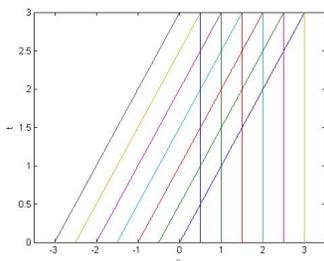


FIGURE 7 – Caractéristiques pour F dépendant de la variable spatiale.

On voit donc qu'un problème apparaît aux points de croisements des caractéristiques. C'est ce qu'on appelle un *choc*. Plusieurs solutions sont possibles à partir de ces points. Comment choisir la plus adaptée au problème ?

C'est ici qu'il est nécessaire d'introduire la notion de *solution faible et de solution entropique*.

En effet, de manière générale, il n'est pas toujours possible de définir une solution "classique" à un système, notamment lorsque $\phi_0 \notin C^1(\mathbb{R})$. On a recours dans ces cas à des *solutions faibles*.

On dit que ϕ est solution faible du système (8) si et seulement si $\phi \in L^\infty(\mathbb{R} \times \mathbb{R}_+)$ et vérifie :

$$\forall \chi \in C_c^1(\mathbb{R} \times \mathbb{R}_+, \mathbb{R}), \int \int_{\mathbb{R} \times \mathbb{R}_+} \left[\phi(x, t) \frac{\partial \chi}{\partial t}(x, t) + f(\phi(x, t)) \frac{\partial \chi}{\partial x}(x, t) \right] dx dt + \int_{\mathbb{R}} \phi_0(x) \chi(x, 0) dx = 0$$

Il est cependant possible que plusieurs solutions faibles conviennent. Reprenons, pour illustrer le propos, l'équation de Burgers :

$$\begin{cases} \frac{\partial \phi}{\partial t}(x, t) + \phi(x, t) \frac{\partial \phi}{\partial x}(x, t) = 0, & x \in \mathbb{R}, t > 0 \\ \phi(x, 0) = \phi_0(x) = \begin{cases} -1 & \text{si } x < 0 \\ 1 & \text{si } x > 0 \end{cases}, & x \in \mathbb{R} \end{cases}, \quad (9)$$

Alors les fonctions ϕ_1 et ϕ_2 définies par :

$$\phi_1(x, t) = \begin{cases} -1 & \text{si } x < 0 \\ 1 & \text{si } x > 0 \end{cases} \quad \text{et } \phi_2(x, t) = \begin{cases} -1 & \text{si } x < -2t \\ \frac{x}{2t} & \text{si } -2t < x < 2t \\ 1 & \text{si } x > 2t \end{cases}$$

sont toutes deux des solutions faibles de l'équation de Burgers. Comment choisir la solution qui convient ? C'est ici qu'interviennent les solutions entropiques.

Pour le système (8), on dit que $\phi \in L^\infty(\mathbb{R} \times \mathbb{R}_+)$ est *solution entropique* si pour toute fonction convexe $\eta \in C^1(\mathbb{R})$, appelée *entropie*, et pour toute fonction $\psi \in C^1$ telle que $\psi' = f'\eta'$, appelée *flux d'entropie*, on a $\forall \chi \in C_c^1(\mathbb{R} \times \mathbb{R}_+, \mathbb{R})$,

$$\int \int_{\mathbb{R} \times \mathbb{R}_+} \left[\eta(\phi(x, t)) \frac{\partial \chi}{\partial t}(x, t) + \psi(\phi(x, t)) \frac{\partial \chi}{\partial x}(x, t) \right] dx dt + \int_{\mathbb{R}} \eta(\phi_0(x)) \chi(x, 0) dx \geq 0$$

Une méthode pour déterminer la solution faible vérifiant la condition entropique du système (8), consiste à résoudre le problème de diffusion associé suivant :

$$\frac{\partial \phi}{\partial t}(x, t) + \frac{\partial f(\phi(x, t))}{\partial x} - \epsilon \frac{\partial^2 \phi(x, t)}{\partial x^2} = 0 \quad (10)$$

Soit ϕ_ϵ la solution de ce nouveau problème, on peut montrer alors que lorsque l'on fait tendre ϵ vers 0, ϕ_ϵ converge vers la solution entropique du système (8).

Abordons maintenant la question de l'utilisation d'une différence finie d'ordre supérieur. Les termes d'ordre supérieur à 1 induisent un lissage de la solution. Cette propriété présente un intérêt tout particulier lorsqu'il s'agit de supprimer les "petites" discontinuités qui peuvent apparaître en certains points, au cours de la résolution, alors que la solution serait censée rester continue. En revanche ils peuvent aussi entraîner une perte d'information au niveau de la formation de singularités. En effet, le lissage induit par ces termes peut supprimer l'apparition de fortes discontinuités, de chocs ou de coins notamment, alors qu'il serait primordial de les garder.

Il faut donc prendre des précautions lors de l'utilisation de schémas d'ordre supérieur afin de prévenir des pertes d'informations trop importantes.

3.2.3 Algorithme

L'étude des parties précédentes, nous permet d'aboutir aux constats suivants :

1. Il est nécessaire d'adapter le schéma aux différences finies en chaque point, il faut donc opter pour un schéma upwind.
2. Il est important de déterminer et de respecter la condition CFL entre les pas de temps et d'espace dans le schéma fixé.
3. Si l'on choisit d'utiliser des différences finies d'ordre supérieur, afin de lisser le résultat, il faut prendre garde que cela n'entraîne pas une perte d'information au niveau des singularités qui peuvent apparaître au cours de la résolution (chocs, discontinuités, coins).

Ceci nous conduit à proposer l'algorithme "basique" donné par J.A. Sethian dans [1]. L'algorithme qu'il expose est d'ordre 2 et suit le principe du schéma upwind. Pour palier au point 3, il introduit une fonction *switch* permettant de détecter la présence d'une singularité et de supprimer, le cas échéant, la composante d'ordre 2. Par ailleurs, la condition CFL liée à l'algorithme est $\frac{\Delta t}{h} \in [0, \frac{1}{2}[$ puisque nous utilisons une différence finie d'ordre 2. Le champ F étendu à tout l'espace est représenté par une matrice. Cette représentation est légitime car les vitesses, en chaque point, sont orientées selon

la normale à la courbe de niveau passant par ce point, il nous suffit donc seulement de rassembler dans la matrice les normes de la vitesse en chaque point. Cela fournit l'algorithme suivant :

$$\phi_{i,j}^{n+1} = \phi_{i,j}^n - \Delta t (max(F_{i,j}, 0)\nabla^+ + min(F_{i,j}, 0)\nabla^-) \quad (11)$$

où

$$\nabla^+ = \sqrt{max(A, 0)^2 + min(B, 0)^2 + max(C, 0)^2 + min(D, 0)^2}$$

$$\nabla^- = \sqrt{max(B, 0)^2 + min(A, 0)^2 + max(D, 0)^2 + min(C, 0)^2}$$

et

$$A = D^{-x}\phi_{i,j}^n + h m(D^{-x-x}\phi_{i,j}^n, D^{+x-x}\phi_{i,j}^n)$$

$$B = D^{+x}\phi_{i,j}^n - h m(D^{+x+x}\phi_{i,j}^n, D^{+x-x}\phi_{i,j}^n)$$

$$C = D^{-y}\phi_{i,j}^n + h m(D^{-y-y}\phi_{i,j}^n, D^{+y-y}\phi_{i,j}^n)$$

$$D = D^{+y}\phi_{i,j}^n - h m(D^{+y+y}\phi_{i,j}^n, D^{+y-y}\phi_{i,j}^n)$$

La fonction m correspond à la fonction *switch* discutée plus haut, elle est définie par :

$$m(x, y) = \begin{cases} x & \text{ssi } xy \geq 0 \text{ et } |x| \leq |y| \\ y & \text{ssi } xy \geq 0 \text{ et } |x| > |y| \\ 0 & \text{ssi } xy < 0 \end{cases}$$

Pour finir, on définit :

$$D^{+x+x}\phi(x, t) = \frac{\phi(x + 2h, t) - 2\phi(x + h, t) + \phi(x, t)}{h^2}$$

$$D^{+x-x}\phi(x, t) = \frac{\phi(x + h, t) - 2\phi(x, t) + \phi(x - h, t)}{h^2}$$

$$D^{-x-x}\phi(x, t) = \frac{\phi(x, t) - 2\phi(x - h, t) + \phi(x - 2h, t)}{h^2}$$

L'application de cet algorithme dans le cas vu précédemment, avec F constant égale à 1 dans tout l'espace, fournit les résultats rassemblés dans la figure 7, avec pour jeu de paramètre $\Omega = [-5, 5] \times [-5, 5]$ et $\Delta t = 0.01$. Nous avons choisi quatre valeurs différentes de h pour mettre en évidence la condition CFL.

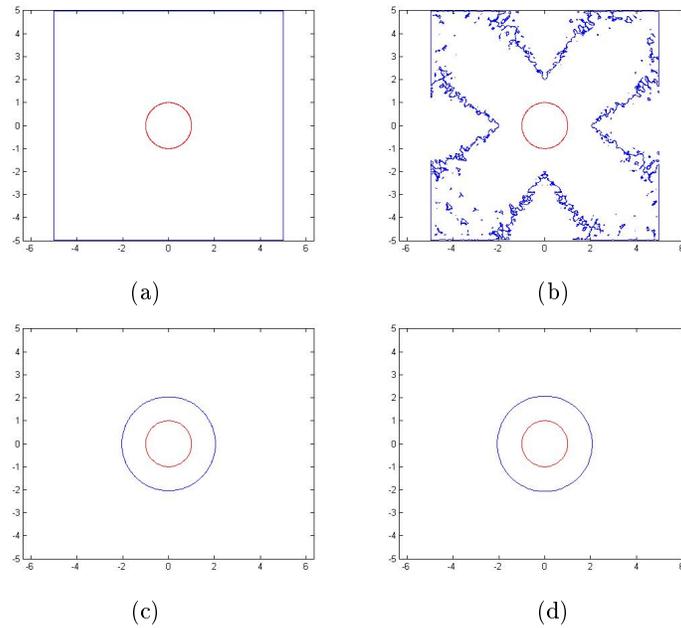


FIGURE 8 – Résultats de la résolution numérique après 50 itérations, les cercles rouges correspondent au foyer initial. (a) $h = 0.01$, (b) $h = 0.02$, (c) $h = 0.05$ et (d) $h = 0.1$.

Ces simulations permettent de valider la condition CFL définie.

Des simulations avec un grand nombre d'itérations mettent en évidence la robustesse de ce schéma, les anomalies aux bords qui apparaissaient au cours de l'application des différences finies fixes sont supprimées comme le montre la figure 9. Ces simulations ont été réalisées avec pour jeu de paramètre $\Omega = [-5, 5] \times [-5, 5]$, $h = 0.1$ et $\Delta t = 0.01$ ($\sigma = 0.1$).

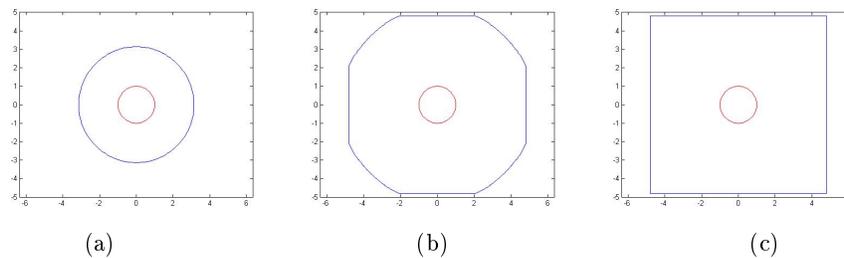


FIGURE 9 – Résultats de simulation, (a) après 100 itérations, (b) après 200 itérations et (c) après 500 itérations.

On remarque ainsi que même lorsque l'on sort du domaine Ω , aucune anomalie aux bords n'apparaît.

3.2.4 Fonction distance signée

Lorsque les transformations de la fonction ϕ sont plus conséquentes, il se peut qu'elle perde, au fur et à mesure des itérations, certaines propriétés et qu'elle se déforme. Ce qui peut entraîner l'apparition de gradients très importants et donc fausser les résultats.

Il peut, dans ces cas là, s'avérer pertinent de remplacer la fonction ϕ , au cours de l'algorithme, par une fonction ayant un meilleur comportement. Par exemple, et c'est la solution que nous choisirons,

on peut la remplacer par la fonction distance signée à la surface considérée à un certain stade de l'algorithme.

Une approche pour déterminer cette fonction consiste à résoudre l'équation :

$$\frac{\partial \phi}{\partial t} = \text{sgn}(\phi^0)(1 - |\nabla \phi|) \quad (12)$$

où ϕ^0 est définie par la fonction ϕ à remplacer et la fonction sgn est la fonction signe définie par :

$$\text{sgn}(x) = \begin{cases} +1 & \text{si } x > 0 \\ 0 & \text{si } x = 0 \\ -1 & \text{si } x < 0 \end{cases}$$

A noter que l'on considère que la fonction ϕ a conservé, ce qui sera toujours le cas compte-tenu du schéma upwind choisis, la propriété d'être positive à l'extérieur, négative respectivement à l'intérieur, du contour fermé correspondant à la ligne de niveau zéro de ϕ .

La résolution jusqu'au temps T de cette équation, donc détermination de la fonction $\phi(x, T)$, donne alors la fonction distance signée pour tout les points situés à une distance inférieure ou égale à T .

G.Russo et P. Smereka proposent l'implémentation numérique suivante, à noter que nous nous basons à nouveau sur un schéma upwind :

$$\phi_{i,j}^{n+1} = \begin{cases} \phi_{i,j}^n - \frac{\Delta t}{h} (\text{sgn}(\phi_{i,j}^0) |\phi_{i,j}^n| - D_{i,j}) & \text{si } (i, j) \in \Sigma \\ \phi_{i,j}^n - \Delta t (\text{sgn}(\phi_{i,j}^0) G(\phi)_{i,j}) & \text{sinon} \end{cases}$$

où Σ est l'ensemble des noeuds du quadrillage situé à moins d'un pas d'espace de la ligne de niveau zéro.

Ainsi $\Sigma = \{(i, j) / \phi_{i,j}^0 \phi_{i-1,j}^0 < 0 \text{ ou } \phi_{i,j}^0 \phi_{i+1,j}^0 < 0 \text{ ou } \phi_{i,j}^0 \phi_{i,j-1}^0 < 0 \text{ ou } \phi_{i,j}^0 \phi_{i,j+1}^0 < 0\}$.

On définit par ailleurs :

$$D_{i,j} = \frac{2\Delta x \phi_{i,j}^0}{[(\phi_{i+1,j}^0 - \phi_{i-1,j}^0)^2 + (\phi_{i,j+1}^0 - \phi_{i,j-1}^0)^2]^{\frac{1}{2}}}$$

$$G(\phi)_{i,j} = \begin{cases} \sqrt{\max(a_+^2, b_-^2) + \max(c_+^2, d_-^2)} - 1 & \text{si } \phi_{i,j}^0 > 0 \\ \sqrt{\min(a_-^2, b_+^2) + \min(c_-^2, d_+^2)} - 1 & \text{si } \phi_{i,j}^0 < 0 \end{cases}$$

Où on pose les quantités $x_+ = \max(x, 0)$ et $x_- = \min(x, 0)$ pour tout réel x , et :

$$\begin{cases} a = D^{-x} \phi_{i,j} \\ b = D^{+x} \phi_{i,j} \\ c = D^{-y} \phi_{i,j} \\ d = D^{+y} \phi_{i,j} \end{cases}$$

Afin de mettre en évidence l'efficacité de ce schéma, on choisit de calculer la fonction distance signée à un carré de côté 1. On fixe $\Omega = [-5, 5] \times [-5, 5]$, $\Delta t = 0.01$ et $h = 0.1$ (la condition CFL est donc bien vérifiée puisque $\sigma = 1$). On initialise ϕ^0 nulle sur le carré, positive à l'extérieur et négative à l'intérieur. Les valeurs de ϕ à l'intérieur et à l'extérieur sont tirées aléatoirement de manière à souligner l'efficacité de l'algorithme. Sur les courbes de la figure 10, nous avons choisis de tracer les lignes de niveaux correspondant aux valeurs $\{-0.5, 0, 0.5, 1, 1.5, 2, 3, 4, 5\}$.

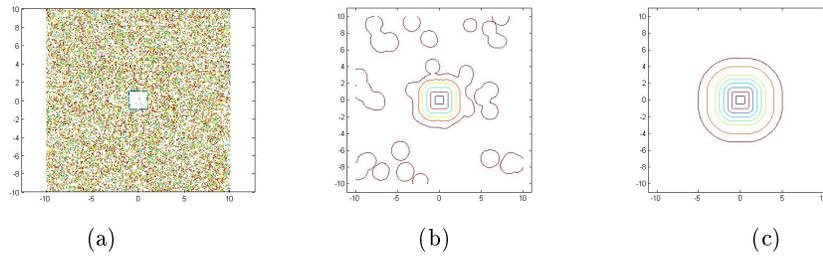


FIGURE 10 – Résultats de la simulation, (a) lignes de niveau de ϕ^0 , (b) lignes de niveau après 200 itérations et (c) lignes de niveau après 500 itérations

Les résultats sont donc très probants, notons cependant que G. Russo et P. Smereka mettent en évidence une erreur relative par rapport à la fonction distance signée d'environ 0.2%. Cette erreur est intrinsèque à l'algorithme et ne dépend pas du nombre d'itérations. Nous vous renvoyons à la lecture de [2] pour plus de précisions.

4 Application de la méthode

Dans cette partie on s'attache à appliquer la méthode développée dans la partie précédente à un cas pratique en deux dimensions. On se propose d'étudier l'optimisation d'une doublure de capot de voiture. La forme du capot en lui-même correspond plus à un choix esthétique et est donc imposé par les designers du modèle de voiture. Pour lui conférer les propriétés mécaniques qui nous intéressent (notamment sa rigidité) il est nécessaire de le doubler par une pièce sur laquelle on peut agir plus librement notamment au niveau de la forme.

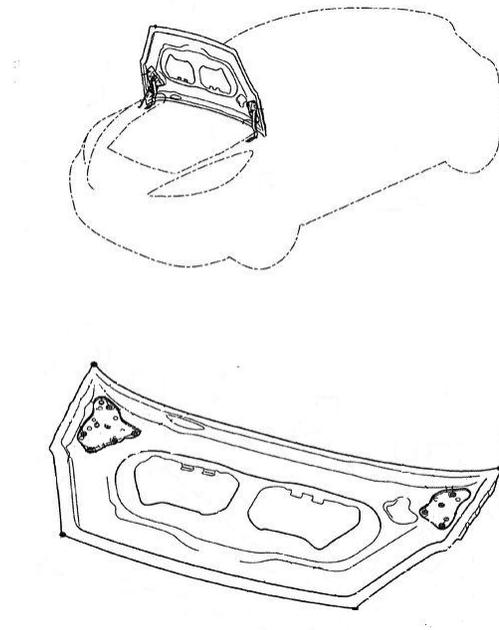


FIGURE 11 – Exemple d'une doublure de capot de voiture (image issue de <http://www.google.com/patents/EP2128002B1?cl=fr>)

L'objectif n'étant pas de traiter un exemple dans toute sa complexité mais surtout de mettre

en évidence l'emploi de la méthode d'optimisation, on prendra plusieurs hypothèses simplificatrices qui seront détaillées dans la suite.

4.1 Définition du problème d'optimisation

Pour poser le problème d'optimisation topologique il nous faut au préalable définir le *modèle*, le *critère* et l'*ensemble admissible* (cf partie 2).

En ce qui concerne le *modèle*, on se place dans le cadre de l'élasticité linéaire.

Tout d'abord on choisit la forme initiale représentée sur la figure ci-dessous. La frontière est séparée en deux parties Γ et Γ_D , on a donc $\partial\Omega = \Gamma \cup \Gamma_D$. On considère, par ailleurs, que la frontière Γ_D est fixée. On applique des efforts f uniformément répartis sur la surface Ω . L'objectif de l'optimisation consiste à déterminer la forme qui répond le mieux à ces efforts.

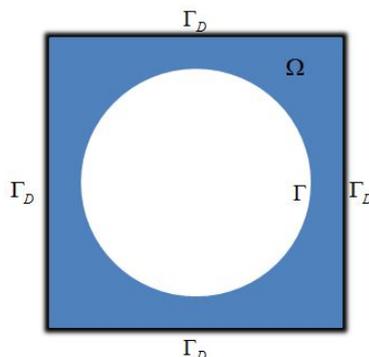


FIGURE 12 – Forme initiale Ω_0

Ceci nous permet de définir le problème d'élasticité linéaire suivant :

$$\left\{ \begin{array}{ll} -\operatorname{div}\sigma = f & \text{sur } \Omega \\ \sigma = 2\mu\epsilon(u) + \lambda\operatorname{tr}(\epsilon(u))I & \text{sur } \Omega \\ u = 0 & \text{sur } \Gamma_D \\ \sigma n = 0 & \text{sur } \Gamma \end{array} \right.$$

Définissons les différents éléments de ce système.

u correspond au champ de déplacement de la structure soumis aux efforts f .

Le tenseur de Green ϵ correspond aux déformations locales, physiquement il correspond à la fois aux variations de volumes (sans changement de forme) et aux distortions (déformations sans changement de volume). Sous l'hypothèse des petites déformations, que l'on prendra ici, il est défini par la relation $\epsilon = \frac{1}{2}(\operatorname{grad}(u) + {}^t\operatorname{grad}(u))$.

σ est le tenseur des contraintes de Cauchy, physiquement les termes diagonaux correspondent à des contraintes normales tandis que les termes extradiagonaux sont les contraintes de cisaillement. σ permet de déterminer le vecteur contrainte associé à tout éléments de surface, défini par $T = \sigma n$, n étant la normale extérieure à la surface considérée.

λ et μ sont les coefficients de Lamé qui permettent de relier σ et ϵ au travers de la loi de Hooke qui correspond à la seconde equation de ce système.

Ces différentes conditions nous permettent de commencer à définir l'*ensemble des formes admissibles*, il est cependant nécessaire de fixer aussi les caractéristiques cibles que l'objet final devra posséder. On considère que la doublure de capot est contenue dans un carré de côté 1. Toutes les grandeurs seront considérées comme adimensionnée (ainsi par la suite on considèrera que le carré de côté 1 sera de masse égale à 1, la masse est, dans notre cas, assimilée à l'aire de la surface). De plus on se fixera une masse cible valant $\frac{1}{2}$.

Nous avons choisi une forme initiale vérifiant la condition de masse pour s'intéresser plus particulièrement à l'évolution de la topologie.

Ainsi on définit l'ensemble des formes admissibles de la manière suivante :

$$O_{admissibles} = \left\{ \Omega / Aire(\Omega) = \frac{1}{2}, \Gamma_D \text{ fixée} \right\}$$

La figure ci-dessous représente des éléments de l'ensemble des formes admissibles considéré.

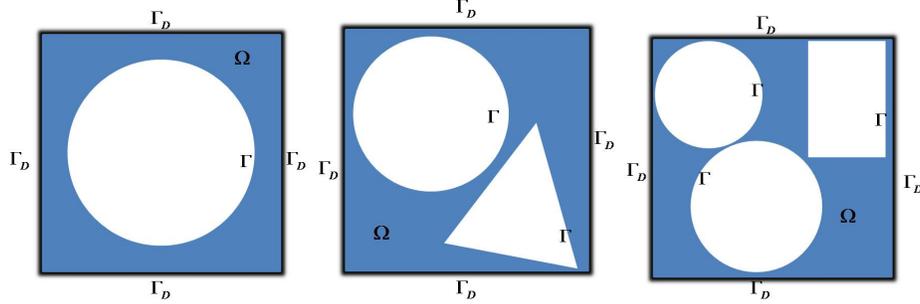


FIGURE 13 – Exemple d'éléments admissibles

Finalement le critère que l'on se fixe est la compliance. L'objectif sera, dans notre cas, de minimiser ce critère.

Définissons la notion de compliance.

La compliance est l'inverse de la raideur, cela correspond à une mesure de la capacité de réponse d'un système mécanique soumis à un effort. Elle est défini par la relation suivante :

$$\epsilon = S : \sigma$$

où S est le tenseur de souplesse (tenseur des compliances d'élasticité).

On l'a défini ici par :

$$J_1(\Omega) = \int_{\Omega} f \cdot u = \int_{\Omega} \sigma : \epsilon = \int_{\Omega} (2\mu\epsilon : \epsilon + \lambda \text{tr}(\epsilon)^2)$$

Physiquement, J_1 correspond au travail des forces extérieures.

Par ailleurs, il est plus intéressant et plus simple de choisir de pénaliser la condition de masse imposée ($Aire(\Omega) = \frac{1}{2}$) que de parcourir l'ensemble des formes de surface fixée. Plus précisément, nous choisissons de faire passer cette condition dans le critère à minimiser sous la forme suivante :

$$J_2(\Omega) = \frac{1}{\xi} (Aire(\Omega) - \frac{1}{2})^2$$

Où ξ est choisi en accord avec le pas de temps Δt de l'algorithme d'optimisation. Le choix de ξ sera fixé par les tests numériques actuellement en cours.

Cette dernière condition permet de s'assurer que les variations de formes ne seront pas trop subites en s'assurant que les variations du gradient de forme, donnant le champ de vecteur F , ne soient pas trop importantes.

Le nouveau critère correspond alors à la somme de la compliance et de cette nouvelle fonction. Il est alors clair que minimiser cette somme reviendra à trouver la forme minimisant les deux éléments J_1 et J_2 car ces deux termes sont positifs, il est par ailleurs immédiat de voir que $J_2(\Omega)$ sera minimal pour Ω d'aire proche de $\frac{1}{2}$, la forme optimale aura donc bien la masse souhaitée.

Il est alors nécessaire de définir la notion de dérivée par rapport au domaine qui est essentielle dans un problème d'optimisation lié à la forme donc au domaine.

Soit alors un domaine de référence Ω_0 du plan.

On considère des ensembles de la forme $\Omega = (Id + \theta)\Omega_0$ où Id est l'identité du plan et $\theta \in W^{1,\infty}(\mathbb{R}^2, \mathbb{R}^2)$ (que l'on définit plus bas).

Ainsi, on peut interpréter la fonction θ comme le champ de vecteur transportant le domaine de référence (on verra plus loin que θ correspondra en fait au champ F défini dans la partie 3).

Nous rappelons que $W^{1,\infty}(\mathbb{R}^2, \mathbb{R}^2)$ correspond à l'ensemble des fonctions lipschitziennes de \mathbb{R}^2 dans \mathbb{R}^2 qui sont uniformément bornées sur \mathbb{R}^2 et dont le gradient est aussi uniformément borné sur \mathbb{R}^2 . On munit cet espace de la norme définie par :

$$\|\phi\|_{W^{1,\infty}(\mathbb{R}^2, \mathbb{R}^2)} = \sup_{x \in \mathbb{R}^2} (|\phi(x)|_{\mathbb{R}^2} + |\nabla \phi(x)|_{\mathbb{R}^2 \times \mathbb{R}^2}).$$

Pour des fonctions θ vérifiant $\|\theta\|_{W^{1,\infty}(\mathbb{R}^2, \mathbb{R}^2)} < 1$, on peut montrer que $u = Id + \theta$ est un difféomorphisme, c'est-à-dire que u est une bijection et u et u^{-1} sont de classes C^1 . Pour la démonstration nous renvoyons le lecteur à l'ouvrage [4] de G. Allaire. Sous ces conditions, nous pouvons alors définir la dérivée d'une fonction $J(\Omega)$ en Ω_0 comme la différentielle de l'application $\theta \rightarrow J((Id + \theta)\Omega_0)$ en 0. Il existe une forme linéaire, notée $J'(\theta)$, continue sur $W^{1,\infty}(\mathbb{R}^2, \mathbb{R}^2)$ telle que :

$$J((Id + \theta)\Omega_0) = J(\Omega_0) + J'(\Omega_0)(\theta) + o(\theta) \text{ avec } \lim_{\theta \rightarrow 0} \frac{o(\theta)}{\|\theta\|_{W^{1,\infty}(\mathbb{R}^2, \mathbb{R}^2)}} = 0$$

On peut alors définir le gradient de la compliance et de la condition de masse par (ici Ω_0 correspond à la forme initiale définie plus haut) :

$$J'_1(\Omega_0)(\theta) = - \int_{\Gamma} \theta(s) \cdot n(s) (2\mu \epsilon(u(s)) : \epsilon(u(s)) + \lambda \text{tr}(\epsilon(u(s))))^2 ds$$

$$J'_2(\Omega_0)(\theta) = \frac{2}{\xi} (\text{Aire}(\Omega) - \frac{1}{2}) \int_{\partial\Omega_0} \theta(s) \cdot n(s) ds$$

Nous renvoyons le lecteur à la lecture de [4] pour plus de détails et les démonstrations de ces résultats.

La minimisation du *critère* implique que après transformation nous ayons $J((Id + \theta)\Omega_0) \leq J(\Omega_0)$ et donc nécessairement $J'(\Omega_0)(\theta) \leq 0$. On choisira alors θ vérifiant :

$$\theta = \begin{cases} (2\mu \epsilon(u) : \epsilon(u) + \lambda \text{tr}(\epsilon(u)^2 - \frac{2}{\xi} (\text{Aire}(\Omega) - \frac{1}{2}))n & \text{sur } \Gamma \\ 0 & \text{sur } \Gamma_D \end{cases}$$

Cette relation ne définit θ que sur la frontière, or, comme cela est précisé dans la partie 3, il est nécessaire pour l'algorithme de l'étendre sur tout Ω . La fonction θ ainsi étendue correspondra au champ de vecteur F de la partie précédente. L'extension de la fonction peut être effectuée à l'aide de la résolution du système suivant qui nous assure que θ vérifiera les conditions de dérivation vu plus haut :

$$\begin{cases} \Delta \theta = 0 & \text{sur } \Omega \\ \theta = 0 & \text{sur } \Gamma_D \\ \frac{\partial \theta}{\partial n} = (2\mu \epsilon(u) : \epsilon(u) + \lambda \text{tr}(\epsilon(u)^2 - \frac{1}{\xi} (\text{Aire}(\Omega) - \frac{1}{2}))n & \text{sur } \Gamma \end{cases} \quad (13)$$

Après avoir posé le problème, nous allons en détailler les différentes étapes de la résolution numérique.

4.2 Résolution numérique

La méthode des level-set est une méthode de gradient itérative sur la variable θ . Le principe de la résolution numérique est donc simple, à l'étape n on calcule le gradient de la fonction *critère* fixée, on en déduit la fonction θ_n en résolvant le système (13), on pose alors $\Omega_{n+1} = (Id + \theta_n)\Omega_n$.

Nous allons maintenant détailler l'implémentation numérique de cette résolution. Celle-ci repose sur un interfaçage entre deux logiciels : Matlab et Salome-Meca, ce dernier est une interface graphique, on s'en servira pour faire appel au solveur Code Aster. Sous Matlab on effectue la modification de la forme en fonction des données fournies par le calcul d'élasticité linéaire effectué grâce au solveur Code Aster. Le calcul est lui-même effectué sur le maillage de la forme transmis par

Matlab. Pour plus de clarté, le cycle d'optimisation est représenté dans le diagramme ci-dessous. Nous détaillerons ensuite chacune des étapes.

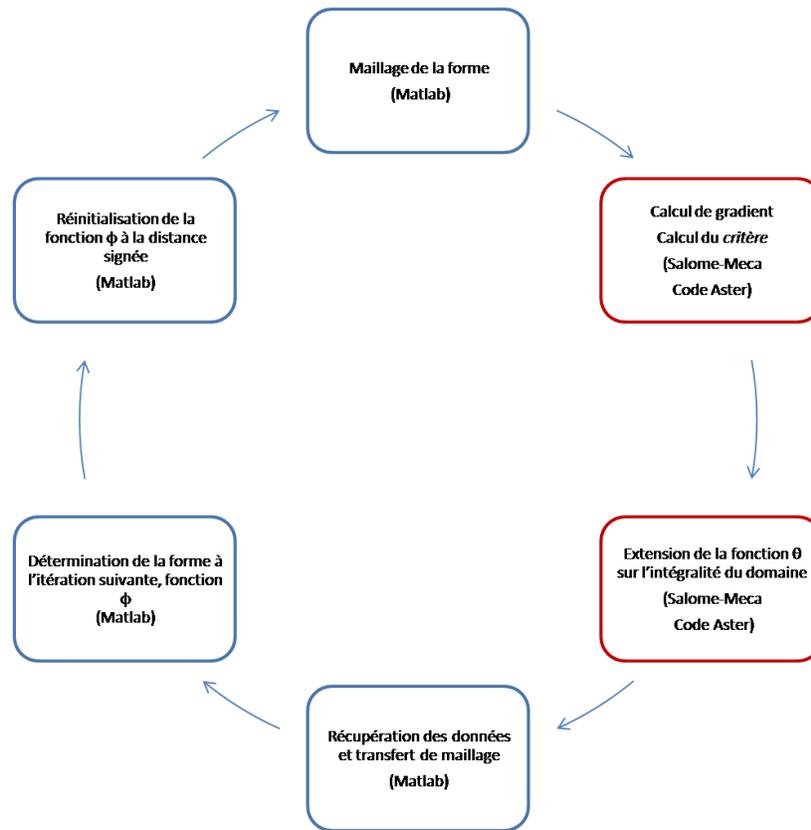


FIGURE 14 – Boucle d'optimisation

Maillage de la forme

L'une des méthodes pour transférer la géométrie de l'objet depuis le logiciel Matlab jusqu'au solveur de Salome-Meca, le Code Aster, est de lui faire passer le domaine Ω sous la forme d'un maillage. Dans les méthodes classiques de level-set, le principe est de garder tout au long de l'algorithme un maillage fixe sur lequel nous faisons évoluer la forme. Cependant cette méthode est plus longue à mettre en place et compte-tenu du temps imparti restant il a été nécessaire de choisir une méthode certes plus coûteuse mais dont l'implémentation est plus simple. On choisit donc de remailler la forme, par des éléments triangulaires, sous Matlab à chaque passage dans la boucle. Pour cela on utilise le mailleur en deux dimensions *Triangle* développé par J.R. Shewchuk (il est disponible à l'adresse suivante <http://www.cs.cmu.edu/~quake/triangle.html>).

Calcul du gradient et extension de la fonction θ_n

Le maillage est lu par le solveur sous la forme d'un fichier texte. Sur ce maillage les différents calculs nécessaires à la détermination du gradient de la fonction *critère* (déplacements, déformations,...) sont effectués. On détermine ainsi la fonction θ_n sur la frontière Γ_D . Ils sont ensuite utilisés par le solveur Code Aster pour résoudre le système d'extension de la fonction θ_n à l'intégralité du domaine Ω_n . Celle-ci est ensuite récupérée sous la forme d'un fichier texte contenant la valeur de la fonction en chacun des noeuds du maillage.

Transfert de maillage

Il est ensuite nécessaire de traduire cette fonction depuis le maillage utilisé sous Salome-Meca sur le quadrillage que l'on utilise sous Matlab pour déterminer la géométrie de l'objet à l'étape suivante. Ce transfert, effectué sous Matlab, se fait par interpolation. Pour chaque point du quadrillage on

détermine dans quel triangle du maillage il se situe, et on interpole la valeur en ce point à partir de la valeur en chacun des trois sommets.

Détermination de la forme à l'étape $n + 1$

A cette étape, on a récupéré le champ de vecteur F_n (il correspond, sur le domaine Ω_n , à la fonction θ_n) qui va permettre de déterminer la fonction ϕ_{n+1} par l'emploi de l'algorithme défini dans la partie 3.2.3. On rappelle qu'alors la nouvelle forme correspond à la ligne de niveau 0 de cette fonction.

Réinitialisation à la distance signée

Cette étape sert à s'assurer de la régularité de la fonction ϕ_{n+1} pour le calcul à l'étape $n + 2$. On utilise pour cela l'algorithme détaillé dans la partie 3.2.4. On recommence ensuite la boucle en remaillant la nouvelle forme.

On arrête la boucle lorsque la norme de la différence entre deux fonctions consécutives devient inférieure à un critère ε fixé :

$$|\phi_{n+1} - \phi_n| < \varepsilon$$

De même que pour le ξ , ce sont les tests numériques qui permettent de caler la valeur de ε la plus adaptée au problème.

Les essais numériques étant en cours de réalisation, il nous est malheureusement impossible de vous présenter dans ce rapport les résultats de l'application de la boucle d'optimisation à l'exemple présenté dans la partie précédente mais nous espérons pouvoir les présenter lors de la soutenance finale.

5 Bilan

Nous revenons, ici, plus particulièrement sur le déroulement du projet et sur son état d'avancement. Nous dressons ainsi le bilan de ce qui a été fait et de ce qu'il pourrait être envisageable et intéressant de faire dans la continuité de ce projet.

Les objectifs de ce projet étaient de prendre en main une méthode numérique relativement récente et d'en étudier l'application dans le domaine mécanique. En outre le coeur du projet consistait plus en l'étude de la méthodologie qu'en l'obtention d'un produit final. Ce projet s'est déroulé en plusieurs étapes.

Tout d'abord une phase d'apprentissage, il a non seulement fallu découvrir et se familiariser avec la méthode, ce qui va de pair avec la phase de recherche bibliographique, mais il a aussi été nécessaire de prendre en main les deux logiciels. Si cela a été simple avec Matlab, la prise en main de Salome-Meca a été rendu plus difficile pour des raisons techniques. N'ayant pas pu installer le logiciel sur notre ordinateur, nous n'avons pu travailler sur Salome-Meca qu'avec les ordinateurs du rez-de-chaussée du E6, ce qui a ralenti l'avancement du projet.

L'étape suivante a été d'adapter l'ensemble des algorithmes sous Matlab. Durant cette étape, tout un cheminement a été effectué pour comprendre le choix de la forme de l'algorithme final au dépens d'autres algorithmes plus classiques et surtout plus simples. Pour réaliser cela il a fallu tester chaque méthode, comprendre leurs limitations et les solutions à apporter.

Il a ensuite été question de définir le problème à étudier en détail. Puis, pour finir, il s'agissait de réaliser l'interfaçage des deux logiciels, de rechercher les formats de transfert les plus adaptés pour les données du problème et, de manière plus générale, de mettre en place de la boucle. A l'heure où nous rédigeons ce rapport nous sommes encore dans cette dernière étape qui devrait être terminée pour la soutenance.

La tâche principale qu'il resterait à faire, dans la continuité de ce projet, consisterait en l'optimisation de la boucle algorithmique. Notamment au niveau du choix de maillage, dans l'idéal il faudrait que l'algorithme tourne sur un maillage fixe, ce qui correspondrait plus à l'esprit des méthodes level-set et qui permettrait un gain de temps précieux en supprimant l'étape, coûteuse, de remaillage à chaque itération. Cependant si nous reconnaissons que la boucle d'optimisation

est loin d'être optimisée, nous rappelons que, compte-tenu du temps imparti, nous avons préféré concentrer nos efforts afin de parvenir à faire tourner une boucle d'optimisation sur l'exemple.

Une autre piste de réflexion intéressante serait d'étendre l'étude à des problèmes plus généraux en trois dimensions.

Références

- [1] J. A. Sethian, *Level set methods and fast marching methods : evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science*, ser. Cambridge monographs on applied and computational mathematics. Royaume-Uni : Cambridge, U.K. : Cambridge University Press, cop. 1999, 1999.
- [2] R. Giovanni and P. Smereka, “A remark on computing distance functions,” *Journal of Computational Physics*, p. 20, 2000.
- [3] G. Allaire, “Introduction à l’optimisation de structures,” 2010. [Online]. Available : <http://www.cmapx.polytechnique.fr/allaire/map562/amphi1.pdf>
- [4] G. Allaire and M. Schoenauer, *Conception optimale de structures*. New York : Springer, 2007.

Table des figures

1	Deux topologies, (b) et (c) ont la même topologie, (a) possède un trou en moins donc est de topologie différente.	4
2	Définition de la fonction ϕ , les flèches représentent le champ F (image issue de [2]) .	6
3	Résultats de simulation sous Matlab. (a),(b) correspondent au schéma progressif, (c),(d) correspondent au schéma rétrograde et (e),(f) au schéma centré. (a), (c) et (e) sont les résultats après 5 itérations (b), (d) et (f) sont ceux obtenus après 15 itérations. Sur chaque graphique le cercle rouge correspond au foyer initial.	8
4	Résultats de simulation sous Matlab après 20 itérations. (a) correspond au schéma progressif et (b) au schéma rétrograde. Les cercles rouge correspondent au foyer initial.	9
5	Résultats des simulations après 30 itérations, le cercle rouge correspond au foyer initial. (a) $\sigma = 2$, (b) $\sigma = 1$ et (c) $\sigma = 0.1$	9
6	Caractéristiques du système avec $F = \frac{1}{3}$	10
7	Caractéristiques pour F dépendant de la variable spatiale.	11
8	Résultats de la résolution numérique après 50 itérations, les cercles rouges correspondent au foyer initial. (a) $h = 0.01$, (b) $h = 0.02$, (c) $h = 0.05$ et (d) $h = 0.1$	14
9	Résultats de simulation, (a) après 100 itérations, (b) après 200 itérations et (c) après 500 itérations.	14
10	Résultats de la simulation, (a) lignes de niveau de ϕ^0 , (b) lignes de niveau après 200 itérations et (c) lignes de niveau après 500 itérations	16
11	Exemple d’une doublure de capot de voiture (image issue de http://www.google.com/patents/EP2128002B1?cl=fr)	16
12	Forme initiale Ω_0	17
13	Exemple d’éléments admissibles	18
14	Boucle d’optimisation	20