

# TD4 - Multiplicateurs de Lagrange

Alexandra Bac

Optimisation - MIRA  
ESIL - IRM 2ème année

## Algorithme de Newton-Raphson

On veut résoudre un système de  $n$  équations non linéaires à  $n$  inconnues :

$$\begin{cases} f_1(x_1, \dots, x_n) = 0 \\ \vdots \\ f_n(x_1, \dots, x_n) = 0 \end{cases}$$

Par exemple :

$$\begin{cases} 4\pi r + 2\pi h + 2\lambda\pi r h & = 0 \\ 2\pi r + \lambda\pi r^2 & = 0 \\ \pi r^2 h & = 1 \end{cases} \quad (1)$$

S'il s'agissait de résoudre une équation  $f(x) = 0$  (on a une seule inconnue), on pourrait utiliser l'algorithme de Newton et itérer :

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

à partir d'un point  $x_0$  proche de la solution.

Mais ici, il s'agit bien de résoudre  $f(\vec{x}) = \vec{0}$ , mais avec :

$$\begin{aligned} f : \mathbb{R}^n &\rightarrow \mathbb{R}^n \\ \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} &\mapsto \begin{pmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_n(x_1, \dots, x_n) \end{pmatrix} \end{aligned}$$

L'algorithme de Newton peut en fait se généraliser à la dimension  $n$  de la manière suivante (algorithme de Newton-Raphson) : si  $f$  est  $\mathcal{C}^2$ , alors, l'algorithme :

$$\begin{cases} \text{Soit } x_0 \in \mathbb{R}^n \\ \text{On itère :} \\ \quad f'(x_k)(x_{k+1} - x_k) = -f(x_k) \\ \text{jusqu'à } \|x_{k+1} - x_k\| < \varepsilon \end{cases}$$

où  $f'(x)$  est le jacobien de  $f$ , c'est-à-dire la matrice :

$$\begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$

Récupérez l'algorithme fourni en matériel de TP :

`function [res] = NewtonRaphson(f, fprime, X0, eps, itermax)` Vous l'utiliserez dans la suite du TP pour résoudre des systèmes non linéaires.

**Exercice 1** (Multiplicateurs de Lagrange). On considère une surface implicite :

$$\mathcal{S} = \{(x, y, z) \in \mathbb{R}^3; f(x, y, z) = 0\}$$

et un point  $u = (u_1, u_2, u_3) \in \mathbb{R}^3$ . On veut calculer le projeté de  $u$  sur la surface (donc le point le plus proche de  $u$  appartenant à la surface).

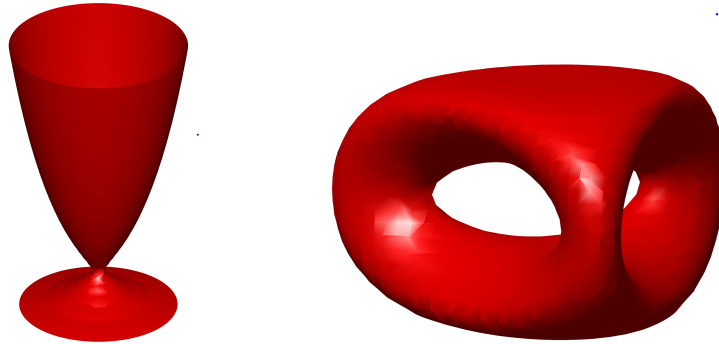
1. Comme nous l'avons vu au TP précédent, le problème comme un problème d'optimisation. Formuler le lagrangien du problème et le système à résoudre pour trouver ce projeté.
2. Cas d'un plan :
  - (a) La surface est alors donnée par  $f(x, y, z) = ax + by + cz - d = 0$ , montrer que le système précédent donne un système linéaire (programmation quadratique) ?
  - (b) Représenter graphiquement la surface et le point en utilisant la fonction `plotImplicit3D` fournie en matériel de TP.
  - (c) Résoudre et représenter le résultat.
3. Cas d'une quadrique :
  - (a) La surface a pour équation  $f(x, y, z) = 0$  où  $f$  est un polynôme de degré 2 (on pourra par exemple tester  $x^2 + 2y^2 + 3z^2 = 1$ ). Que devient le système précédent (vous explicitez les calculs) ?
  - (b) Représenter graphiquement la surface et le point en utilisant la fonction `plotImplicit3D` fournie en matériel de TP.
  - (c) Résoudre en utilisant l'algorithme de Newton Raphson de l'exercice précédent.
4. Testez pour différentes surfaces implicites :
  - Une quadrique
$$\alpha x^2 + \beta y^2 + \gamma z^2 - 1 = 0$$
(jouez sur les paramètres en les prenant positifs, négatifs voire nuls pour faire varier les surfaces dans la famille des quadriques).

- Une surface de genre 2 comme celle ci-dessous à droite

$$2y(y^2 - 3x^2)(1 - z^2) + (x^2 + y^2)^2 - (9z^2 - 1)(1 - z^2) = 0$$

- Le verre ci-dessous à gauche

$$x^2 + y^2 - (\ln(z + 3.2))^2 - 0.02 = 0$$



Aide pour les calculs du cas quadratique :

$$L(X, \lambda) = \|X - U\|^2 + \lambda f(X) \quad f(X) = X^t \cdot A \cdot X + B^t X - 1$$

$$\nabla L(X, \lambda) = \begin{bmatrix} 2(X - U) + \lambda 2A \cdot X + \lambda B \\ \frac{X^t \cdot A \cdot X + B^t X - 1}{0} \end{bmatrix}$$

$$H(L)(X, \lambda) = \begin{bmatrix} 2(I + \lambda A) & | & 2AX + B \\ \hline (2AX + B)^t & | & 0 \end{bmatrix}$$

**Exercice 2** (Méthode de pénalisation). Pour résoudre un problème d'optimisation sous contraintes trop complexe pour les multiplicateurs de Lagrange, on peut utiliser une **méthode de pénalisation** qui approche la solution par un problème sans contraintes.

On considère une surface implicite :

$$\mathcal{S} = \{(x, y, z) \in \mathbb{R}^3; f(x, y, z) = 0\}$$

et un point  $u = (a, b, c) \in \mathbb{R}^3$ . On veut calculer le projeté de  $u$  sur la surface (donc le point le plus proche de  $u$  appartenant à la surface).

- fonction à minimiser :  $J(X) = \|X - u\|^2$
- contraintes :  $f(X) = 0$  où  $f$  est l'équation implicite de la surface

Pour contourner cette minimisation sous contrainte, une heuristique consiste à introduire un  $\alpha$  petit (prendre  $\alpha = 10^{-2}$  pour commencer) et on construit la fonction

$$g(X) = J(X) + \frac{1}{\alpha} (f(X))^2$$

que l'on minimise (avec la méthode de descente de gradient ou de gradients conjugués).

1. Testez la minimisation sur les surfaces implicites précédentes (vous représenterez la surface et la descente en utilisant la fonction :

`plotImplicit3D(f, min, max, pas)`

également fournie dans le matériel de TP.

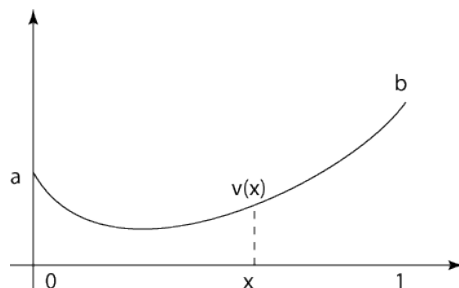
Vous aurez besoin pour pouvoir appliquer la méthode de calculer le gradient de  $g$ .

$$J(X) = (X - u)^t \cdot (X - u) \Rightarrow \nabla J(X) = 2(X - u)$$

$$\nabla (f(X)^2) = 2f(X)\nabla f(X)$$

#### Bonus

**Exercice 3** (Modélisation d'un fil pesant (Ciarlet, Joly) - pénalisation). On veut trouver la position d'un fil pesant fixé à ses deux extrémités. Celle-ci est déterminée comme minimum de l'énergie potentielle de ce fil :



L'élévation est déterminée par  $v(x)$ , on pose  $v(0) = a$ ,  $v(1) = b$ , et la longueur du fil est  $L$ .

Rappelons que l'intégrale d'une fonction  $c : [0, 1] \rightarrow \mathbb{R}$  le long d'un arc paramétrique d'équation  $f(t)$  avec  $t \in [0, 1]$  est donnée par :

$$\int_0^1 c(t) \|f'(t)\| dt$$

Par conséquent, l'énergie potentielle du fil (intégrale de l'élévation le long du fil) est donnée par :

$$\mathcal{E} = \int_0^1 v(x) \sqrt{1 + v'(x)^2} dx$$

La position du fil correspond au minimum de cette énergie sous les contraintes suivantes :

$$\begin{cases} v(0) = a \\ v(1) = b \\ \int_0^1 \sqrt{1 + v'(x)^2} dx - L = 0 \quad (\text{longueur du fil}) \end{cases}$$

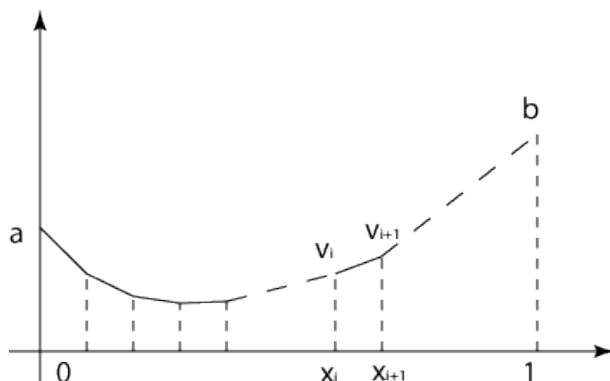
La méthode de **pénalisation** consiste à inclure cette contrainte dans la fonction à minimiser : soit  $\varepsilon > 0$ , on minimise

$$\mathcal{E}' = \int_0^1 v(x) \sqrt{1 + v'(x)^2} dx + \frac{1}{\varepsilon} \left( \int_0^1 \sqrt{1 + v'(x)^2} dx - L \right)^2$$

Pour pouvoir appliquer nos algorithmes d'optimisation, il est nécessaire de se ramener à un espace de dimension finie. On va donc discrétiser et considérer que  $v$  est affine par morceaux :

- soit  $h = 1/N$
- on pose  $x_i = ih$  avec  $i \in \{0 \dots N\}$  (discrétisation de l'axe des  $x$ )
- on appellera  $v_h$  le profil correspondant. Soient  $v_i$  les valeurs de  $v_h$  en  $x_i$ . Sur le segment  $[x_i, x_{i+1}]$  :

$$v_h(x) = v_i + (x - x_i) \frac{(v_{i+1} - v_i)}{h}$$



Fonction discrétisée  $v_h$

On a donc un problème d'optimisation de dimension  $N - 1$  donc les variables sont  $v = (v_1, \dots, v_{N-1})$ . Dans la fonction à minimiser, sur le segment  $[x_i, x_{i+1}]$ , on a :  $v'_h(x) = \frac{(v_{i+1} - v_i)}{h}$ , puis :

$$J(v) = \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} v_h(x) \sqrt{1 + v'_h(x)^2} dx + \frac{1}{\varepsilon} \left( \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} \sqrt{1 + v'_h(x)^2} dx - L \right)^2$$

Posons :  $g(v, i) = \sqrt{1 + \left(\frac{v_{i+1} - v_i}{h}\right)^2}$

$$J(v) = \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} \left( v_i + (x - x_i) \frac{(v_{i+1} - v_i)}{h} \right) g(v, i) dx + \frac{1}{\varepsilon} \left( \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} g(v, i) dx - L \right)^2$$

$$J(v) = \sum_{i=0}^{N-1} g(v, i) \left[ v_i \cdot x + \left( \frac{x^2}{2} - x_i \cdot x \right) \frac{(v_{i+1} - v_i)}{h} \right]_{x_i}^{x_{i+1}} + \frac{1}{\varepsilon} \left( \sum_{i=0}^{N-1} g(v, i) (x_{i+1} - x_i) - L \right)^2$$

$$J(v) = \sum_{i=0}^{N-1} g(v, i) \left( v_i \underbrace{(x_{i+1} - x_i)}_h + \frac{v_{i+1} - v_i}{h} \frac{(x_{i+1} - x_i)^2}{2} \right) + \frac{1}{\varepsilon} \left( \sum_{i=0}^{N-1} g(v, i) (x_{i+1} - x_i) - L \right)^2$$

$$J(v) = \sum_{i=0}^{N-1} g(v, i) \left( h v_i + \frac{h(v_{i+1} - v_i)}{2} \right) + \frac{1}{\varepsilon} \left( \sum_{i=0}^{N-1} g(v, i) h - L \right)^2$$

$$J(v) = \sum_{i=0}^{N-1} g(v, i) \left( \frac{h(v_{i+1} + v_i)}{2} \right) + \frac{1}{\varepsilon} \left( \sum_{i=0}^{N-1} g(v, i) h - L \right)^2$$

1. Récupérez en matériel de TP (Materiel3bis) contenant les fonctions de calcul de l'énergie ( $g, J, \nabla J \dots$ ) et résoudre en utilisant la pénalisation et la méthode du gradient.
2. Exprimer le Lagrangien du problème et résoudre par les multiplicateurs de Lagrange et Newton-Raphson.
3. Comparer les résultats et temps d'exécution.

Pour cela, vous devrez calculer le gradient de la fonction  $J$ . On a :

$$\frac{\partial g(v, i)}{\partial v_i} = -\frac{v_{i+1} - v_i}{h^2 \cdot g(v, i)}$$

et

$$\frac{\partial g(v, i)}{\partial v_{i+1}} = \frac{v_{i+1} - v_i}{h^2 \cdot g(v, i)}$$

Ce sont les deux seules dérivées partielles non nulles de  $g(v, i)$ . Par conséquent, comme :

$$J(v) = \frac{h}{2} \sum_{i=0}^{N-1} g(v, i) (v_{i+1} + v_i) + \frac{1}{\varepsilon} \left( \sum_{i=0}^{N-1} g(v, i) h - L \right)^2$$

On obtient :

$$\begin{aligned} \frac{\partial J}{\partial v_k} &= \frac{h}{2} \left( g(v, k) + g(v, k-1) \right) + \frac{1}{h^2} \left( \frac{v_k^2 - v_{k-1}^2}{g(v, k-1)} - \frac{v_{k+1}^2 - v_k^2}{g(v, k)} \right) \\ &+ \frac{2}{\varepsilon} \left( \sum_{i=0}^{N-1} g(v, i) h - L \right) \cdot \frac{1}{h} \cdot \left( \frac{v_k - v_{k-1}}{g(v, k-1)} - \frac{v_{k+1} - v_k}{g(v, k)} \right) \end{aligned}$$