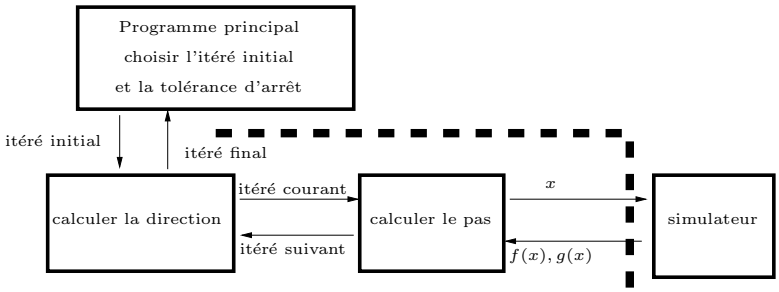
Ensimag 2A – automne 2015
TDs optimisation
Alejandro Blumentals, Franck Iutzeler, Jérôme Malick

TP 1 – structure d'un programme d'optimisation

Objectif : Cette séance consiste à réaliser en `scilab` un programme d'optimisation simple (méthode de gradient à pas constant) en insistant sur sa structure modulaire.



Rappel : L'ensemble d'un programme d'optimisation se présente avec deux parties bien distinctes :
- un simulateur : il est chargé de calculer la fonction (ainsi que le gradient et éventuellement le hessien) en chaque point décidé par l'algorithme. Souvent la fonction à minimiser n'est connue que via ce simulateur.
- l'algorithme proprement dit : il comporte deux boîtes principales correspondant au calcul de la direction, et au calcul du pas.

Le but de cette séance est ainsi de mettre en musique ces différents composants d'une méthode de descente. Pour faire simple, on choisit dans un premier temps la direction de descente donnée par le gradient ($d = -\nabla f(x)$) et un pas de descente constant (par exemple $t = 1$).

Exercice 1 – Fonctions-tests. Considérons les deux fonctions suivantes :

$$\begin{aligned} f_1(x) &= \sum_{k=1}^{n} k\, {x_k}^2 \qquad \text{pour } x \in \mathbb{R}^n, \\ f_2(x) &= (1-x_1)^2 + 100(x_2 - {x_1}^2)^2 \qquad \text{avec } x = (x_1, x_2) \in \mathbb{R}^2. \end{aligned}$$

a) Donner les minimums (globaux) de ces fonctions.

b) Donner l'expression des gradients de f_1 et f_2.

Exercice 2 – Simulateurs. Écrire un simulateur pour chacune des fonctions ci-dessus. Les simulateurs se présentent sous le format

```
function [f,g] = nom_du_sim(x),
```

avec `x` la valeur de la variable pour laquelle il faut évaluer f et son gradient, `f` qui vaut $f(x)$ et `g` qui vaut $g(x) = \nabla f(x)$. Recommandation : travailler avec des vecteurs colonnes !

Exercice 3 – Lignes de niveaux. Visualiser la géométrie des fonctions en affichant quelques lignes de niveaux de f_1 pour $n = 2$ et pour f_2. Le tracé des lignes de niveaux se fait avec la fonction scilab `contour`. Vous pouvez récupérer directement en ligne une fonction pour tracer, à l'adresse `http//bipop.inrialpes.fr/people/malick/Docs/tracer.sci` . Pour la fonction f_2, tracer les lignes de niveaux $[1, 2, 5, 10, 20, 30, 50, 100, 200]$ pour bien visualiser la géométrie de la fonction (choisir une bonne résolution `nx,ny`).

Exercice 4 – Optimiseur (direction et pas ensemble). Focalisons-nous sur un algorithme d'optimisation simple : la méthode du gradient à pas constant. À chaque itération, on fixe deux choses : la direction $d = -g(x)$, puis le pas $t > 0$.

Faire donc une fonction qui, parmi ses paramètres d'entrée, comportera x_0 l'itéré courant et `sim` le nom formel du simulateur. Vous pouvez faire afficher à cette fonction des informations à chaque itération (comme la valeur de f, $\|g\|$, nombre d'évaluations du simulateur,...).

Exercice 5 – Tests.

a) Appliquer le programme précédent pour minimiser f_1 en dimension 2 à partir de $x_0 = (-1, -1)$. Tester plusieurs pas de descente : commencer par $t = 0.1$, puis $t = 0.5$ et $t = 1$. Faire des tests similaires en dimension supérieure.

b) De même pour f_2 : tester differents points initiaux, différentes valeurs de t et différents nombres maximaux d'itérations. Prendre en particulier $x_0 = (-1, 1.2)$, $t = 0.001$ et un nombre d'itérations très important, genre 20000.

c) Avec les fonctions en 2 dimensions (f_1 pour $n = 2$ puis f_2), visualiser la suite des itérés x_k en même temps que les courbes de niveaux de la fonction.

d) Améliorer la recherche linéaire, par exemple en diminuant t jusqu'à obtenir un x_{k+1} meilleur que x_k, ou alors en prenant $t = 1/k$ à la k-ième itération.

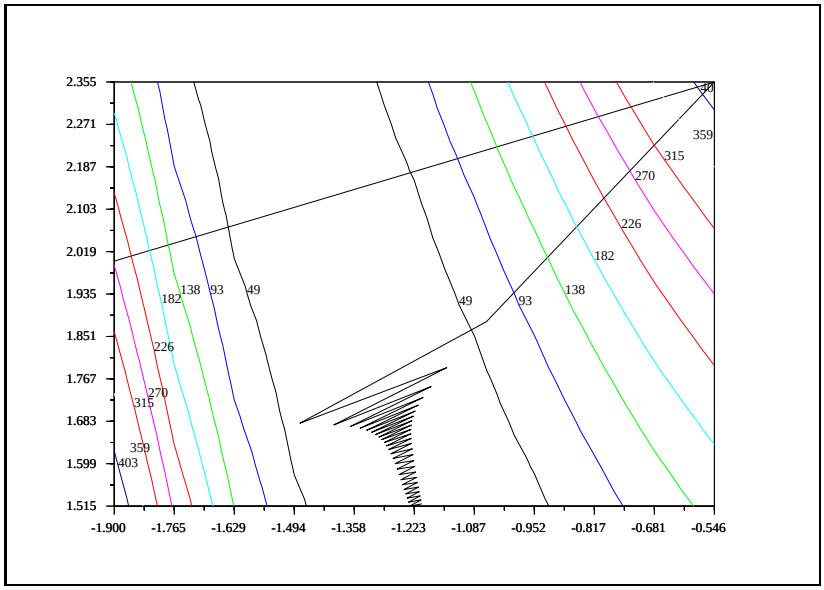


Figure 1 une trajectoire des x_k générés par la méthode de gradient à pas constant