Année 2021-2022
Emmanuel Franck — Mathématiques Appliquées L3

**Informatique S6 - CC4**

On va considérer des algorithmes d'optimisations dans cet examen. On va comparer deux approches totalement différentes pour :

$$\max_x \mathsf{f}(\mathsf{x})$$

Dans un premier temps on va implémenter une classe générale, avant d'implémenter deux méthodes.

**Exercice 1** (**Classe optimisation** (3.5 points)).
*La classe optimisation sera un template de classe (paramètré par une classe* $\mathsf{T}$ *qui sera le type de nombre des fonctions* $\mathsf{f}(\mathsf{x})$ *qu'on maximise) qui a comme attribut :*

- *un entier (pour le nombre max d'itération,*
- *un pointeur de fonction qui en prend un "T" et renvoit un "T" (fonction a maximiser),*
- *un double "eps" (seuil pour la convergence)*
- *un objet de type T (qui contiendra la solution, le maximum)*

*Dans cette classe, on fera le constructeur par défaut, par copie, le destructeur, un constructeur avec paramètres qui remplit les 3 1ers attributs. Codez aussi les mutateurs/accesseurs. Codez une fonction virtuelle pure "solve" qui revoit un* $\mathsf{T}$. *Dans le future elle sera celle qui résoudra le problème de maximisation.*

**Exercice 2** (**Classe gradient** (4.5 points)).
*On va construire la classe gradient qui hérite de la classe optimisation. On le fera pour des réelles donc on propose que gradient hérite de optimisation<* `double` *>.*

**Question.** *La classe gradient contiendra comme attribut :*

- *un pointeur de fonction qui en prend un "double" et renvoit un "double" (fdérivée fonction a maximiser),*
- *un double (taux de descente de gradient)*

*Codez la classe avec le constructeur par défaut, par copie, le destructeur, un constructeur avec paramètres qui remplit les 3 1ers attributs de la classe mère et les deux de cette classe.* ***On devra utiliser ceux de la classe mère****.*

**Question** (descente de gradient). *Codez la méthode solve pour la classe gradien qui :*

- *tire aléatoirement un nombre selon une gaussienne centré en zero de variance 2,*
- *fait une boucle while jusqu a ce que le nombre d'itération max soit atteinte ou l'erreur inférieur a epsilon.*
- *calcul :*

$$\mathsf{x}_{k+1} = \mathsf{x}_k + \eta \mathsf{f}'(\mathsf{x})$$

  *avec l'erreur* $|\ \mathsf{x}_{k+1} - \mathsf{x}_k\ |$

**Exercice 3** (**Classe ES** (9.5 points))**.**
*L'algorithme ES est un algorithme stochastique qui deplace progressivement une gaussienne vers la zone ou le maximum se trouve. Pour cela*

- *on tire des points* $x$ *au hasard selon une gaussienne,*
- *on les classe selon la valeurs de* $f(x)$*,*
- *on prend les points associés au plus grand* $f(x)$*,*
- *on re-estime la gaussienne avec,*
- *on recommence,*

**Question.** *On va construire la classe " ES" qui hérite de la classe optimisation. On le fera pour des réelles donc on propose que "ES" hérite de optimisation*$<$ double $>$*.*

*La classe "Es" contiendra comme attribut :*

- *un vecteur "data" de double qui contiendra des points au hasard (utile pour bouger la gaussienne)*
- *un entier* $n$ *le nombre de points, un entier* $m$*,*
- *le nombre de point utile pour bouger la gaussienne,*

*Codez la classe avec le constructeur par défaut, par copie, le destructeur, un constructeur avec paramètres qui remplit les 3 1ers attributs de la classe mère et les deux de cette classe* $n$ *et* $m$*.* ***On devra utiliser ceux de la classe mère****.*

**Question** (Tri)**.** *Codez l'algo de tri. Cet algo prend un vecteur en entrée.*

- *on initialise* $i_{max} = 0$*,*
- *on balaye le tableau, si* $f(x_i) > f(i_{max})$*, on écrit* $i_m ax = i$ *,*
- *si* $i$ *est different de* $i_{max}$*, on échange* $x_i$ *et* $x_{i_{max}}$*. On peut le faire avec la fonction* $swap(data[i], data[i_{max}])$ *qui échange de place deux élements d'un vecteur.*

**Question** (Génération des points)**.** *Ecrire une fonction qui prend en entrée une moyenne* $\mu$ *et une variance* $\sigma^2$ *et qui remplit le vecteur "data" avec des points tirés selon une loi Gaussienne(*$\mu, \sigma^2$*).*

**Question** (Modification Gaussienne)**.** *Il s'agit d'une fonction qui prend en entrée* $\mu$ *et une variance* $\sigma^2$ *et va les moodifier ave les formules :*

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i$$

*et*

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu)^2$$

*On prend ici les* $m$ *premiers* $x_i$ *du vecteur "data" après le tri. Cela revient à prendre les 1ers* $m$ *points tel que* $f(x_i)$ *soit grand.*

**Question** (Algo ES)**.** *Codez la méthode solve pour la classe ES qui :*

- *fait une boucle while jusqu a ce que le nombre d'itération max soit atteinte ou l'erreur inférieur a epsilon.*
- *on génère des points avec la fonction associée,*

— *on sauvegare la dernière valeur de la moyenne,*
— *on modifie la Gaussienne en recalculant variance et moyenne,*
— *on calcul l'erreur*

**Exercice 4** (**test** (2.5 points))**.**
*Comparer les méthodes sur deux fonctions avec plusieurs run aléatoire :*

$$f(x) = -x^2 - 3x + 1$$

*et*

$$f(x) = -x^2 - 3x + 1 + 3e^{\frac{(x-0.5)^2}{2}}, \quad f^{'}(x) = -2x - 3 + 1.2(x - 0.5)e^{\frac{(x-0.5)^2}{2}}$$

*Ecrivez dans le code votre conclusion sur la comparaison des méthodes.*

Code pour gérer des données aléaatoire gaussienne :
"

```
default_random_engine re(time(0));
normal_distribution<double> distrib(0.0,2.0);
double x0=distrib(re);
```

"