

# PROGRAMMATION C

## TP NOTÉ DU 16 AVRIL 2012

DURÉE: 4 HEURES  
TOUS DOCUMENTS AUTORISÉS

LICENCE MATHS-INFO  
VALENTIN EMIYA

La notation prendra en compte la syntaxe du langage C, l'algorithmique choisie, la clarté du code, les commentaires insérés et les tests effectués dans le `main`. Le sujet repose sur un scénario unique autour de la modélisation de la température dans une pièce. Cependant, les questions sont en grande partie indépendantes.

### INTRODUCTION

**Avant de commencer.** Téléchargez et décompressez le fichier

[http://www.lif.univ-mrs.fr/~vemiya/teaching/2011\\_2012/L2C/examenTPC.zip](http://www.lif.univ-mrs.fr/~vemiya/teaching/2011_2012/L2C/examenTPC.zip) .

Vous complèterez le code comme l'indique l'énoncé.

**Problème.** Nous considérons une pièce donnée, dont nous connaissons la géométrie et la température sur les bords, c'est-à-dire au niveau des murs. Nous souhaitons trouver le meilleur endroit pour placer un radiateur afin que la température moyenne dans toute la pièce soit la plus élevée possible.

Pour cela, nous allons tester plusieurs positions possibles du radiateur et sélectionner la meilleure. Pour une position donnée, la démarche consiste à calculer la température en tout point de la pièce (cf. figure 1) puis la moyenne des températures des points de la pièce.

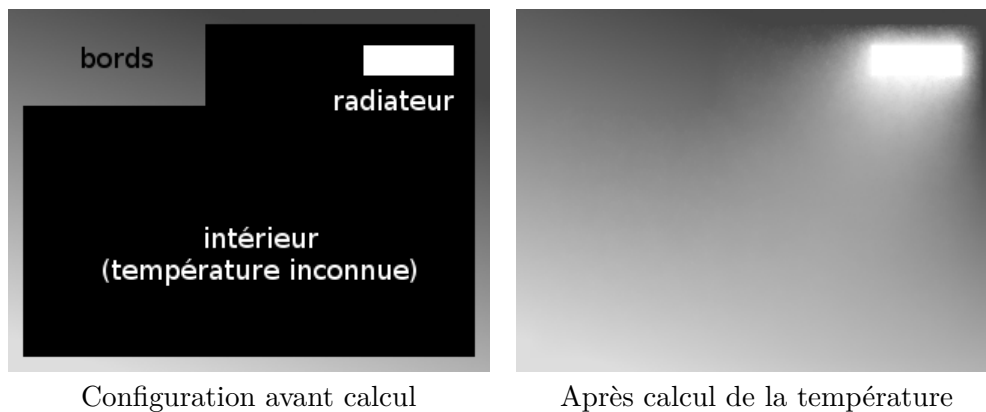


FIGURE 1. Exemple de calcul de la température dans une pièce, pour une position du radiateur.

**Modélisation de la température dans une pièce.** Nous modélisons la température dans une pièce par une image vue de haut de la pièce, en deux dimensions seulement, en négligeant la dimension de hauteur. L'image est en niveaux de gris entre 0 et 1. La valeur de chaque pixel correspond à la température au point considéré : plus la valeur est proche de 0, plus la température est basse. La valeur 0 est réservée pour désigner une température inconnue.

**Résolution du problème.** Nous décomposons le problème en plusieurs tâches élémentaires :

- dans l'exercice 1, calcul de la moyenne des températures dans une pièce où la température est connue partout ;
- dans l'exercice 2, estimation des températures inconnues ;
- dans l'exercice 3, test de plusieurs positions pour le radiateur et sélection de la meilleure.

**Envoi des fichiers.** Pour soumettre votre TP, respectez les instructions suivantes :

- compressez le répertoire racine du TP (celui qui contient le makefile) et son contenu et envoyez le fichier compressé par email ;
- le destinataire doit être `valentin.emiya@lif.univ-mrs.fr` ;
- le sujet doit être "Examen TP" ;
- le corps du mail doit contenir vos nom, prénom et No d'étudiant.

## 1. STATISTIQUES DE TEMPÉRATURE D'UNE PIÈCE

Lorsque la température d'une pièce est connue en tout point, c'est-à-dire lorsqu'aucun pixel n'est à 0, on souhaite calculer deux "statistiques" : la température moyenne et l'écart-type.

**Calcul de la température moyenne.** Écrivez une fonction

```
double moyenne_temperature(Image *p_image)
```

qui renvoie la température moyenne calculée sur l'ensemble des pixels de l'image pointée par `p_image`. Cette fonction n'affiche rien. Testez-là sur `test_stats.pgm`.

**Calcul de l'écart-type.** L'écart-type donne une valeur typique des écarts des températures par rapport à la moyenne. Il est donné par :

$$\sqrt{\frac{1}{LC} \sum_{i=0}^{L-1} \sum_{j=0}^{C-1} (\mathbf{A}_{ij} - m_{\mathbf{A}})^2}$$

où  $\mathbf{A}_{ij}$  est le pixel de la ligne  $i$  et la colonne  $j$  de l'image  $\mathbf{A}$ , de dimension  $L \times C$  et dont la moyenne est  $m_{\mathbf{A}}$ .

Écrivez une fonction

```
double ecart_type_temperature(Image *p_image)
```

qui renvoie l'écart-type calculé sur l'ensemble des pixels de l'image pointée par `p_image`. Cette fonction n'affiche rien. Testez-là sur `test_stats.pgm`.

**Nouveau type pour les statistiques de température.** On souhaite avoir une seule fonction à appeler pour obtenir la moyenne et l'écart-type.

Dans un fichier `temperature.h`, créez un type structuré `Stats_Temperature` qui contient deux champs permettant de stocker un réel chacun : une température moyenne et un écart-type.

**Fonction retournant les statistiques de température.** Écrivez une fonction

```
Stats_Temperature *moyenne_ecart_type(Image *p_image)
```

qui renvoie un pointeur vers une structure contenant la température moyenne et l'écart-type. Elle utilise les fonctions auxiliaires `moyenne_temperature` et `ecart_type_temperature`. Cette fonction n'affiche rien. Testez-là sur `test_stats.pgm`.

## 2. ESTIMATION DE LA TEMPÉRATURE DANS UNE PIÈCE

**Présentation de l’algorithme.** La répartition de la température dans la pièce dépend des sources de chaleur (murs plus ou moins froids, radiateur) et de leur position. Voici un algorithme permettant d’estimer la température en chaque point inconnu <sup>1</sup>.

L’algorithme repose sur la notion de lancer d’une marche aléatoire :

- (1) un lancer part d’un point donné dans la pièce ;
- (2) si la température au point courant est connue, on renvoie sa valeur et on s’arrête ;
- (3) sinon, on tire aléatoirement une direction au hasard – haut, bas, gauche, droite –, avec une chance sur quatre pour chaque direction, on déplace le point courant d’un pixel dans cette direction et on répète l’algorithme à partir de l’étape 2.

Ainsi, une marche aléatoire atteint une température connue (par exemple sur un mur ou sur le radiateur) au bout d’un certain temps. On ne sort jamais de l’espace de la pièce car on a supposé que l’on connaissait la température sur les bords. Un exemple de lancer est donné sur la figure 2.

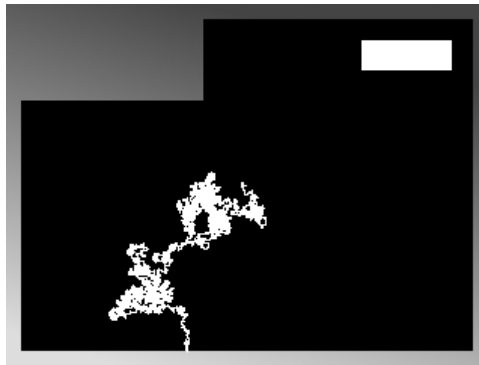


FIGURE 2. Exemple de marche aléatoire, en blanc, lancée depuis un point au centre de la pièce.

Finalement, pour chaque point de la pièce, l’estimation de la température est obtenue de la façon suivante :

- si la température du point est connue, ne rien faire ;
- sinon,
  - faire `nb_lancers` lancers à partir de ce point ;
  - calculer la valeur moyenne des `nb_lancers` résultats des lancers ;
  - la température estimée est égale à cette valeur moyenne.

Remarque : les nouvelles températures obtenues pourront servir de points d’arrêt des lancers dans le calcul des températures inconnues suivantes.

**Mise en œuvre d’une marche aléatoire.** Écrivez une fonction *non-réursive*

```
double lancer(Image *p_image, int i, int j)
```

qui effectue un lancer à partir de la position  $(i, j)$  dans l’image pointée par `p_image` et renvoie le résultat du lancer.

---

1. Il n’est bien-sûr pas nécessaire de connaître la théorie physique sous-jacente pour faire ce sujet. Si cela vous intéresse au-delà de ce TP, voici quelques informations supplémentaires : le résultat obtenu est la répartition de la température à l’équilibre, c’est-à-dire lorsqu’elle est stabilisée et ne varie plus au cours du temps ; le problème résolu est connu sous le nom de problème de Dirichlet ; l’algorithme utilisé utilise la notion de marche aléatoire également appelée mouvement brownien.

**Estimation de la température pour tous les points de la pièce.** Écrivez une fonction

```
void estime_temperature(Image *p_image, int nb_lancers)
```

qui calcule la température dans toute la pièce en utilisant `nb_lancers` lancers pour chaque pixel où la température est inconnue. Cette fonction modifie directement l'image initiale.

Dans un premier temps, vous pouvez faire tourner l'algorithme sur la petite image `test.pgm` (cf. figure 3).

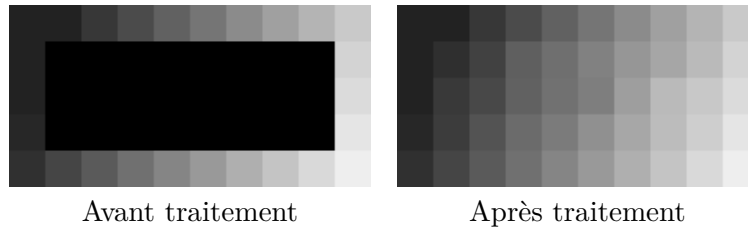


FIGURE 3. Traitement de la petite image de test.

### 3. POSITIONNEMENT OPTIMAL D'UN RADIATEUR DANS UNE PIÈCE

**Sélection de la meilleure position parmi plusieurs possibilités.** Dans le `main`, faites tourner l'algorithme sur les images `config1.pgm` à `config6.pgm` en fixant `nb_lancers=30`, affichez les statistiques de température pour chaque image puis sélectionnez la meilleure position. Correspond-elle également à la position où la variation de température (écart-type) est minimale ?

**Méthode automatique de recherche d'une position optimale.** Décrivez entre `/* ... */` une ou plusieurs stratégies pour rechercher de façon automatique la meilleure position du radiateur (ou une position quasi-optimale); puis écrivez une fonction

```
void place_radiateur(Image *p_image)
```

qui place un radiateur de température égale à 1 et de dimensions  $60 \times 20$  dans une pièce vide (`piece_vide.pgm`). Il est conseillé de créer des fonctions auxiliaires que `place_radiateur` appelle pour les différentes étapes.