

Processus stochastique : programme de simulation

Benjamin Dosse

1 Compilation

Le projet contient un fichier `Makefile`, permettant de compiler le projet à l'aide de l'utilitaire GNU `make`. Le code source du projet est situé dans le dossier `src`. Il est nécessaire de lier la bibliothèque `libm` lors de la compilation, puisque le projet utilise des fonctions déclarées dans le fichier `math.h`. La façon la plus simple de compiler le projet est alors d'exécuter la commande

```
$ make
```

1.1 Déboguage

Il est possible d'activer les symboles de déboguage nécessaires pour des utilitaires comme `gdb` ou `Valgrind`. Pour activer les symboles de déboguage, il suffit d'exécuter la commande

```
$ make debug
```

2 Exécution

La fonction `main` est de type `int`, ne prend aucun argument, et renvoie `-1` en cas d'échec à l'exécution, ou un nombre positif ou nul en cas de réussite à l'exécution.

Le fichier exécutable se situe dans le dossier `bin/` et se nomme `brownian_path.exe` (à des fins de portabilité). À l'emplacement d'exécution du programme, celui-ci écrit les fichiers suivants :

- `generated_data`
- `generated_data/data_bmt.csv`, contient une suite de nombres générés *via* la transformation de Box-Müller ;
- `generated_data/data_donsker.csv`, contient une trajectoire brownienne générée *via* l'algorithme de Donsker ;
- `generated_data/data_naive.csv`, contient une trajectoire brownienne générée *via* la définition même du mouvement brownien ;
- `generated_data/data_schauder.csv`, contient une trajectoire brownienne générée *via* l'algorithme de Schauder.

Si le dossier `generated_data/` n'existe pas, il sera créé avec les permissions POSIX 0700 : seul le propriétaire du dossier peut accéder à son contenu. Il peut lire les fichiers qui y sont stockés, et les exécuter.

Les données inscrites dans chaque fichier seront préalablement affichées sur la sortie standard.

Les données sont stockées dans des fichiers au format CSV afin d'en simplifier la manipulation. La première ligne de chacun de ces fichiers contient des chaînes de caractères.

3 Nettoyage

Invoker la commande

```
$ make clean
```

permet de supprimer les fichiers `.obj`. Cette étape est recommandée avant toute nouvelle compilation, notamment lors de l'utilisation des symboles de déboguage.

La commande

```
$ make mrproper
```

permet de supprimer l'ensemble des fichiers générés lors de la compilation.

4 Licence

L'ensemble des fichiers de ce projet sont placés sous licence MIT. Un fichier `LICENSE.txt` reprend les termes de cette licence. Une traduction non-officielle est disponible dans le fichier `LICENSE-fr-unofficial.txt`.