

Question 1 (5 points)

Un générateur de nombres quasi-aléatoire peut suivre le principe fractal de Van der Corput. Le principe est le suivant : on génère un nombre au milieu de chaque intervalle.

Pour $[0, 1[$ on génère 0.5 ce qui donne deux nouveaux intervalles

$[0, 0.5[$ qui génère 0.25

et

$[0.5, 1[$ qui génère 0.75

Nous avons ensuite 4 intervalles $[0, 0.25[$ $[0.25, 0.5[$ $[0.5, 0.75[$ $[0.75, 1[$ qui nous donnent 4 nouveaux nombres. 0.125, 0.375, 0.625, 0.875 et on continue avec les nouveaux intervalles...

Si on exclut 0 et 1, on a donc une suite du style 0.5, 0.25, 0.75, 0.125, 0.375, 0.625, 0.875...

- Coder un générateur de ce type en dimension 1. (en C)
- Les séquences de Halton consistent à généraliser ce principe dans plusieurs dimensions. Que proposeriez vous pour coder un générateur de ce type adapté pour la dimension 2 et 3.

Question 2 : (4 points)

Coder en C un calcul de π , par la méthode de Monte Carlo en faisant appel à une fonction de génération quasi-aléatoire en 2D supposée écrite. En prenant un rayon à 1, en centrant le cercle en 0, nous avons un disque de surface π . Un quart du disque se retrouve le carré $[0, 1] \times [0, 1]$. Tirer quasi-aléatoirement des points sur ce carré. Cumuler l'ensemble des points qui se retrouvent sur le disque ($x^2 + y^2 \leq \text{Rayon}^2$). La ratio : Nombre de points sur le disque / Nombre de points tirés converge vers π . (2 pts).

Quel est l'intérêt de ce type de générateur quasi-aléatoire par rapport à un générateur pseudo-aléatoire ? (1 pt.)

Question 3 : (5 points)

Un cluster de calcul est étudié. Vous disposez d'un historique sur 3 mois vous donnant le nombre de jobs prenant 1h, 2h, 3h... jusqu'à 48 h. Les jobs prenant plus de 48 heures sont aiguillés sur des super-calculateurs et ne tournent pas sur ce cluster

On peut considérer que les jobs à exécuter sur cette ferme sont de 4 types (classes). Les calculs durant moins de 7 heures, ceux durant entre 7 et 20 heures, ceux qui durent entre 25 et 47 heures. Et ceux qui durent 48 heures. Ces temps correspondent au temps de traitement par le cluster, on considère pour chaque classe nous avons temps de traitement suivant une loi normale (par exemple de moyenne 3,5h et d'écart type 1 h pour la première classe). On ne modélise pas chaque nœud du cluster. On dispose aussi du temps d'attente dans la file d'attente de ce cluster qui est vu comme un seul ordinateur. Préciser comment vous pourriez modéliser le temps d'attente, le type de données dont vous auriez besoin, comment vous les calculeriez ? De quelles fonctions avez vous besoin ? Un histogramme nous donne le nombre de jobs

soumis entre le 18/9 et le 27/11, chaque job pouvant être de l'une des classes considérées. Proposer des éléments de codage en C++. Quels outils (recherchés en TP) pourraient convenir pour ce type de modèle ?

