

Laboratoire 7

Intégration Monte Carlo

Avant de vous lancer dans ce Labo, relisez bien les sections 5.1, 5.2 et 5.4 des notes de cours. Rien de nouveau coté PLPLOT cette semaine.

7.1 Objectifs:

1. Vous familiariser avec la production de nombres aléatoire en C;
2. Apprendre à calculer des intégrales multidimensionnelles par Monte Carlo;
3. Apprendre à estimer l'erreur sur les résultats numériques de calculs de type Monte Carlo.
4. Apprendre à calculer le potentiel gravitationnel associé à des objets d'étendue finie

7.2 Rapport de Laboratoire

La rapport doit contenir des réponses, codes et résultats numériques à l'appui, aux questions posées aux §7.4 et 7.5.

7.3 Utiliser `rand()` pour générer des nombres aléatoires

Le langage C inclut un générateur de nombres aléatoires prenant la forme d'une fonction appelée `rand()`. Nous ne nous préoccupons pas trop de ce qui se passe à l'intérieur de cette boîte noire. Il suffit de savoir que cette fonction produit un entier (i.e, type `int`) prenant une valeur entre zéro et 2147483647 (sur un système monté en 64 bits comme ESIBAC). Un appel à `rand()` est l'équivalent de rouler un dé à autant de faces.

On peut produire une nombre aléatoire (r disons) de type `float` uniformément distribué dans l'intervalle unitaire $[0, 1]$ à l'aide d'une instruction du genre:

```
r = 1.*rand()/RAND_MAX ;
```

où `RAND_MAX= 2147483647` est une constante de type `int` prédéfinie incluse dans la librairie `<stdlib.h>`, et la variable `r` doit préalablement avoir été déclarée `float`, bien entendu. La pré-multiplication par `1.` est essentielle ici pour convertir l'entier retourné par `rand()` en type `float` **avant** la division par `RAND_MAX`, sinon c'est garanti que r vaudra pratiquement tout le temps zéro puisque `rand() ≤ RAND_MAX= 2147483647`.

La fonction `rand()` utilise un germe prédéfini valant 1; il est possible de changer ce germe à l'aide d'une autre fonction C prédéfinie appelée `srand`, contenue dans `<stdlib.h>`. Cette fonction-action accepte un germe (variable de type `int` fournie par l'utilisateur), et ne renvoie

aucune valeur numérique au code l'ayant appelé. On doit donc l'invoquer à l'aide d'une instruction du genre:

```
int mongerme=1234 ; /* Declaration et initialisation du germe */
...
srand(mongerme) ;
...
```

IMPORTANT: `srand()` ne doit être appelée qu'une fois avant le premier appel à `rand()`. Tout appel à `srand()` entre deux appels à `rand()` détruira les propriétés statistiques de votre séquence de nombres aléatoires. Vous n'avez pas vraiment à toucher au germe aux fins de ce Labo. En guise d'exemple d'utilisation de `rand()` et `srand()`, le petit bout de code suivant produit 10 nombres aléatoires entre zéro et un, et les renvoie à l'écran:

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
/* Declarations ----- */
float r ;
int i, n=10 ;
int mongerme=1234 ;
/* Executable ----- */
srand(mongerme) ;
for (i=0 ; i<n ; i++) {
    r=1.*rand()/RAND_MAX ;
    printf ("Nombre aleatoire %d = %f\n",i,r) ;
}
}
```

Tapez ce petit code, compilez le et assurez vous que tout se passe comme prévu.

7.4 Calcul du volume d'une hypersphère en D -dimensions

La première partie de ce labo vise à vous faire calculer le volume d'une hypersphère à l'aide de l'approche Monte Carlo. La logique est la même que celle expliquée au début du chapitre 5 des notes de cours dans le cas de l'exemple de l'étang. On considère un hypercube de côté $L = 2$ dans un espace 5-dimensionnel (v, w, x, y, z) . Ce cube est centré sur l'origine $(0, 0, 0, 0, 0)$, et s'étend donc de -1 à 1 dans chacun des cinq axes de coordonnées (cartésiennes) de l'espace 5D. Le volume de l'hypercube est donc $2 \times 2 \times 2 \times 2 \times 2 = 32!$ Une hypersphère de rayon 1 centrée à l'origine du système de coordonnées sera donc complètement contenue dans l'hypercube, le touchant seulement au centre de ses faces du cube (combien de faces a un hypercube en cinq dimensions spatiales ...?).

Il s'agit maintenant de générer des points (v, w, x, y, z) uniformément distribués dans notre espace 5-dimensionnel; chaque coordonnée, devant être distribuée entre -1 et $+1$, sera donc produite par un appel à un générateur de nombre aléatoires distribué uniformément dans l'intervalle $[0, 1]$; par exemple,

```
float u, v, x, y, z ;
...
u=-1.+2.*rand()/RAND_MAX ;
v=-1.+2.*rand()/RAND_MAX ;
x=-1.+2.*rand()/RAND_MAX ;
y=-1.+2.*rand()/RAND_MAX ;
z=-1.+2.*rand()/RAND_MAX ;
```

Avant d'aller plus loin, assurez vous de bien comprendre pourquoi, si $0 \leq 1 \cdot \text{rand()}/\text{RAND_MAX} \leq 1$, alors $-1 \leq -1 + 2 \cdot \text{rand()}/\text{RAND_MAX} \leq 1$. Une fois ça compris, pour chacun de ces points, on calcule la distance d à l'origine; même dans un espace à cinq dimensions, cette distance est néanmoins donnée par:

$$d = \sqrt{v^2 + w^2 + x^2 + y^2 + z^2} . \quad (7.1)$$

Si et seulement si $d \leq R$, alors on ajoute 1 à une variable compteur, (préalablement initialisée à zéro); sinon on passe au point aléatoire suivant. Un canevas de code C pour cet algorithme pourrait avoir l'allure suivante:

```
#include <stdio.h>
#include <stdlib.h>
#define N 1000                                /* Nombre de points 5D generes */
int main(void)
{
/* Declarations ----- */
float d, u, v, x, y, z, som ;
int i ;
/* Executable ----- */
som=0. ;                                     /* initialisation de la variable-compteur */
for (i=0 ; i<N ; i++) {
    u= ...                                  /* generer un point 5D */
    d = ...                                 /* calculer la distance a l'origine */
    if (d < 1.) { som += 1. ; } /* incrementation de la variable compteur */
}
}
```

Remarquez qu'il est préférable ici de définir la variable `som` comme un `float`, au cas où `N` se retrouverait très grand. Il ne reste plus qu'à diviser `som` par `N` et multiplier par le volume de l'hypercube contenant la sphère pour obtenir un estimé Monte Carlo du volume de l'hypersphère. Je vous laisse imaginer de quoi devrait avoir l'air l'instruction en C effectuant ce dernier petit calcul...

La procédure pour le reste de cette partie du labo est maintenant la suivante:

1. Écrivez un code C utilisant l'approche Monte Carlo décrite ci-dessus pour calculer le volume d'une sphère en trois dimension spatiales;
2. Calculez le volume par Monte Carlo pour $N= 10, 100, 1000, 10000$, et 100000 ;
3. Estimez combien de points (x, y, z) vous devrez utiliser pour arriver au résultat connu ($V = 4\pi R^3/3$) à mieux de 1%. Surprise, si tout s'est bien passé vous venez de compléter l'étape de validation pour ce labo!
4. Répétez maintenant le calcul pour une hypersphère en cinq dimensions spatiales, toujours pour $N= 10, 100, 1000, 10000$, et 100000 . Examinez comment la valeur calculée du volume varie en fonction de N .
5. Basé sur les résultats de votre calcul précédent en 3D, estimez la valeur de N requise pour arriver à une valeur du volume précise à mieux de 1%.
6. Pouvez vous "deviner" la valeur exacte du volume de l'hypersphère en 5D, en terme d'une fraction entière de π ?

7.5 Calcul d'une intégrale multidimensionnelle par Monte Carlo

Nous allons passer à un problème plus “physique”, soit le calcul du potentiel gravitationnel à l'extérieur d'un objet d'étendue finie. Pour une masse ponctuelle de grandeur M , vous savez déjà que le potentiel gravitationnel est donné par

$$\Phi(r) = -\frac{GM}{r}, \quad (7.2)$$

où $G = 6.67 \times 10^{-11} \text{ N m}^2 \text{ kg}^{-2}$ est la constante gravitationnelle de Sir Isaac. Pour une masse d'étendue finie, la stratégie consiste à l'approximer par un très grand nombre de petits éléments de masse considérés ponctuels, en faire la somme des potentiels correspondants. C'est le **principe de superposition**, auquel satisfait le champ gravitationnel (ainsi que les champs électrique et magnétique), qui permet d'aborder le problème de cette façon. Nous allons de plus supposer que sa densité (ρ , unités: kg m^{-3}) de notre objet est constante, ce qui implique qu'un petit élément de volume dV a une masse donnée par

$$dm = \rho dV.$$

Dénotons la position de cet élément de masse par (x', y', z') , et exprimons son volume comme $dV = dx' \times dy' \times dz'$; sa contribution au potentiel gravitationnel mesuré à un point (x_0, y_0, z_0) sera alors donnée par

$$d\Phi(x_0, y_0, z_0) = -\frac{G\rho dx' dy' dz'}{\sqrt{(x_0 - x')^2 + (y_0 - y')^2 + (z_0 - z')^2}}. \quad (7.3)$$

où le dénominateur de l'intégrand correspond à la distance entre l'élément de masse, situé à (x', y', z') , et le point (x_0, y_0, z_0) où nous cherchons à calculer le potentiel gravitationnel. La Figure 7.1 illustre la géométrie du problème et sa relation à la notation, dans le cas d'un objet de forme arbitrairement patatoïdale. Le potentiel gravitationnel total au point (x_0, y_0, z_0) , associée à la distribution de masse dans son ensemble, est obtenu en intégrant cette expression sur le volume de l'objet; en coordonnées cartésiennes:

$$\begin{aligned} \Phi(x_0, y_0, z_0) &= \iiint_V d\Phi(x_0, y_0, z_0) \\ &= -G\rho \iiint \frac{dx' dy' dz'}{\sqrt{(x_0 - x')^2 + (y_0 - y')^2 + (z_0 - z')^2}}. \end{aligned} \quad (7.4)$$

Notez bien, car c'est **très** important pour tout ce qui suit, que l'intégrale se fait par rapport aux variables “prime”, qui mesurent la position de chaque élément de masse contribuant à l'intégrale; la position (x_0, y_0, z_0) à laquelle le potentiel est calculé est ici fixe, du point de vue du calcul de cette intégrale.

Pour un objet de forme compliquée, comme sur la Fig. 7.1, vous pouvez facilement imaginer que la spécification des bornes d'intégration en x , y et z ne serait pas exactement de la tarte. Afin d'éviter ce genre de complications, nous allons nous intéresser ici à un monolithe (noir) de forme rectangulaire parallélépipédique, ayant un rapport d'aspect $x : y : z = 1 : 4 : 9$.

La dernière partie de ce Labo consiste à vous faire calculer le potentiel gravitationnel à l'extérieur de ce mythique monolithe, par intégration Monte Carlo, et de vérifier que le potentiel gravitationnel tombe bien en $1/r$ à grande distance du monolithe. Si l'on fait coïncider son centre avec l'origine de notre système de coordonnées cartésienne, le potentiel gravitationnel à (x_0, y_0, z_0) , donné par l'éq. (7.4), devient:

$$\Phi(x_0, y_0, z_0) = -G\rho \int_{-4.5}^{4.5} \int_{-2}^2 \int_{-0.5}^{0.5} \frac{dx' dy' dz'}{\sqrt{(x_0 - x')^2 + (y_0 - y')^2 + (z_0 - z')^2}}. \quad (7.5)$$

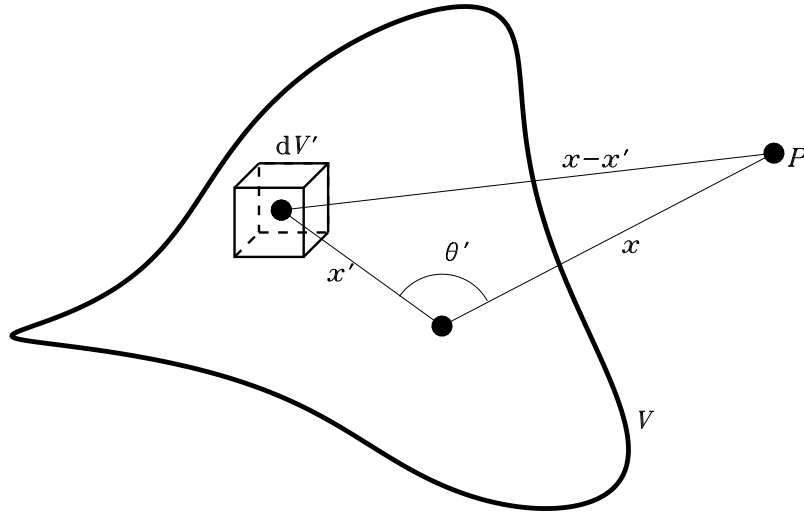


Figure 7.1: Géométrie et notation pour la formulation du calcul du potentiel gravitationnel associé à un objet ici de forme arbitrairement patatoïdale, occupant un volume V de l'espace. Le potentiel gravitationnel est calculé au point $P \equiv (x_0, y_0, z_0)$.

En définissant une fonction f comme suit:

$$f(x_0, y_0, z_0, x', y', z') = \frac{1}{\sqrt{(x_0 - x')^2 + (y_0 - y')^2 + (z_0 - z')^2}}, \quad (7.6)$$

on peut ramener l'éq. (7.5) à notre forme classique pour une intégrale définie:

$$\Phi(x_0, y_0, z_0) = -G\rho \int_{-4.5}^{4.5} \int_{-2}^2 \int_{-0.5}^{0.5} f(x_0, y_0, z_0, x', y', z') dx' dy' dz' \quad (7.7)$$

La fonction f mesure simplement l'inverse de la distance entre le point (x_0, y_0, z_0) où le potentiel est calculé, et le point (x', y', z') du petit élément de masse contribuant au potentiel.

Il s'agit maintenant pour vous de calculer cette intégrale par la méthode de Monte Carlo. Vous vous rappelez certainement que l'idée de l'intégration Monte Carlo est d'échantillonner l'intégrand à un grand nombre (N) de positions choisies *aléatoirement* à l'intérieur du monolithe, et d'estimer le potentiel comme la moyenne de ces évaluations. L'équation (7.7) est donc remplacée par:

$$\Phi(x_0, y_0, z_0) = -G\rho \frac{(L_x L_y L_z)}{N} \sum_{r=1}^N f(x_0, y_0, z_0, x_r, y_r, z_r), \quad (7.8)$$

où la fonction f est la même qu'auparavant, $L_x = 1$, $L_y = 4$ et $L_z = 9$ sont les mesures du monolithe, et les x_r , y_r et z_r sont des nombres aléatoires tirés de distributions uniformes couvrant ici les intervalles:

$$x_r \in [-0.5, 0.5], \quad y_r \in [-2, 2], \quad z_r \in [-4.5, 4.5]. \quad (7.9)$$

Votre première tâche est maintenant de créer une fonction (appelée `potgravmc`, tiens), qui fait ce calcul Monte Carlo. Cette nouvelle fonction doit être appelée avec trois arguments, soit les valeurs x_0, y_0, z_0 auxquelles on veut calculer le potentiel. Vous devriez utiliser la même structure globale de code qu'au laboratoire 3: un programme principal (`main`) qui fait appel à la routine d'intégration `potgravmc`, elle même faisant appel à une fonction `integrant` qui calcule l'éq. (7.6). Remarquez que, contrairement à ce qui aurait du être codé pour une version utilisant la méthode du trapèze en 3D, ici il n'y a pas de maillage en coordonnées "prime" à calculer, pas de triples boucles sur ces coordonnées, plus d'éléments de volume; il n'y a plus que N évaluations de l'intégrand (essentiellement la fonction f ci-dessus), donc une seule boucle se répétant N fois. Et pas de panique, tous les éléments requis ont déjà été couverts dans la première partie de ce Labo!

Une fois la fonction écrite, les étapes à exécuter sont les suivantes:

1. Évaluez d'abord le potentiel gravitationnel le long d'une diagonale $x = y = z$ partant du centre du monolithe, plus spécifiquement aux points $(x_0, y_0, z_0) = (1, 1, 1), (3, 3, 3), (10, 10, 10), (30, 30, 30), (100, 100, 100), (300, 300, 300)$, et $(1000, 1000, 1000)$. Pour chacun de ces points, effectuez **dix** calculs distincts, chacun utilisant un germe différent pour le générateur de nombres aléatoires.
2. Pour chaque groupe de 10 valeurs Φ_k associées à chaque (x_0, y_0, z_0) , calculer la valeur moyenne du potentiel gravitationnel pour vos dix essais, ainsi que la variance:

$$\bar{\Phi} = \frac{1}{10} \sum_{k=1}^{10} \Phi_k, \quad \sigma = \left(\frac{1}{10} \sum_{k=1}^{10} (\Phi_k - \bar{\Phi})^2 \right)^{1/2}.$$

3. Portez en graphique log-log $\bar{\Phi}$ versus $r_0 = \sqrt{x_0^2 + y_0^2 + z_0^2}$, soit la distance au centre du monolithe. Utilisez `pljoin` pour tracer, à chaque valeur de r_0 , une "barre d'erreur" couvrant l'intervalle $[\bar{\Phi} - \sigma, \bar{\Phi} + \sigma]$.
4. La quantité $\sigma/\bar{\Phi}$ est une mesure de la précision relative de votre calcul Monte Carlo; ce calcul est-il plus précis près ou loin du monolithe? Pourquoi?
5. À partir de quelle distance du monolithe votre potentiel gravitationnel chute-t-il en $1/r$, à l'intérieur des barres d'erreur? Pourquoi?

Finalement, avant de quitter le lab vous devriez vous convaincre que votre calcul du volume de l'hypersphère revenait essentiellement au calcul d'une intégrale du genre:

$$I = \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} g(u, v, x, y, z) du dv dx dy dz, \quad (7.10)$$

où l'intégrand $g(u, v, x, y, z)$ vaut un dans l'hypersphère et zéro à l'extérieur. Une fois convaincu(e).., le labo 7 est terminé.