

(Abgabe: bis zum *17. Mai 2017, 16:00 Uhr*)

Quellcode, Filme und Bilder bitte in "/home/comphys/comphys_ss17_Abgabe/" im Cip Pool ablegen. Der schriftliche Teil kann entweder als Pdf beigelegt oder im Postfach von Prof. Rieger abgegeben werden. Unter "/home/comphys/comphys_ss17/exercises_supplemental/" finden sie die jeweilig Dateien, die für die Bearbeitung hilfreich sind.

1. [2,5 Punkte] Zufallszahlengeneratoren

Nutzen Sie den Zufallszahlengenerator „std::mt19937“ zum Lösen nachstehender Aufgaben:

- Schreiben Sie eine C++-Funktion "double gleichminus1bis1(int seed), welche gleichverteilte Zufallszahlen über dem Intervall $(-1; 1)$ liefert. Prüfen Sie Ihren Algorithmus, indem Sie 10^9 Samples generieren und den Mittelwert sowie die Varianz mit den analytisch berechenbaren Werten vergleichen. Erzeugen Sie außerdem ein Dichtehistogramm Ihrer Samples mit geeigneter Schrittweite und zeichnen Sie zum Vergleich die Dichtefunktion darüber.
- Schreiben Sie eine C++-Funktion "double linearverteilt(int seed, double a)", welche Zufallszahlen auf dem Intervall $(0; a)$ mit der Wahrscheinlichkeitsdichte $p(x) = \frac{2}{a^2}x$ liefert. Prüfen Sie Ihren Algorithmus mit $a = 2$, indem Sie 10^9 Samples generieren und den Mittelwert sowie die Varianz mit den analytisch berechenbaren Werten vergleichen. Erzeugen Sie außerdem ein Dichtehistogramm Ihrer Samples mit geeigneter Schrittweite und zeichnen Sie zum Vergleich die Dichtefunktion darüber.
- Schreiben Sie eine C++-Funktion "double gaussverteilt(int seed, double mu, double sigma)", welche Zufallszahlen in \mathbb{R} mit der Wahrscheinlichkeitsdichte $p(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$ liefert. Prüfen Sie Ihren Algorithmus mit $\mu = 5$ und $\sigma = 2$, indem Sie 10^9 Samples generieren und den Mittelwert sowie die Varianz mit den analytisch berechenbaren Werten vergleichen. Erzeugen Sie außerdem ein Dichtehistogramm Ihrer Samples mit geeigneter Schrittweite und zeichnen Sie zum Vergleich die Dichtefunktion darüber.
- Schreiben Sie eine C++-Funktion "void gleichverteiltAufKugeloberflaeche(int seed, double & x, double & y, double & z)", welche gleichverteilte Zufallspunkte (x,y,z) auf der Oberfläche der Einheitskugel liefert.

2. [2,5 Punkte] direct-Pi

Schreiben Sie eine Funktion "double direct-Pi(unsigned long long int nr_samples)", welche mit den in Vorlesung und Übung vorgestellten Methoden π schätzt. Rufen Sie diese Funktion für die Werte $\text{nr_samples}=10^2, 10^4, 10^6, 10^8, 10^9$ auf und vergleichen Sie jeweils mit dem exakten Wert. Geben Sie mögliche Gründe für das Stagnieren der Konvergenz an.

3. [2,5 Punkte] Markov-Pi

- Implementieren Sie den in der Vorlesung vorgestellten Algorithmus "double Markov-Pi(unsigned long long int nr_samples)".
- Vergleichen Sie die Konvergenzgeschwindigkeit sowie die Approximationsgenauigkeit mit "double direct-Pi(unsigned long long int nr_samples)", indem Sie für die Samplezahlen aus Aufgabe 2 die Funktion evaluieren.

4. [2,5 Punkte] Volumenberechnung à la direct-Pi

Das Verfahren aus Aufgabe 2 kann auch verwendet werden, um Volumina zu berechnen. Betrachten Sie dazu die Funktion $f(x, y) = x^2 \cdot \cos(y) + y^2 \cdot \sin(x + y) + 2$ auf der Menge $(-1; 1) \times (-1; 1)$. Schreiben Sie eine Funktion, welche das Volumen zwischen $(-1; 1) \times (-1; 1)$ und der darüberliegenden Funktionsfläche schätzt.

Anmerkung: Selbstverständlich gibt es effektivere Methoden, Flächen und Volumina zu berechnen. Wird die Dimension des betrachteten Problems jedoch sehr groß, so liefern diese Verfahren auch dort noch verlässliche Resultate, wo herkömmliche "Maßbestimmungsmethoden" bereits versagen.