Computational Physics SoSe 2022
10. Übungsblatt

Ausgabe: 16.06.2022 Prof. H. Jelger Risselada
Abgabe: 23.06.2022 20 Uhr

Aufgabe 0: Verständnisfragen 0 Punkte

1) Welchen Vorteil bieten Pseudo-Zufallszahlen? Wann ist es sinnvoll, „echte“ Zufallszahlen zu verwenden?

2) Welche Anforderungen werden an gute Zufallszahlen gestellt? Wie können diese getestet werden?

3) Welche Verfahren kennen Sie, um Zufallszahlen zu generieren? Welche Verteilungen können Sie damit erzeugen?

Aufgabe 1: Linear kongruente Generatoren 10 Punkte

Zufallszahlengeneratoren (RNGs) haben viele verschiedene Anwendungen. Im Falle der Physik werden sie beispielsweise häufig verwendet, um verschiedene Startwerte für Algorithmen oder potentielle neue Schritte einer Monte-Carlo-Simulation zu ziehen. Solche Zufallszahlengeneratoren generieren jedoch keinen wahren Zufall, sondern Zahlen, welche ab einer gewissen Anzahl von Ziehungen einem bestimmten Muster folgen. Je mehr Ziehungen es braucht, bis sich ein solches Muster zeigt, desto besser ist der Zufallszahlgenerator. Vor allem für Algorithmen, welche eine große Anzahl an Zufallszahlen benötigen, ist es daher sinnvoll, gute Zufallsgeneratoren zu verwenden. In dieser Aufgabe wird daher die Güte zweier RNGs miteinander verglichen.

Generieren Sie Pseudo-Zufallszahlen, indem Sie einen linear kongruenten Generator

$$r_{n+1} = (ar_n + c) \bmod m \tag{1}$$

selbst implementieren.

a) Schreiben Sie ein Programm, um die ersten N Glieder der Integer-Folge r_n abhängig von den 5 Parametern r_0, a, c und m zu generieren. Verwenden Sie hierbei 64-Bit-Integer. Teilen Sie r_{n+1} durch m, um einen floating point-Generator für Zufallszahlen in $[0, 1[$ zu bekommen.

b) Untersuchen Sie für die zwei Parametersätze

   (i) $r_0 = 1234$, $a = 20$, $c = 120$, $m = 6075$,

   (ii) $r_0 = 123456789$, $a = 65539$, $c = 0$, $m = 2^{31} = 2147483648$
   (RANDU Generator von IBM),

   Ihren floating-point-Generator auf Gleichverteilung, indem Sie für $N = 10^5$ Werte ein Histogramm erstellen. Teilen Sie hierfür das Intervall $[0, 1[$ in 20 Bins auf. Vergleichen Sie das Ergebnis Ihrer Generatoren mit dem Mersenne Twister 19937. Was stellen Sie fest?
   Hinweis: Verwenden Sie `mt19937` und `uniform_real_distribution` aus der Standardbibliothek bzw. `MT19937` aus `numpy.random`.

c) Erzeugen Sie mithilfe des Neumannschen Rückweisungsverfahrens Zufallszahlen im Intervall $[0, 1[$, die der Verteilung $p(x) = n \sin^4(\pi x)$ folgen. Wie müssen Sie n wählen, damit $p(x)$ im angegebenen Intervall normiert ist? Verwenden Sie zum Erzeugen der gleichverteilten Zufallszahlpaare (x, y) die drei Generatoren aus Aufgabenteil b). Histogrammieren Sie die Zufallszahlen für jeden der drei Generatoren und vergleichen Sie diese mit $p(x)$. Wie unterscheiden sich die Ergebnisse?

d) Testen Sie die drei floating-point-Generatoren aus b) nun auf Korrelationen, indem Sie Paare aus aufeinanderfolgenden Punkten in einem zweidimensionalen Quadrat $[0,1]^2$ auftragen. Benutzen Sie hierzu $N = 10^5$ Werte. Was sehen Sie? Können Sie hiermit die Ergebnisse aus c) erklären?

e) Untersuchen Sie die Korrelationen der drei floating point-Generatoren in höherer Ordnung, indem Sie drei aufeinanderfolgende Zahlen als Punkte in einem dreidimensionalen Würfel $[0,1]^3$ auftragen. Benutzen Sie hierzu wieder $N = 10^5$ Werte. Welche der getesteten Zufallsgeneratoren sind für physikalische Simulationen geeignet und welche nicht?
Hinweis: Betrachten Sie die Plots aus verschiedenen Winkeln, um mögliche Strukturen zu erkennen.

Aufgabe 2: Beliebige Verteilungen — 10 Punkte

Viele Zufallszahlengeneratoren (RNGs) sind von ihrer Natur aus zunächst gleich- oder normalverteilt. Es gibt jedoch auch Anwendungen, für welche eine andere Verteilung von Zufallszahlen benötigt wird. Häufig ist es einfacher, eine solche Verteilung aus einer anderen Verteilung, für welche man bereits einen RNG hat, zu erzeugen, anstatt einen vollständig neuen RNG für diese Verteilung zu suchen. Daher beschäftigt sich diese Aufgabe damit, wie sich verschiedene Verteilungen ineinander umformen lassen.

Ein Zufallszahgenerator, der gleichverteilte Zahlen zwischen 0 und 1 erzeugt, kann auch eingesetzt werden, um beliebige Verteilungen zu erzeugen. Verwenden Sie für die folgenden Aufgabenteile den Mersenne Twister 19937 (siehe Aufgabe 1b)). Erzeugen Sie in jeder Teilaufgabe jeweils ein Histogramm mit 10^5 Zufallszahlen und einem Plot der analytischen Verteilung.

a) Implementieren Sie den Box-Muller-Algorithmus, um eine Gauß-Verteilung ($\mu = 0$ und $\sigma = 1$) zu erzeugen.

b) Verwenden Sie den zentralen Grenzwertsatz, um eine Gauß-Verteilung zu erzeugen. Bilden Sie dafür die Summe von 12 gleichverteilten Zufallszahlen aus $[0,1[$. Wie bekommt man eine Verteilung mit Mittelwert $\mu = 0$ und Standardabweichung $\sigma = 1$? Welche Nachteile hat diese Methode, z.B. in Korrektheit und Effizienz?

c) Verwenden Sie das von Neumannsche Rückweisungsverfahren, um eine Gauß-Verteilung ($\mu = 0$ und $\sigma = 1$) in den Grenzen $-\infty$ bis ∞ zu erzeugen. Generieren Sie hierzu gleichverteilte Zufallszahlen in der Fläche $A = \{(x,y) \mid x \in \mathbb{R} \text{ und } 0 < y < \alpha \exp(-|x|)\}$. Für welches α ist das Vorgehen am effizientesten? Was passiert, wenn Sie α größer oder kleiner wählen?
Hinweis: Verwenden Sie `exponential_distribution` aus der Standardbibliothek, um exponentialverteilte Zufallszahlen zu erzeugen.

d) Verwenden Sie die Transformationsmethode, um die Cauchy-Verteilung

$$p(x) = \frac{1}{\pi} \frac{1}{1+x^2} \tag{2}$$

in den Grenzen $-\infty$ bis ∞ aus einer Gleichverteilung im Intervall $[0,1[$ zu erzeugen. Beschreiben Sie Ihr Vorgehen.

Aufgabe 3: Monte-Carlo-Integration (Bonusaufgabe) — 10 Punkte

Monte-Carlo-Algorithmen werden in der Physik häufig verwendet, beispielsweise um den Gleichgewichtszustand des Ising-Modells für verschiedene Temperaturen zu ermitteln. In dieser Aufgabe wird mit der Monte-Carlo-Integration, vorbereitend auf die kommende Aufgabe zur Monte-Carlo-Simulation, ein vergleichsweise einfacher Monte-Carlo-Algorithmus implementiert.

Wir berechnen Integrale durch Monte-Carlo-Integration, d.h. indem wir die Integrale als Mittelwert bezüglich einer Zufallsvariablen x mit Verteilung $p(x)$ umschreiben,

$$\langle f \rangle = \int p(x) f(x)\, dx = \frac{1}{N} \int_{i=1}^{N} f(x_i) \tag{3}$$

und das Integral durch N-maliges Ziehen einer Zufallszahl aus der Verteilung $p(x)$ berechnen. Wir wollen hier $f(x)$ so wählen, dass wir Standard-Zufallszahlengeneratoren für $p(x)$ verwenden können, die Gleichverteilungen $p(x) = 1$ im Intervall $x \in [0, 1[$ generieren.

a) Das klassische Beispiel ist die Berechnung von π:

$$\int_{|\boldsymbol{r}|<1} d^2\boldsymbol{r} \tag{4}$$

Ziehen Sie N in $[0, 1[^2$ gleichverteilte Zufallszahlenpaare (x_i, y_i) und zählen Sie, wie oft $x_i^2 + y_i^2 < 1$ gilt. Wie berechnet sich daraus das Integral (4) und welcher Wahl von $f(x)$ entspricht dieses Vorgehen?

b) Sie kennen das in a) gesuchte Ergebnis exakt. Berechnen Sie den Fehler als Funktion von N für $N = 10^k$ mit $k = 1, \ldots, 6$ und plotten Sie Ihr Ergebnis doppelt-logarithmisch. Welche N-Abhängigkeit sollten Sie finden?
Berechnen Sie 1000 mal das Integral (4) mit $N = 1000$ und plotten Sie ein Histogramm der Verteilung der Ergebnisse. Wie sollte die Verteilung aussehen?

c) Schreiben Sie eine Monte-Carlo-Integrationsroutine zur Berechnung des Flächeninhalts einer Ellipse

$$\int_{(x/a)^2+(y/b)^2<1} dxdy \tag{5}$$

als Funktion der Parameter a und b.

d) Erweitern Sie die Routine auf die Monte-Carlo-Integration eines Integrals

$$\int_{(x/a)^2+(y/b)^2<1} f(x, y)\, dxdy \tag{6}$$

einer Funktion $f(x, y)$ über eine Ellipse. Berechnen Sie damit

$$\int_{(x/a)^2+(y/b)^2<1} e^{-x^2}\, dxdy \tag{7}$$

(Kontrolle: 2.993 . . .).