

Übungen zur Computerorientierten Physik

5 Random walk mit Zurücksetzen

Ein Random Walk, hier in einer Dimension, ist der Zufallsweg $x(t)$ eines Teilchens als Funktion der diskreten Zeit $t = 0, 1, 2, \dots$. Hier betrachten wir den einfachsten Falls, dass das Teilchen bei $x(0) = 0$ startet und in jedem Zeitschritt mit Wahrscheinlichkeit $1/2$ nach rechts, also $x(t+1) = x(t) + 1$, und mit Wahrscheinlichkeit $1/2$ nach links geht, also $x(t+1) = x(t) - 1$. Es ist bekannt, dass das Teilchen bei Mittelung über viele Läufe sich im Mittel typischerweise von der 0 entfernt, d.h. $\langle x^2(t) \rangle = t$. Das Teilchen ist genauso häufig bei $x > 0$ wie bei $x < 0$, deshalb mittelt sich das zu $\langle x(t) \rangle = 0$ weg.

Ziel ist die Berechnung der mittleren (mean) *first-passage time* $T(x_0)$ ($x_0 > 0$), also die Zeit die es dauert, bis das Teilchen erstmals den Punkt x_0 passiert, d.h. $x(t) \geq x_0$ erstmals für $t = T(x_0)$.

Wie in einer aufsehererregenen Arbeit¹ 2011 von Evans und Majumdar gezeigt wurde, kann es die Zeit $T(x_0)$ um den Punkt zu “finden” reduzieren, wenn man hin und wieder das Teilchen an den Nullpunkt zurück setzt, also $x(t) = 0$, egal wo es gerade ist. Das soll in jedem Zeitschritt t mit der *resetting Wahrscheinlichkeit* p_r passieren.

- Erklären Sie warum es sinnvoll ist, das Teilchen manchmal zurück zu setzen, also $p_r > 0$? Was passiert wenn p_r groß wird?
- Laden Sie sich das unvollständige Programm `reset_walk_fragment.c` vom StudIP und speichern es (z.B.) unter `reset_walk.c` ab. Es kann einfach compiliert werden:

```
cc -o reset_walk reset_walk.c -lm -g -Wall
```

- Schauen Sie sich an, was das Programm bisher macht, also das Hauptprogramm.
- Kompletieren Sie die Funktion `reset_step()`, die das Teilchen gemäß der oben angegebenen Regeln bewegt. Testen Sie die Funktion geeignet, am besten mit `gdb`.

(6 Punkte)

- Führen Sie Simulationen für $x_0 = 3$, $x_0 = 5$ und $x_0 = 8$ durch (ggf. letztlich auch bei Mittelung über 10000 statt der voreingestellten 1000 Läufe), für geeignet gewählte Bereiche von p_r . Sie können um die Simulation bequemer zu machen, die Hauptfunktion `main()` um eine Schleife erweitern, in der `p_reset` geeignet verändert wird. (2 Punkte)

¹Martin R. Evans and Satya N. Majumdar, “Diffusion with Stochastic Resetting”, Phys. Rev. Lett. **106**, 160601 (2011)

- Stellen Sie das Ergebnis dar, z.B. mit `gnuplot`.

Fitten Sie für kleine p_r (z.B. in $[0, 0.1]$) dabei an die unterschiedlichen Daten die Funktion $f(p_r) = a[e^{b\sqrt{p_r}} - 1]/p_r$ mit Parametern a und b . (2 Punkte)