

Rechnernutzung in der Physik

Institut für Experimentelle Teilchenphysik
Institut für Theoretische Teilchenphysik
Interfakultatives Institut für Anwendungen der Informatik

Prof. G. Quast, Prof. M. Steinhauser

Dr. A. Mildenerger, Dipl.-Phys. Jens Hoff, Dr. M. Zeise

WS2011/12 – Blatt 05

<http://comp.physik.kit.edu>

Prog.: Di, 29.11.2011 / Ausarb.: Fr, 02.12.2011

Aufgabe 11: Einführung zum Programmieren in Mathematica

freiwillig

Auf der Vorlesungshomepage finden Sie die Datei `Intro_Programmieren.nb`. In diesem Notebook werden Programmierkonstrukte und fortgeschrittene Aspekte der `Mathematica` Kernsprache vorgestellt und erklärt. Diese sind bei der Bearbeitung des aktuellen Übungsblattes und der folgenden von großer Hilfe.

Aufgabe 12: π per Monte-Carlo-Methode

Programmtestat und Ausarbeitung¹

Der Wert von π soll mittels Monte-Carlo-Methode approximiert werden. Das Verfahren basiert darauf, zufällig Punkte in einem Quadrat zu erzeugen und diese danach zu klassifizieren, ob sie innerhalb oder außerhalb des eingeschriebenen Kreises liegen. Das Verhältnis der Flächen von Kreis und Quadrat entspricht $\pi/4$ und kann durch das Verhältnis der jeweiligen „Treffer“ approximiert werden.

1. Generieren Sie $N = 1000$ Zufallszahlen innerhalb des Rechtecks $-1 \leq x, y \leq 1$. Bestimmen Sie dann das Verhältnis der Punkte im Einheitskreis und im Rechteck ($= N$), dieses ergibt eine Approximation für $\pi/4$.
2. Zeichnen Sie die Zufallspunkte, wobei diejenigen im Einheitskreis grün und diejenigen außerhalb schwarz sein sollen. Quadrat und Kreis sollen ebenfalls angezeigt werden.

Zur Ausarbeitung:

- Ausdruck des Programmes mit *handschriftlichen* Kommentaren versehen,
- Ausdruck des Plots für $N = 1000$ und
- Tabelle (*handschriftlich*) $\{N, \langle \text{Approx. für } \pi \rangle, \langle \text{rel. Abweichung (in \%)} \rangle\}$ für N aus $\{1000, 10000, 100000\}$.

Einige nützliche Befehle:

`Module`, `Table`, `Random`, `For`, `If`, `Join`, `Select`, `GatherBy`, `ListPlot`, `Show`,
`Graphics`, `Circle`, `Rectangle`.

Aufgabe 13: Perfekte, befreundete und gesellige Zahlen

freiwillig

Eine Zahl, deren Summe an echten Teilern (die Zahl selbst ist nicht eingeschlossen, 1 ist ein echter Teiler) wieder die Zahl ergibt, bezeichnet man als perfekte oder vollkommene Zahl. Ein

¹Es werden im zweiten Block der Vorlesung mindestens fünf schriftliche Ausarbeitungen angeboten, drei davon sind erforderlich.

Zahlenpaar (a, b) , bei dem die Summe der echten Teiler von a der Zahl b entspricht und umgekehrt, nennt man befreundete Zahlen.

1. Schreiben Sie zunächst eine Funktion, die alle echten Teiler einer Zahl bestimmt und diese in Form einer Liste zurückgibt. Hierfür können Sie einen naiven Algorithmus verwenden (also durch Ausprobieren).
2. Schreiben Sie nun eine Funktion, die Ihre Methode zur Bestimmung der echten Teiler verwendet und für zwei übergebene Zahlen `True` zurückgibt, falls die erste Zahl Summe der echten Teiler der zweiten Zahl ist und ansonsten `False`.
3. Damit können Sie nun zwei recht einfache Funktionen implementieren, die testen, ob eine Zahl perfekt oder ein Zahlenpaar befreundet ist. Finden Sie auf diese Weise die ersten vier vollkommenen Zahlen.

Die ersten vollkommenen Zahlen sind: 6, 28, 496, 8 128 und 33 550 336. Die ersten befreundeten Zahlen waren schon Pythagoras bekannt: 220 und 284. Erst Jahrhunderte später fand Fermat (17 296, 18 416) und Decartes (9 363 584, 9 437 056).

Wird die eben beschriebene Eigenschaft von einer bestimmten Menge von Zahlen erfüllt, spricht man von geselligen Zahlen. Diese Zahlen bilden eine Kette, in der die aktuelle Zahl gleich der Teilersumme der nächsten ist und die letzte entsprechend Teilersumme der ersten Zahl.

4. Implementieren Sie diese Eigenschaft ebenfalls als `Mathematica`-Modul, dem eine Liste als Argument dient. Testen Sie Ihr Modul anhand einiger (kleiner) Zyklen aus <http://djm.cc/sociable.txt>. Geben Sie evtl. eine Maximalanzahl an Iterationen vor. Was passiert, wenn Sie eine Primzahl eingeben? (Hinweis: Gehen Sie geschickt vor und verwenden Sie die Mengenoperationen von `Mathematica`!)

Stellen Sie zudem sicher, dass Ihre Funktionen tatsächlich nur auf Integer-Argumente wirken. Im Übrigen ist es ein offenes Problem der Mathematik, ob ungerade Zahlen in diesen Kategorien existieren.

Einige nützliche Befehle:

`Total`, `Mod`, `While`, `Break`, `Intersection`, `Complement`, `Options`, `Map`, `Append`, `Prepend`, `Part`, `Span`.

Hinweis:

Mit dem Rechnernamen `fphctssh.physik.uni-karlsruhe.de` können Sie von überall aus mittels `ssh/scp` Programm auf einen Poolrechner zugreifen.
