

Monte Carlo-Simulation

Typische Fragestellungen:

Mathematik: Berechne die Wahrscheinlichkeit eines Ereignisses.

Technik: Berechne wichtige Kenngrößen einer Konstruktion (Belastung, Tragfähigkeit).

Bei manchen dieser Fragestellungen spielt der Zufall eine Rolle, weshalb man kein exaktes Resultat sondern nur eine Wahrscheinlichkeit und/oder Schätzung erhalten kann (z.B. Belastung einer Struktur durch Verkehr, Erdbeben ...)

Manchmal wäre zwar das Resultat exakt aus allen Daten berechenbar, aber leider stehen diese nicht zur Verfügung oder die Berechnung wird zu kompliziert oder zu langwierig.

Für solche Fälle bietet sich die Monte Carlo-Simulation an. Man modelliert das ganze System als Computermodell und lässt den Computer extrem viele Beispiele erzeugen und statistisch auswerten. Aus der statistischen Erfassung der Ergebnisse lassen sich so Rückschlüsse auf die gesuchten Werte machen:

z.B. ein Würfelspiel:

A hat 2 Würfel, B hat einen Würfel. A schlägt B, wenn einer seiner Würfel mehr Augen anzeigt als der Würfel von B, andernfalls verliert er. Wie gut sind seine Chancen?

Dieses Problem lässt sich mit der Wahrscheinlichkeitsrechnung bzw. Kombinatorik exakt lösen. Man kann aber ebenso leicht den Computer einige 1000 Mal gegen sich selbst würfeln und den Sieger ermitteln lassen.

Ist p die Gewinnwahrscheinlichkeit von A und q jene von B, so gilt $p+q = 1$. Das Gesetz der großen Zahlen besagt, dass A bei einer hohen Anzahl N von Spielen mit hoher Wahrscheinlichkeit $p * N$ gewinnen wird. Hat er also in der Simulation genau a Spiele tatsächlich gewonnen, so folgt

$$a \sim p * N \quad \text{oder} \quad p \sim a / N$$

Der Computer spielt also N mal, zählt die Anzahl a der Siege von A und dividiert: $p \sim a / N$. Man approximiert die Wahrscheinlichkeit p durch die **relative Häufigkeit** a / N .

Das Problem ist nur noch: Wie würfelt man in C/C++?