Quasi Monte-Carlo Methoden

Marcel Horstmann

26.06.2008

Einleitung / Motivation

- Beispiel: Bestimmung von π
  - mit der Monte-Carlo Methode
  - mit der Quasi Monte-Carlo Methode

Fehlerabschätzung für QMC-Integration

- Maß für Gleichmäßigkeit von Punktfolgen
- Fehlerschranke: Koksma-Hlawka

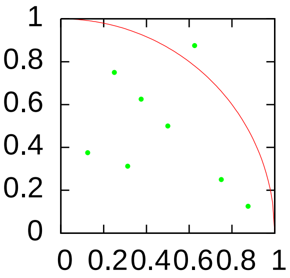
Quasi-Zufallsgeneratoren

- intuitiver Ansatz: Gitter
- Van der Corput Sequenz
- Halton
- Sobol

Bestimmung von π mit Monte-Carlo

Ermittlung von π über das Verhältnis von Kreisfläche A_k zu Quadratfläche A_Q

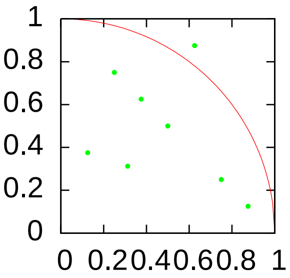
- $A_Q = r^2$



Bestimmung von π mit Monte-Carlo

Ermittlung von π über das Verhältnis von Kreisfläche A_k zu Quadratfläche A_Q

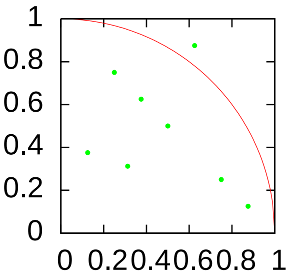
- $A_Q = r^2$
- Viertelkreis: $A_k = \frac{1}{4}\pi r^2$



Bestimmung von π mit Monte-Carlo

Ermittlung von π über das Verhältnis von Kreisfläche A_k zu Quadratfläche A_Q

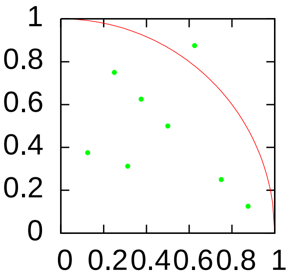
- $A_Q = r^2$
- Viertelkreis: $A_k = \frac{1}{4}\pi r^2$
- $\pi = 4 \cdot \frac{A_k}{A_Q} = 4 \cdot \frac{AnzahlTreffer}{AnzahlWuerfe}$



Bestimmung von π mit Monte-Carlo

Ermittlung von π über das Verhältnis von Kreisfläche A_k zu Quadratfläche A_Q

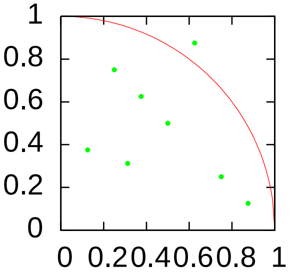
- $A_Q = r^2$
- Viertelkreis: $A_k = \frac{1}{4}\pi r^2$
- $\pi = 4 \cdot \frac{A_k}{A_Q} = 4 \cdot \frac{AnzahlTreffer}{AnzahlWuerfe}$
- nebenstehendes Beispiel: 8 Würfe, 7 Treffer

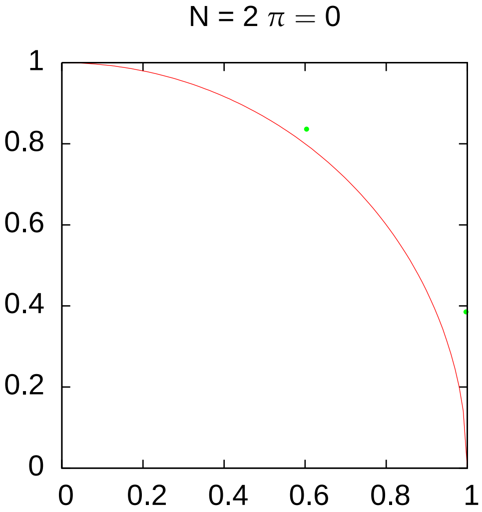


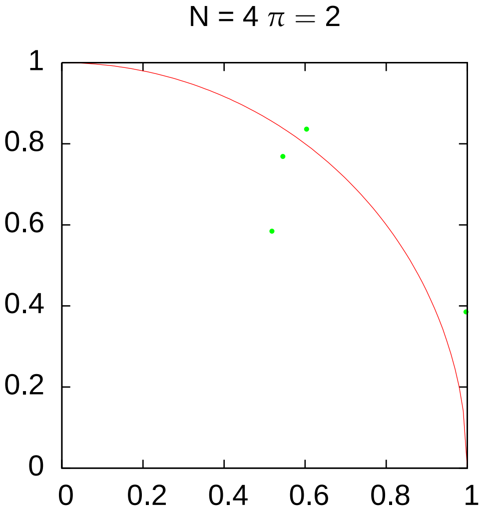
Bestimmung von π mit Monte-Carlo

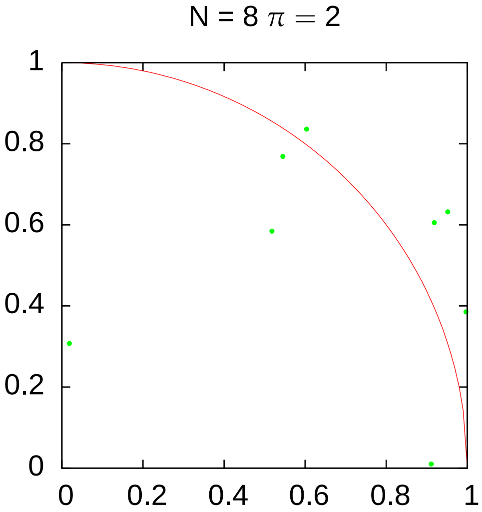
Ermittlung von π über das Verhältnis von Kreisfläche A_k zu Quadratfläche A_Q

- $A_Q = r^2$
- Viertelkreis: $A_k = \frac{1}{4}\pi r^2$
- $\pi = 4 \cdot \frac{A_k}{A_Q} = 4 \cdot \frac{AnzahlTreffer}{AnzahlWuerfe}$
- nebenstehendes Beispiel: 8 Würfe, 7 Treffer
  - $\pi = 4 \cdot \frac{7}{8} = 3,5$

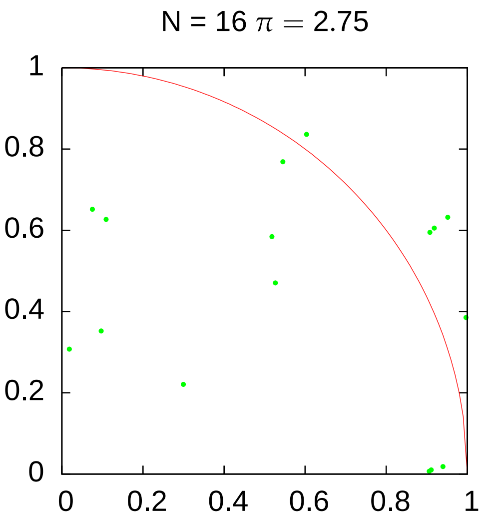


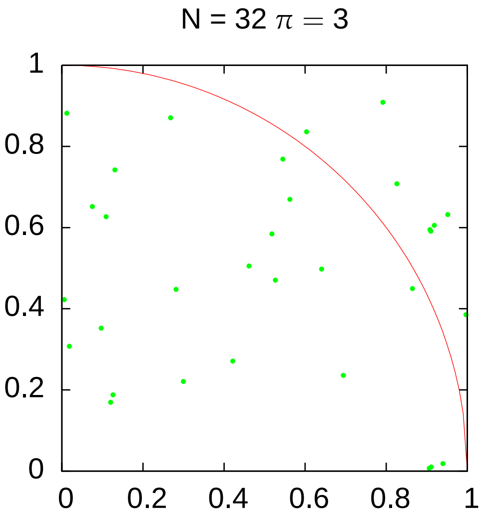

N = 2 $\pi = 0$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

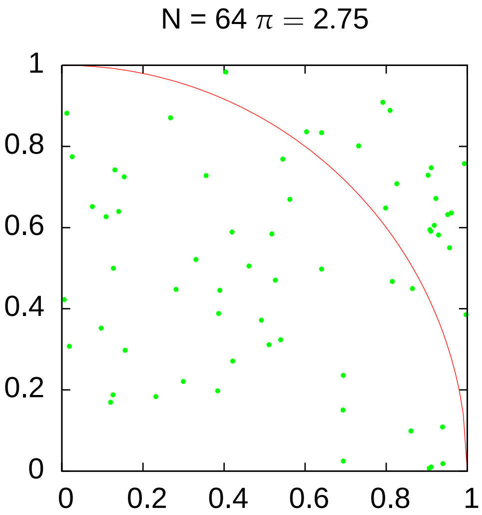

N = 4 $\pi = 2$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

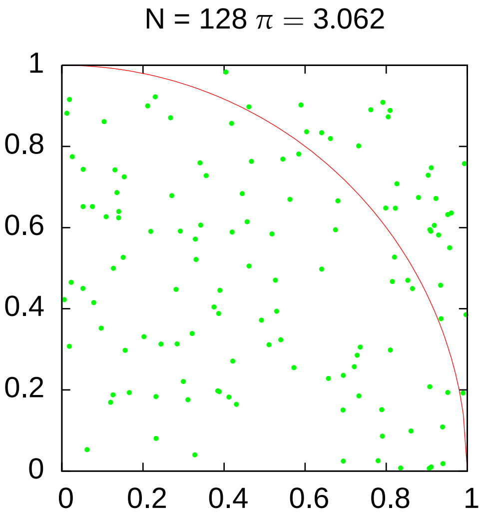

N = 8 $\pi = 2$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

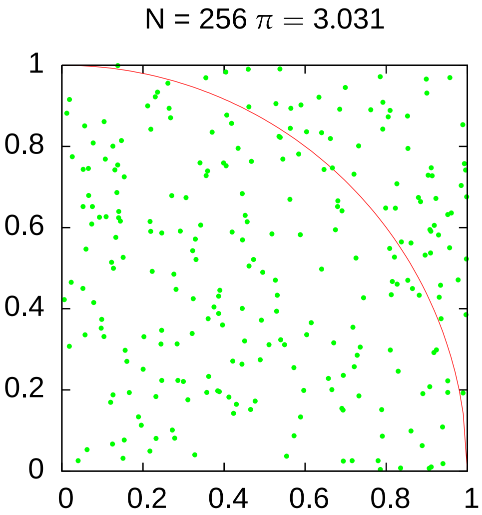
Beispiel: Bestimmung von π

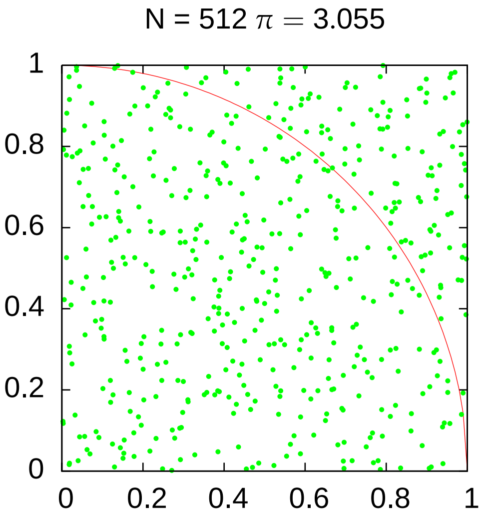


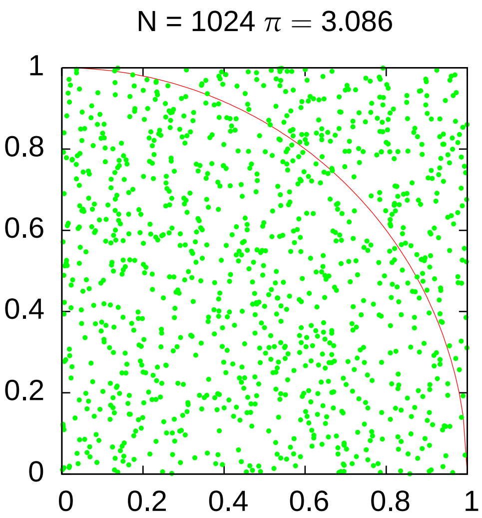

N = 32 $\pi = 3$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

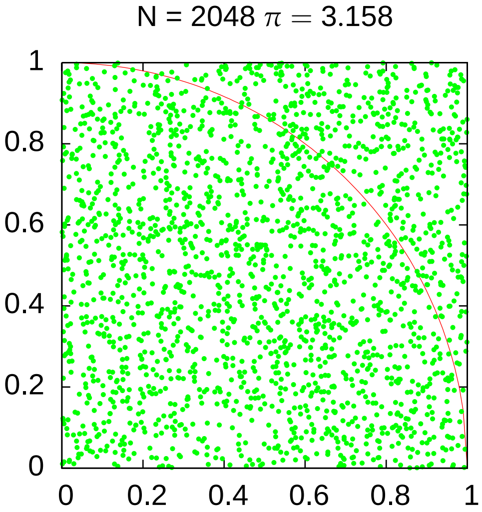

N = 64 $\pi = 2.75$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

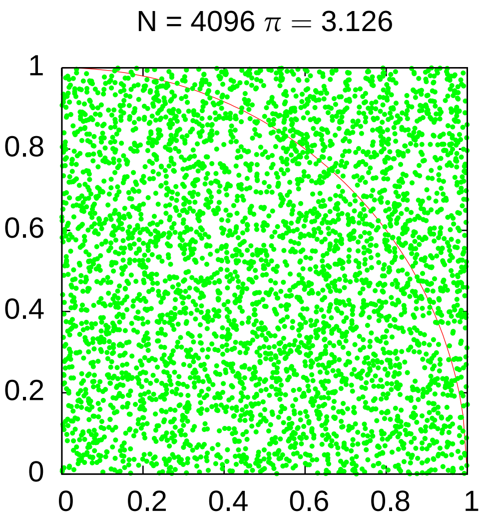

N = 128 $\pi = 3.062$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

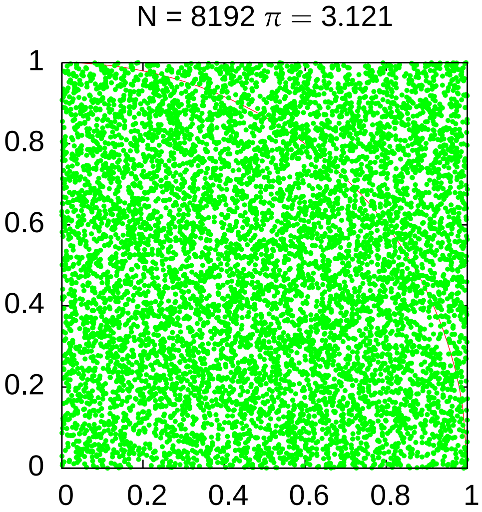

N = 256 $\pi = 3.031$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

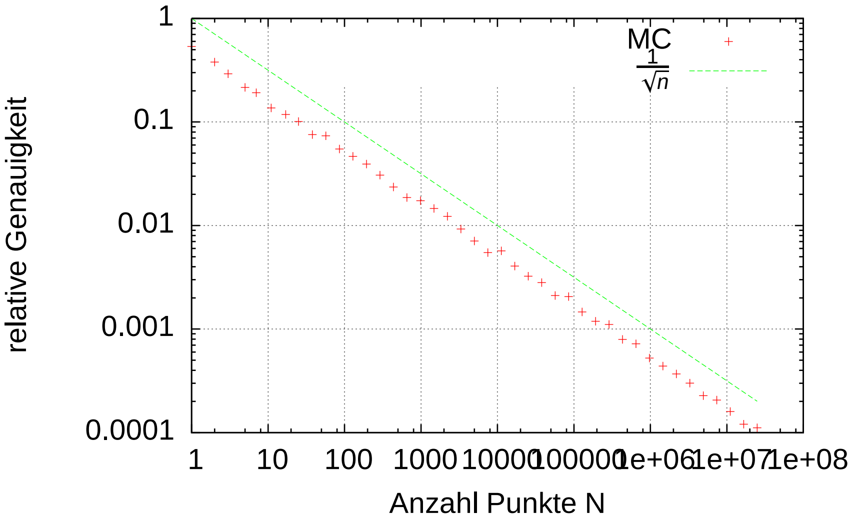

N = 512 $\pi = 3.055$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


N = 1024 $\pi = 3.086$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


N = 2048 $\pi = 3.158$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


N = 4096 $\pi = 3.126$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


N = 8192 $\pi = 3.121$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


relative Genauigkeit
1
0.1
0.01
0.001
0.0001
MC
$\frac{1}{\sqrt{n}}$
1
10
100
1000
10000
100000
1e+06
1e+07
1e+08
Anzahl Punkte N

Monte-Carlo Methoden

$$\int f\, d\, V \approx \frac{1}{N} \sum_{i=0}^{N-1} f(x_i) \pm V \sqrt{\frac{Var(f)}{N}}$$

Monte-Carlo Methoden

$$\int f \, d \, V \approx \frac{1}{N} \sum_{i=0}^{N-1} f(x_i) \pm V \sqrt{\frac{Var(f)}{N}}$$

- $\frac{1}{\sqrt{N}}$ - Konvergenz bedeutet:
  eine Größenordnung mehr Genauigkeit
  -> zwei Größenordnungen mehr Rechenaufwand!

Monte-Carlo Methoden

$$\int f \, d\,V \approx \frac{1}{N} \sum_{i=0}^{N-1} f(x_i) \pm V \sqrt{\frac{Var(f)}{N}}$$

- $\frac{1}{\sqrt{N}}$ - Konvergenz bedeutet:
  eine Größenordnung mehr Genauigkeit
  -> zwei Größenordnungen mehr Rechenaufwand!
- Das dieser Wurzelterm so vertraut ist bedeutet aber nicht, das es nicht auch besser geht!

Monte-Carlo Methoden

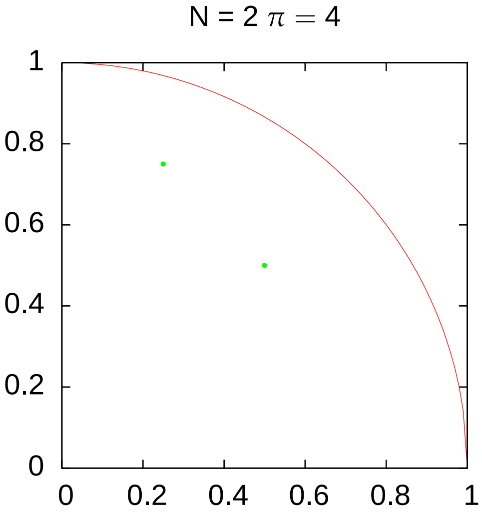
$$\int f\, d\, V \approx \frac{1}{N} \sum_{i=0}^{N-1} f(x_i) \pm V \sqrt{\frac{Var(f)}{N}}$$

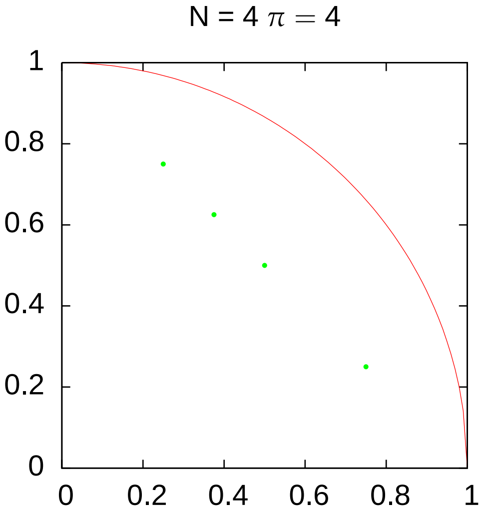
- $\frac{1}{\sqrt{N}}$ - Konvergenz bedeutet:
  eine Größenordnung mehr Genauigkeit
  -> zwei Größenordnungen mehr Rechenaufwand!
- Das dieser Wurzelterm so vertraut ist bedeutet aber nicht, das es nicht auch besser geht!
- Eine bessere Konvergenz lässt sich durch gleichmäßigere Punktwahl erreichen

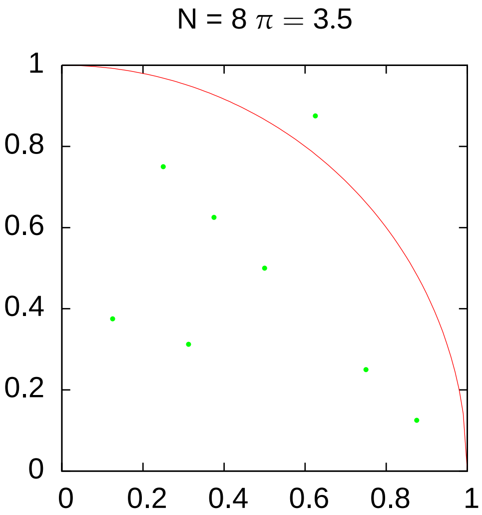
Monte-Carlo Methoden

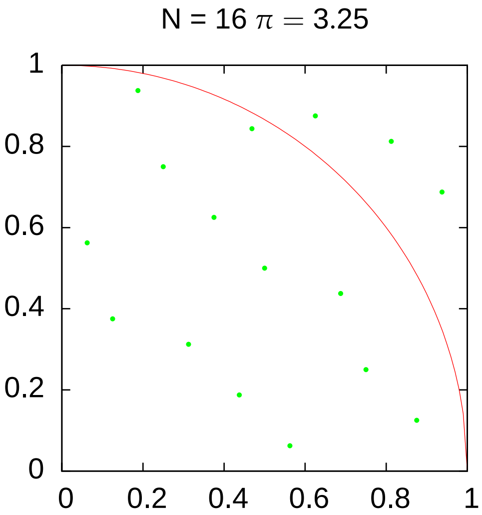
$$\int f \, dV \approx \frac{1}{N} \sum_{i=0}^{N-1} f(x_i) \pm V \sqrt{\frac{Var(f)}{N}}$$

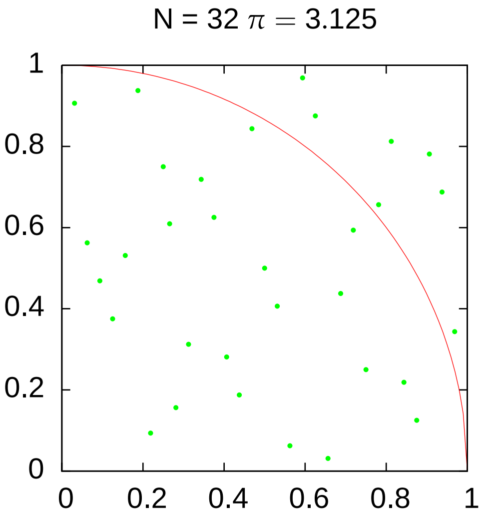
- $\frac{1}{\sqrt{N}}$ - Konvergenz bedeutet:
  eine Größenordnung mehr Genauigkeit
  -> zwei Größenordnungen mehr Rechenaufwand!
- Das dieser Wurzelterm so vertraut ist bedeutet aber nicht, das es nicht auch besser geht!
- Eine bessere Konvergenz lässt sich durch gleichmäßigere Punktwahl erreichen
  - Quasi Monte-Carlo Methoden!


N = 2 $\pi = 4$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

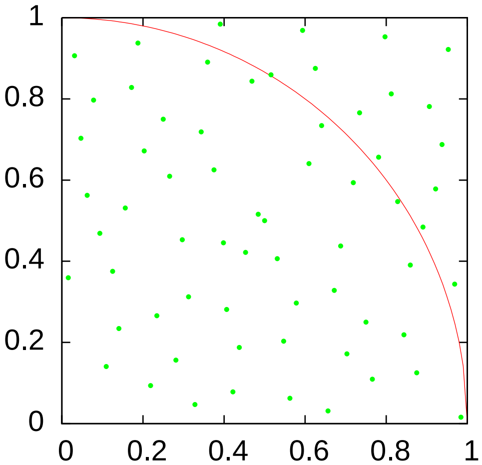

N = 4 $\pi = 4$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

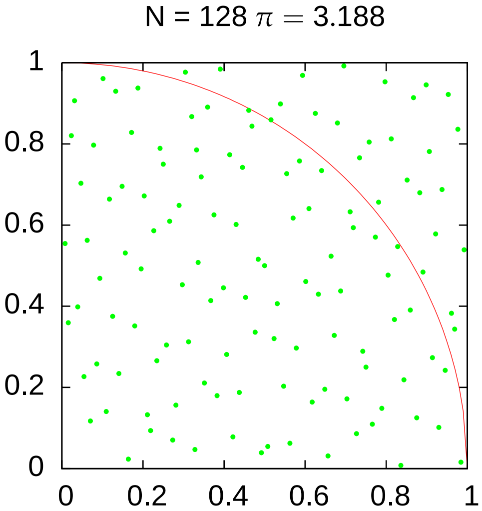

N = 8 $\pi = 3.5$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


N = 16 $\pi = 3.25$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

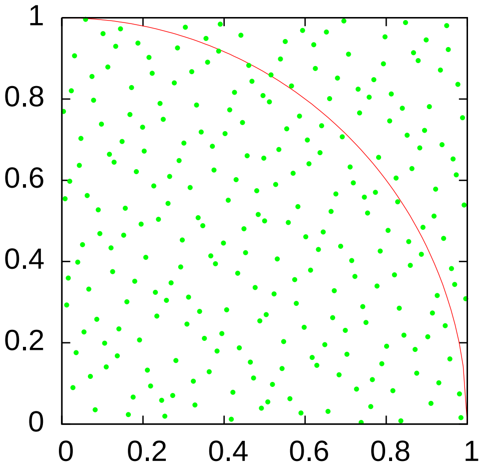

N = 32 $\pi = 3.125$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

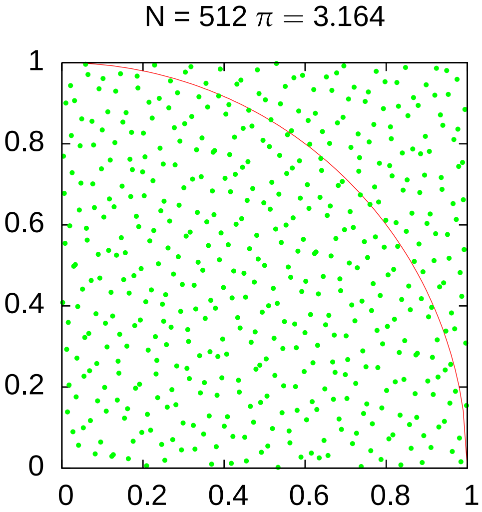
N = 64 $\pi = 3.125$

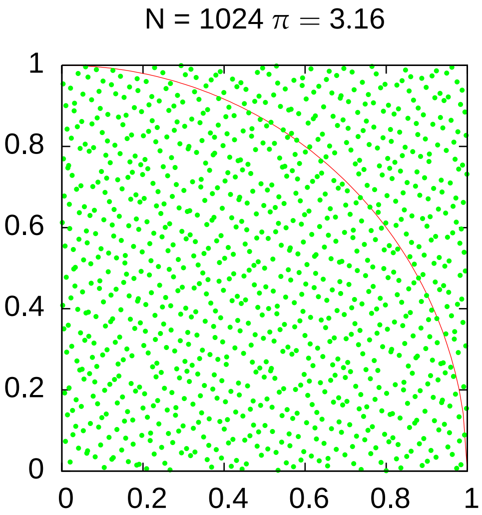


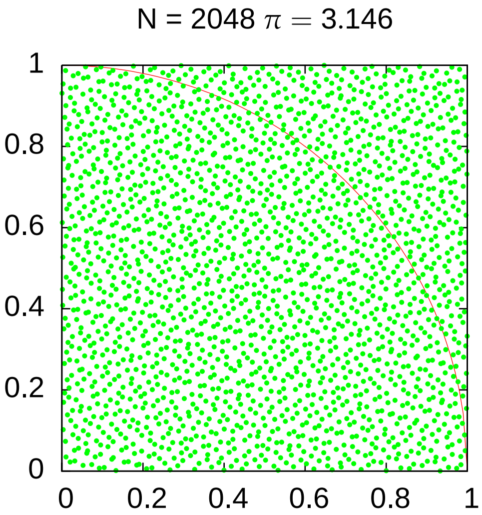

N = 128 $\pi = 3.188$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

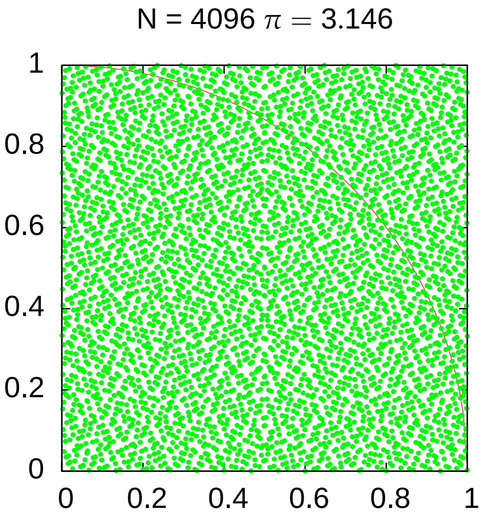
N = 256 $\pi = 3.188$

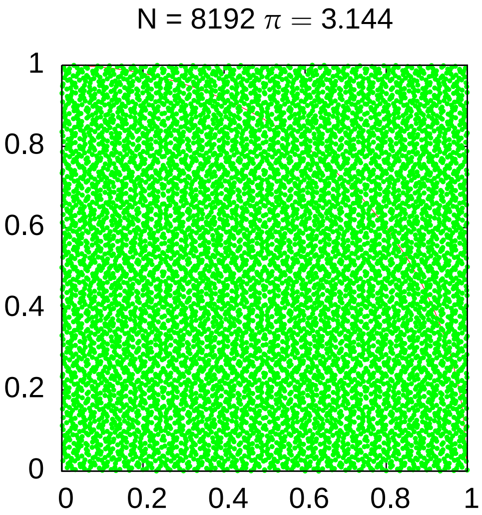


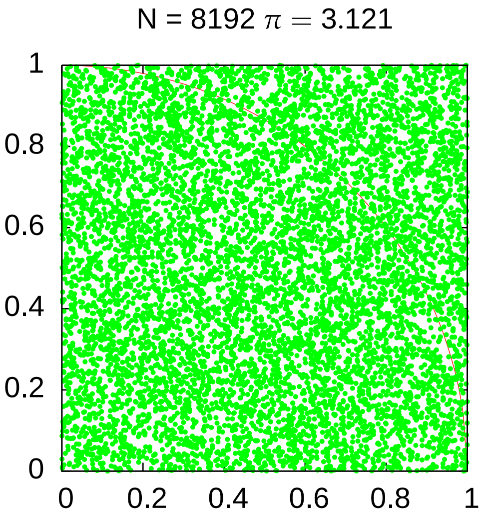

N = 512 $\pi = 3.164$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

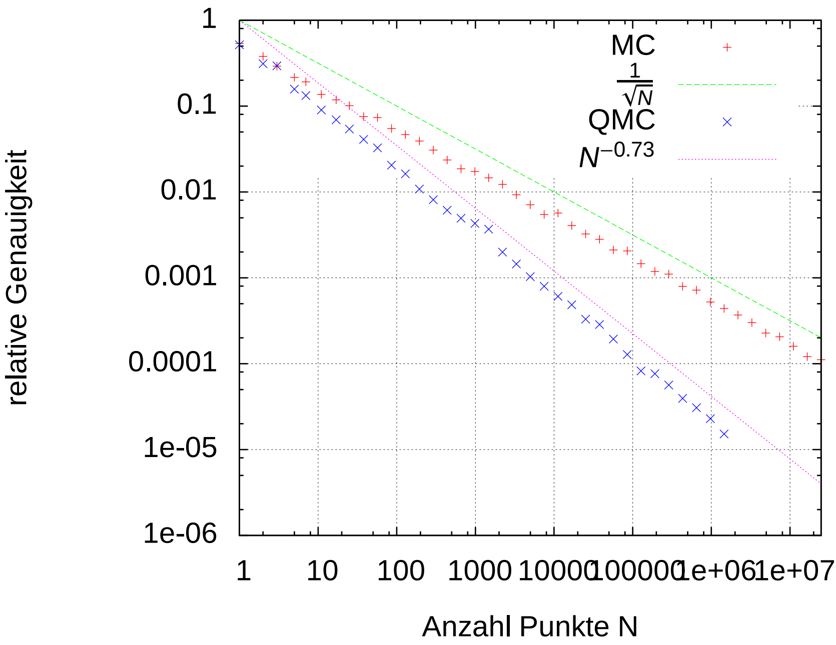

N = 1024 $\pi = 3.16$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


N = 2048 $\pi = 3.146$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


N = 4096 $\pi = 3.146$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


N = 8192 $\pi = 3.144$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


N = 8192 π = 3.121
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


MC
$\frac{1}{\sqrt{N}}$
QMC
$N^{-0.73}$
relative Genauigkeit
Anzahl Punkte N
1
0.1
0.01
0.001
0.0001
1e-05
1e-06
1
10
100
1000
10000
100000
1e+06
1e+07

Wie groß ist der Fehler bei QMC-Integration?

$$\int_{[0,1)^d} f(x)dx \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i) \tag{1}$$

Wie groß ist der Fehler bei QMC-Integration?

$$\int_{[0,1)^d} f(x)dx \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i) \tag{1}$$

Wie groß ist der Fehler bei QMC-Integration?

$$\int_{[0,1)^d} f(x)dx \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i) \tag{1}$$

- Fehler hängt offensichtlich von Anzahl der Punkte und deren „Gleichmäßigkeit“ ab

Wie groß ist der Fehler bei QMC-Integration?

$$\int_{[0,1)^d} f(x)dx \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i) \tag{1}$$

- Fehler hängt offensichtlich von Anzahl der Punkte und deren „Gleichmäßigkeit“ ab
- Um konkretere Aussagen machen zu können benötigen wir zunächst ein Maß für diese „Gleichmäßigkeit“

Wie groß ist der Fehler bei QMC-Integration?

$$\int_{[0,1)^d} f(x)dx \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i) \tag{1}$$

- Fehler hängt offensichtlich von Anzahl der Punkte und deren „Gleichmäßigkeit“ ab
- Um konkretere Aussagen machen zu können benötigen wir zunächst ein Maß für diese „Gleichmäßigkeit“
  - Die Diskrepanz!

Definition der Diskrepanz

Diskrepanz D: Abweichung der Punkte $x_1 \cdots x_n$ von idealer Gleichförmigkeit auf dem Intervall B: $[0,1)^d$

Definition der Diskrepanz

Diskrepanz D: Abweichung der Punkte $x_1 \cdots x_n$ von idealer Gleichförmigkeit auf dem Intervall B: $[0,1)^d$

$$D(x_1, \cdots, x_n; B) = sup_{A \in B} \left| \frac{\#\{x_i \in A\}}{n} - V(A) \right| \tag{2}$$

Definition der Diskrepanz

Diskrepanz D: Abweichung der Punkte $x_1 \cdots x_n$ von idealer Gleichförmigkeit auf dem Intervall B: $[0,1)^d$

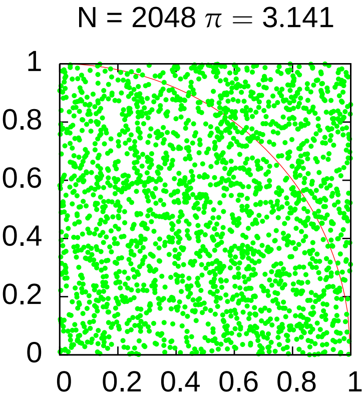
$$D(x_1, \cdots, x_n; B) = sup_{A \in B} \left| \frac{\#\{x_i \in A\}}{n} - V(A) \right| \qquad (2)$$

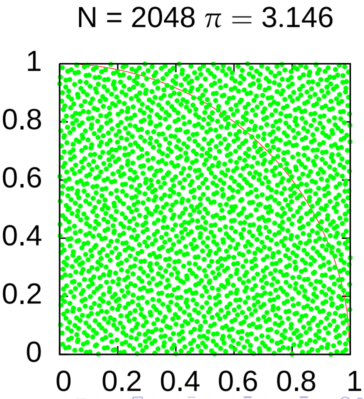
Dabei stellt A eine Teilmenge von B der Form:

$$\Pi_{j=1}^{d} [u_j, v_j), \; 0 \leq u_j < v_j \leq 1$$

dar.

Beispiel: LCG und Sobol





Diskrepanz eindimensionaler Folgen

- Für alle geeigneten Folgen x_i muss die Diskrepanz D für $n \to \infty$ verschwinden!

Diskrepanz eindimensionaler Folgen

- Für alle geeigneten Folgen x_i muss die Diskrepanz D für $n \to \infty$ verschwinden!
- Für endliche n:

Diskrepanz eindimensionaler Folgen

- Für alle geeigneten Folgen x_i muss die Diskrepanz D für $n \to \infty$ verschwinden!
- Für endliche n:
- in einer Dimension: $D(x_1, \cdots, x_n) \geq \frac{1}{n}$

Diskrepanz eindimensionaler Folgen

- Für alle geeigneten Folgen x_i muss die Diskrepanz D für $n \to \infty$ verschwinden!
- Für endliche n:
- in einer Dimension: $D(x_1, \cdots, x_n) \geq \frac{1}{n}$
  - Minimum wird durch: $x_i = \frac{2i-1}{n}$, $i = 1, ..., n$ realisiert. Problem hier: alle x_i werden durch Wahl von n festgelegt und unterscheiden sich sämtlich von der Folge zu $n+1$ - die Genauigkeit der Integration lässt sich also nicht beliebig steigern!

Diskrepanz eindimensionaler Folgen

In der Praxis relevanter:

- Durch einen Generator wird eine unendliche Folge $x_1, x_2, \cdots$ festgelegt

Diskrepanz eindimensionaler Folgen

In der Praxis relevanter:

- Durch einen Generator wird eine unendliche Folge $x_1, x_2, \cdots$ festgelegt
- Uns interessiert nun die Diskrepanz der n ersten Elemente dieser Folge!

Diskrepanz eindimensionaler Folgen

In der Praxis relevanter:

- Durch einen Generator wird eine unendliche Folge $x_1, x_2, \cdots$ festgelegt
- Uns interessiert nun die Diskrepanz der n ersten Elemente dieser Folge!
- Für diese existiert eine untere Schranke:

$$D(x_i, ..., x_n) \geq \frac{c\, logn}{n},\ c = const$$

Diskrepanz mehrdimensionaler Folgen

Wesentlich relevanter: Diskrepanz mehrdimensionaler Folgen!

Diskrepanz mehrdimensionaler Folgen

Wesentlich relevanter: Diskrepanz mehrdimensionaler Folgen!

- wesentlich weniger gesicherte Erkenntisse vorhanden

Diskrepanz mehrdimensionaler Folgen

Wesentlich relevanter: Diskrepanz mehrdimensionaler Folgen!

- wesentlich weniger gesicherte Erkenntisse vorhanden
- weit verbreitete Annahme: Die ersten n Elemente einer Folge $x_i, ..., x_n$ in Dimensionen $d \geq 2$ besitzen eine Diskrepanz von

$$D(x_1, ..., x_n) \geq c_d \frac{(logn)^d}{n}.$$

Diskrepanz mehrdimensionaler Folgen

Wesentlich relevanter: Diskrepanz mehrdimensionaler Folgen!

- wesentlich weniger gesicherte Erkenntisse vorhanden
- weit verbreitete Annahme: Die ersten n Elemente einer Folge $x_i, ..., x_n$ in Dimensionen $d \geq 2$ besitzen eine Diskrepanz von

$$D(x_1, ..., x_n) \geq c_d \frac{(log n)^d}{n}.$$

- Folgen die dies erreichen werden häufig als „low-discrepancy sequences“ bezeichnet

Dimensionsabhängigkeit

Obwohl $\frac{1}{n}$ schneller fällt wie jede Potenz von $log\ n$:

- für große Dimensionen wird $(log\ n)^d$ schnell sehr groß

Dimensionsabhängigkeit

Obwohl $\frac{1}{n}$ schneller fällt wie jede Potenz von $log\ n$:

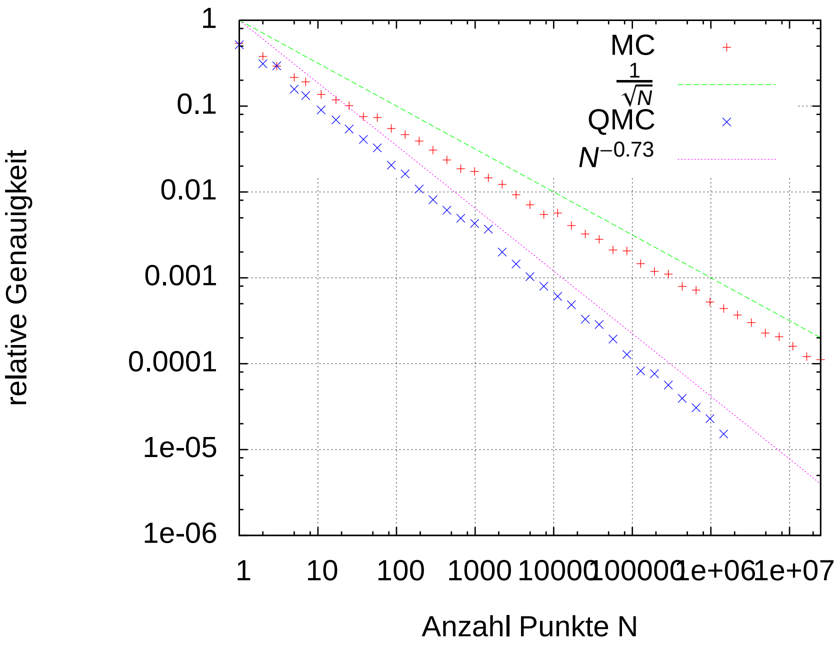
- für große Dimensionen wird $(log\ n)^d$ schnell sehr groß
- QMC-Methoden sind daher für sehr hochdimensionale Probleme tendenziell weniger geeignet

Dimensionsabhängigkeit

Obwohl $\frac{1}{n}$ schneller fällt wie jede Potenz von $log\ n$:

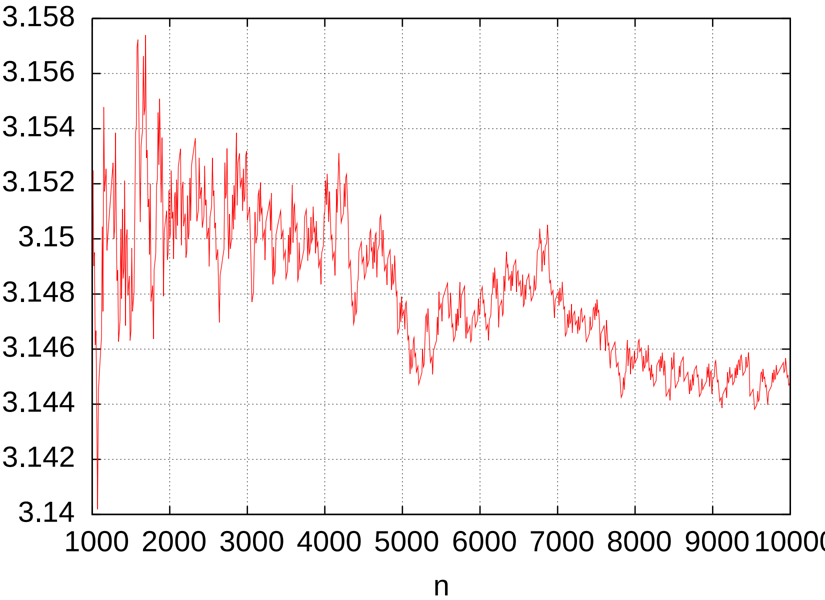
- für große Dimensionen wird $(log\ n)^d$ schnell sehr groß
- QMC-Methoden sind daher für sehr hochdimensionale Probleme tendenziell weniger geeignet
- allerdings zeigen viele Untersuchungen, das auch hochdimensionale finanzmathematische Probleme mit QMC effizienter wie mit MC zu lösen sind

Übersicht



MC
$\frac{1}{\sqrt{N}}$
QMC
$N^{-0.73}$
1
0.1
0.01
0.001
0.0001
1e-05
1e-06
1
10
100
1000
10000
100000
1e+06
1e+07
relative Genauigkeit
Anzahl Punkte N

Bestimmung von π mit QMC - wo abbrechen?



Fehlerschranke

$$\left| \int_{[0,1)^d} f(x)dx - \frac{1}{n} \sum_{i=1}^{n} f(x_i) \right|$$

Fehlerschranke

Koksma-Hlawka Ungleichung

$$\left| \int_{[0,1)^d} f(x)dx - \frac{1}{n} \sum_{i=1}^{n} f(x_i) \right| \leq V(f)\, D(x_1, \dots, x_n)$$

Fehlerschranke

Koksma-Hlawka Ungleichung

$$\left| \int_{[0,1)^d} f(x)dx - \frac{1}{n} \sum_{i=1}^{n} f(x_i) \right| \leq V(f)\, D(x_1, \ldots, x_n)$$

- $V(f)$ hängt nur von f ab

Fehlerschranke

Koksma-Hlawka Ungleichung

$$\left| \int_{[0,1)^d} f(x)dx - \frac{1}{n} \sum_{i=1}^{n} f(x_i) \right| \leq V(f)\, D(x_1, \dots, x_n)$$

- $V(f)$ hängt nur von f ab
- Diskrepanz D hängt nur vom verwendeten Generator ab

Fehlerschranke

Die Koksma-Hlawka Ungleichung ist zur Fehlerabschätzung leider relativ wertlos

- $V(f)$ ist kaum bestimmbar

Fehlerschranke

Die Koksma-Hlawka Ungleichung ist zur Fehlerabschätzung leider relativ wertlos

- $V(f)$ ist kaum bestimmbar
- vor allem meistens schwerer bestimmbar wie das eigentliche Integral!

Fehlerschranke

Die Koksma-Hlawka Ungleichung ist zur Fehlerabschätzung leider relativ wertlos

- $V(f)$ ist kaum bestimmbar
- vor allem meistens schwerer bestimmbar wie das eigentliche Integral!
- auch die Diskrepanz D ist im allgemeinen nicht bekannt

Fehlerschranke

Die Koksma-Hlawka Ungleichung ist zur Fehlerabschätzung leider relativ wertlos

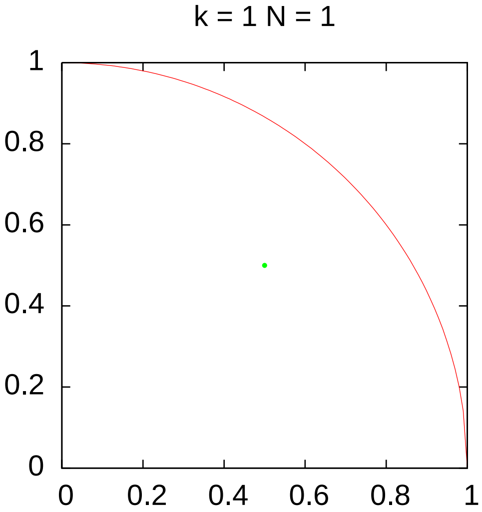
- $V(f)$ ist kaum bestimmbar
- vor allem meistens schwerer bestimmbar wie das eigentliche Integral!
- auch die Diskrepanz D ist im allgemeinen nicht bekannt
- Erheblicher Nachteil der Quasi-Monte-Carlo Methode gegenüber MC!

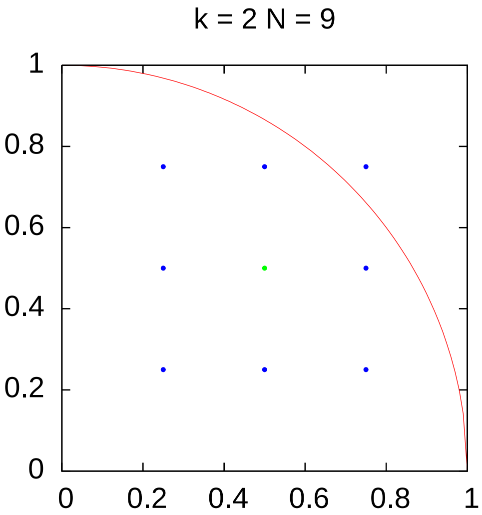
Fehlerschranke

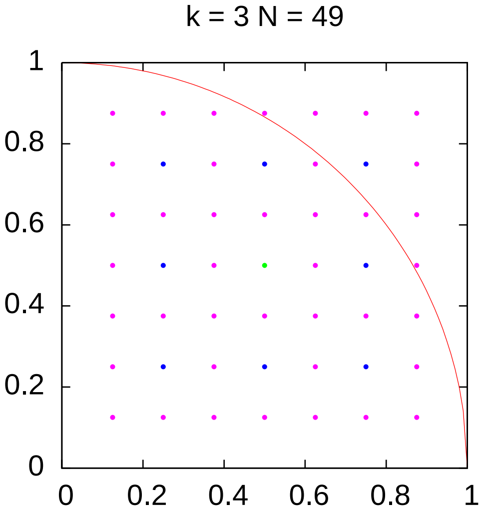
Die Koksma-Hlawka Ungleichung ist zur Fehlerabschätzung leider relativ wertlos

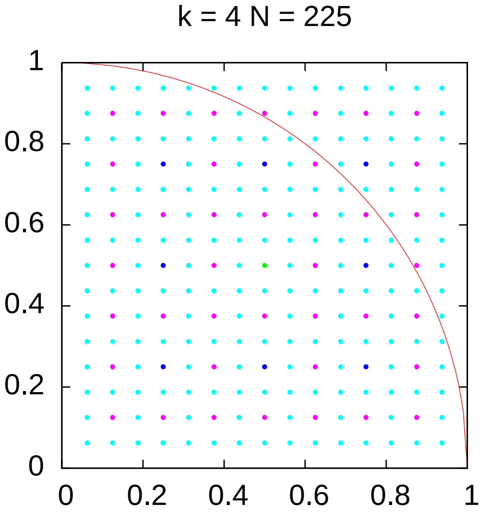
- $V(f)$ ist kaum bestimmbar
- vor allem meistens schwerer bestimmbar wie das eigentliche Integral!
- auch die Diskrepanz D ist im allgemeinen nicht bekannt
- Erheblicher Nachteil der Quasi-Monte-Carlo Methode gegenüber MC!
  - Ansatz hier: Randomized QMC (nur als Ausblick...)

Übersicht

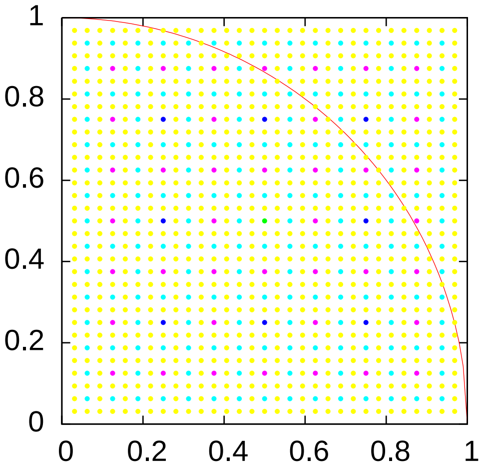

k = 1 N = 1
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


k = 2 N = 9
0
0.2
0.4
0.6
0.8
1
0
0.2
0.4
0.6
0.8
1


k = 3 N = 49
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


k = 4 N = 225
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

k = 5 -> $N = 31^2 = 2^{5 \cdot 2} - 2^5 = 961$



Gitter

$$N = 2^{kd} - 2^k$$

- Zahl der Punkte steigt sehr schnell mit k und der Anzahl der Dimensionen an!

Gitter

$$N = 2^{kd} - 2^k$$

- Zahl der Punkte steigt sehr schnell mit k und der Anzahl der Dimensionen an!
- typische Probleme in der Finanzmathematik: $d \approx 50$

Gitter

$$N = 2^{kd} - 2^k$$

- Zahl der Punkte steigt sehr schnell mit k und der Anzahl der Dimensionen an!
- typische Probleme in der Finanzmathematik: $d \approx 50$
- im ersten Durchlauf $N \approx 2^{50}$

Gitter

$$N = 2^{kd} - 2^k$$

- Zahl der Punkte steigt sehr schnell mit k und der Anzahl der Dimensionen an!
- typische Probleme in der Finanzmathematik: $d \approx 50$
- im ersten Durchlauf $N \approx 2^{50}$
  - viel, aber noch machbar!

Gitter

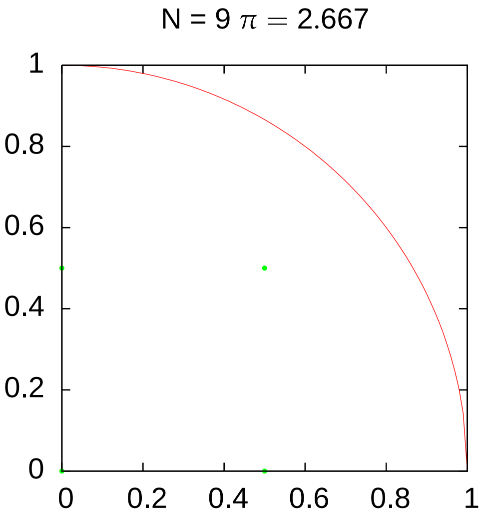
$$N = 2^{kd} - 2^{k}$$

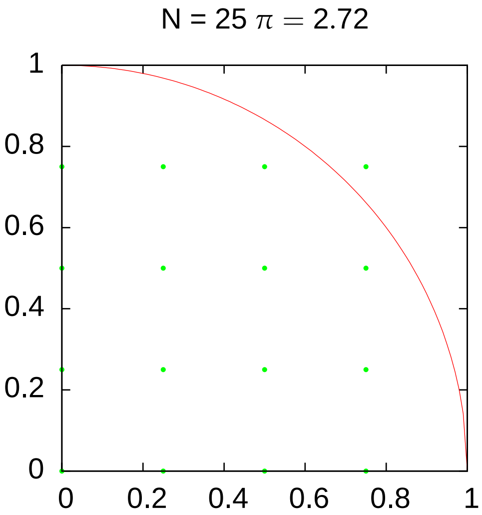
- Zahl der Punkte steigt sehr schnell mit k und der Anzahl der Dimensionen an!
- typische Probleme in der Finanzmathematik: $d \approx 50$
- im ersten Durchlauf $N \approx 2^{50}$
  - viel, aber noch machbar!
- für k = 2: $N \approx 2^{100}$

Gitter

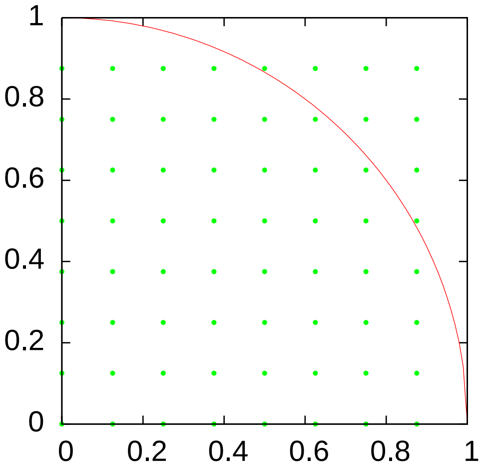
$$N = 2^{kd} - 2^k$$

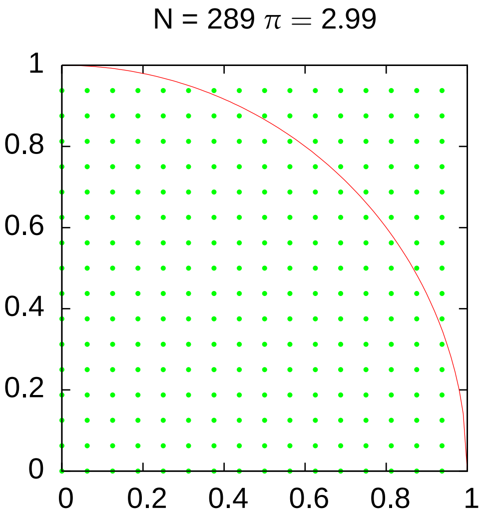
- Zahl der Punkte steigt sehr schnell mit k und der Anzahl der Dimensionen an!
- typische Probleme in der Finanzmathematik: $d \approx 50$
- im ersten Durchlauf $N \approx 2^{50}$
  - viel, aber noch machbar!
- für k = 2: $N \approx 2^{100}$
  - läuft auf heute verfügbaren Computern länger als das Universum alt ist!

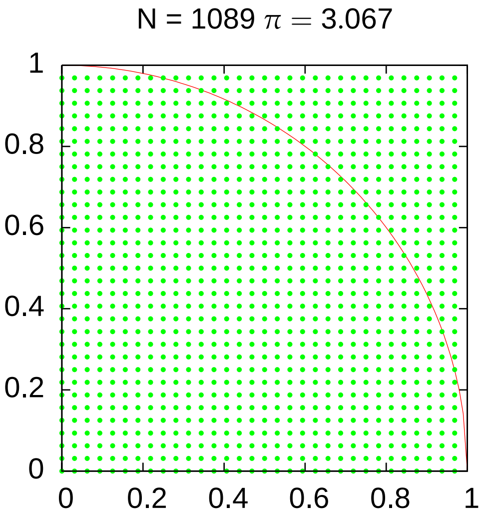

N = 9 $\pi = 2.667$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

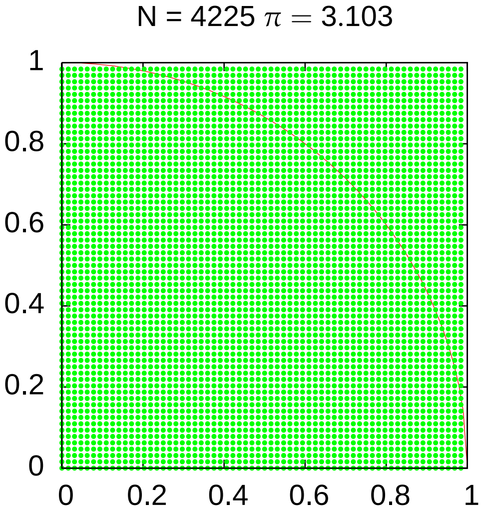

N = 25 $\pi = 2.72$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

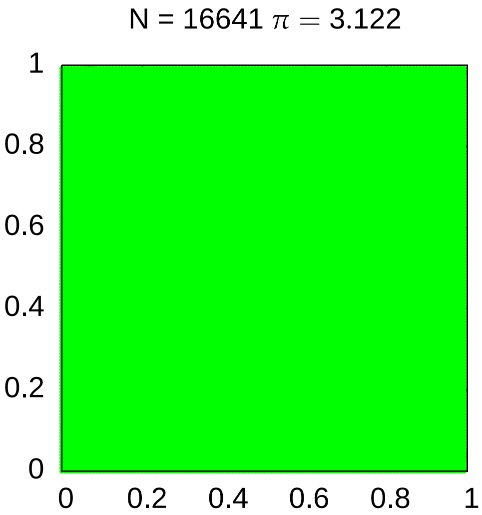
N = 81 $\pi = 2.864$




N = 289 $\pi = 2.99$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


N = 1089 $\pi = 3.067$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


N = 4225 $\pi = 3.103$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


N = 16641 $\pi = 3.122$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

weitere Nachteile

- zwischen den Punkten bleiben relativ große quadratische Löcher

weitere Nachteile

- zwischen den Punkten bleiben relativ große quadratische Löcher
- wie gerade am π-Beispiel gesehen resultiert daraus schlechte Konvergenz!

Van der Corput Sequenz

0 0 0 0

Van der Corput Sequenz

0	0	0	0
1	1	0.	

Van der Corput Sequenz

0	0	0	0
1	1	0.1	

Van der Corput Sequenz

0	0	0	0
1	1	0.1	1/2

Van der Corput Sequenz

0	0	0	0
1	1	0.1	1/2
2	10	0.	

Van der Corput Sequenz

0	0	0	0
1	1	0.1	1/2
2	10	0.	

Van der Corput Sequenz

0	0	0	0
1	1	0.1	1/2
2	10	0. 1	

Van der Corput Sequenz

0	0	0	0
1	1	0.1	1/2
2	10	0.01	1/4

Van der Corput Sequenz

0	0	0	0
1	1	0.1	1/2
2	10	0.01	1/4
3	11	0.11	3/4

Van der Corput Sequenz

0	0	0	0
1	1	0.1	1/2
2	10	0.01	1/4
3	11	0.11	3/4
4	100	0.001	1/8

Van der Corput Sequenz

0	0	0	0
1	1	0.1	1/2
2	10	0.01	1/4
3	11	0.11	3/4
4	100	0.001	1/8
5	101	0.101	5/8

Van der Corput Sequenz

0	0	0	0
1	1	0.1	1/2
2	10	0.01	1/4
3	11	0.11	3/4
4	100	0.001	1/8
5	101	0.101	5/8
6	110	0.011	3/8

Van der Corput Sequenz

0	0	0	0
1	1	0.1	1/2
2	10	0.01	1/4
3	11	0.11	3/4
4	100	0.001	1/8
5	101	0.101	5/8
6	110	0.011	3/8
7	111	0.111	7/8

Van der Corput Sequenz

0	0	0	0
1	1	0.1	1/2
2	10	0.01	1/4
3	11	0.11	3/4
4	100	0.001	1/8
5	101	0.101	5/8
6	110	0.011	3/8
7	111	0.111	7/8

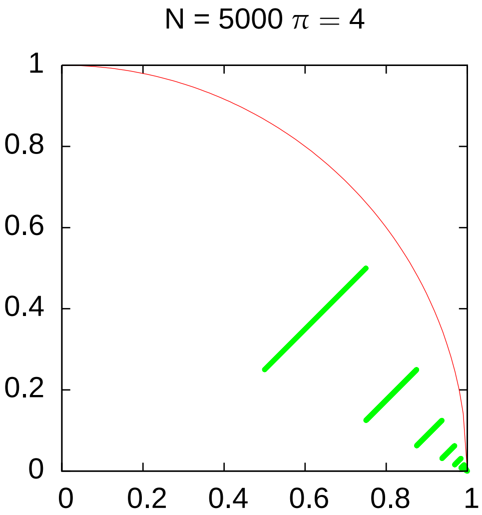
Das sieht doch ganz erfolgsversprechend aus!

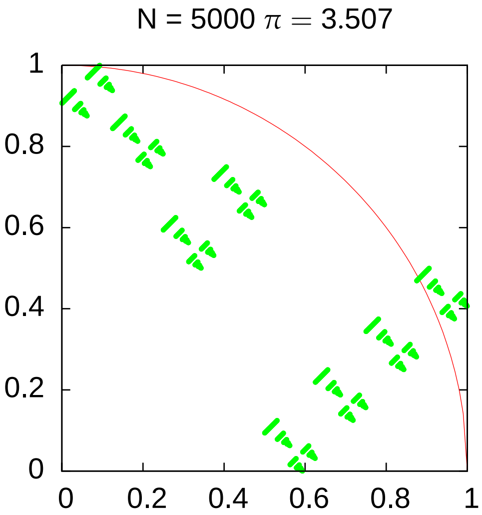
Van der Corput Sequenz

0	0	0	0
1	1	0.1	1/2
2	10	0.01	1/4
3	11	0.11	3/4
4	100	0.001	1/8
5	101	0.101	5/8
6	110	0.011	3/8
7	111	0.111	7/8

Das sieht doch ganz erfolgsversprechend aus!

- versuchen wir doch mal damit π zu bestimmen...


N = 5000 $\pi = 4$
0
0.2
0.4
0.6
0.8
1


N = 5000 $\pi = 3.507$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

Erweiterung auf mehrere Dimensionen

Ups! So gehts also nicht

Erweiterung auf mehrere Dimensionen

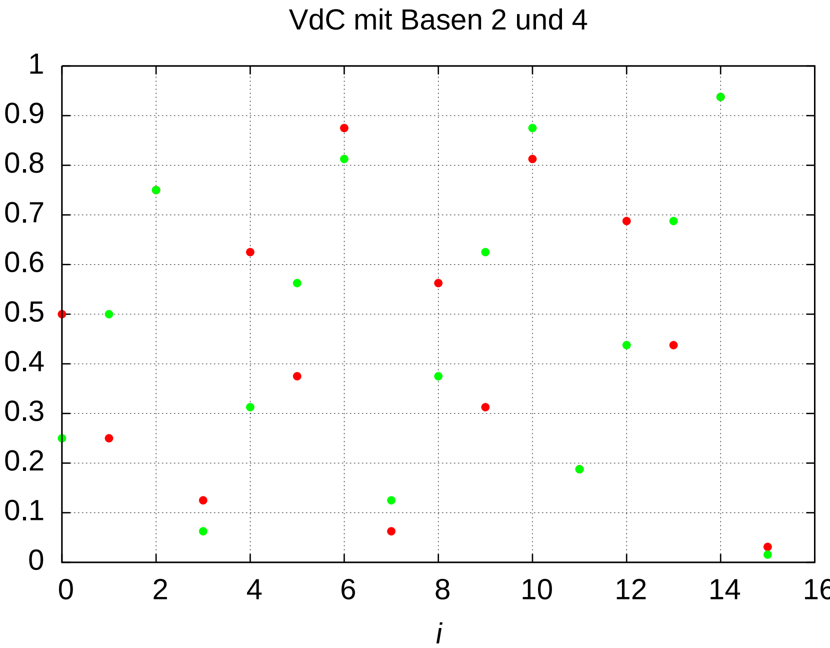
Van der Corpus Sequenz auf mehrere Dimensionen erweitern

- Halton Sequenz!

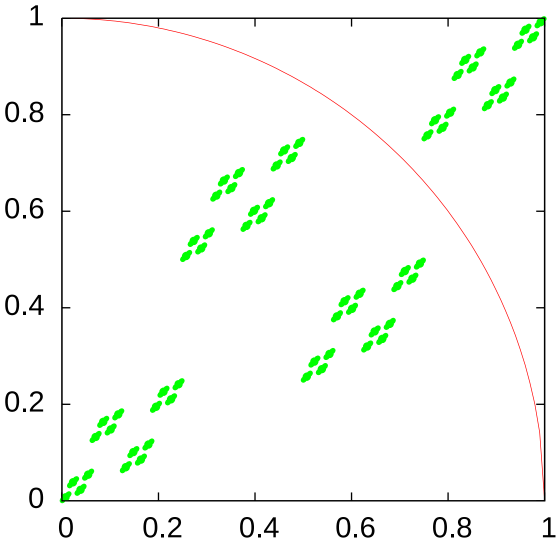
Erweiterung auf mehrere Dimensionen

Van der Corpus Sequenz auf mehrere Dimensionen erweitern

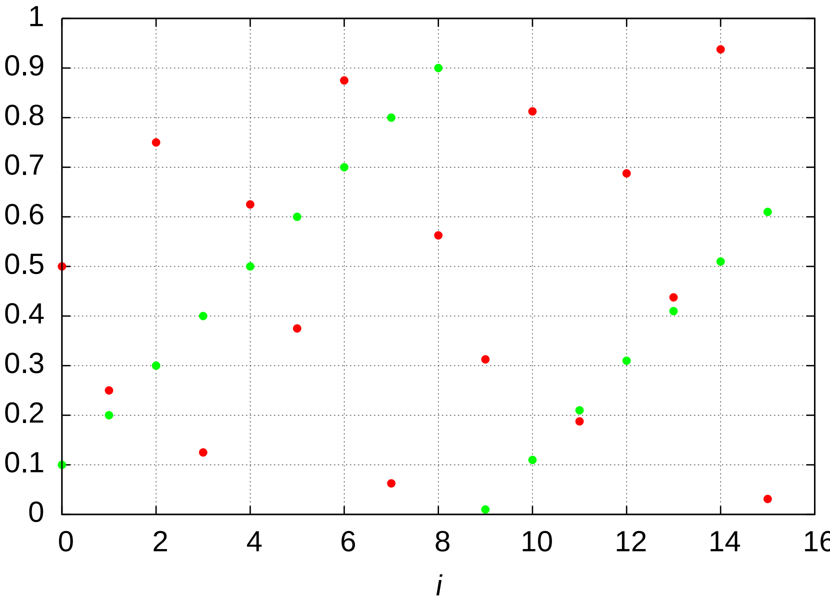
- Halton Sequenz!
- für jede Dimension eine andere VdC Sequenz mit aufeinanderfolgenden Primzahlen $b = 2, 3, 5, 7, 11, \cdots$ als Basen

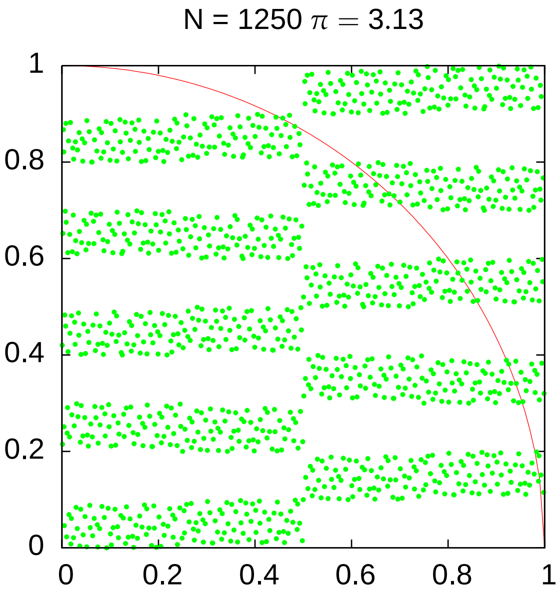

VdC mit Basen 2 und 4
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
2
4
6
8
10
12
14
16
i

N = 1250 $\pi = 3.002$



VdC mit Basen 2 und 10

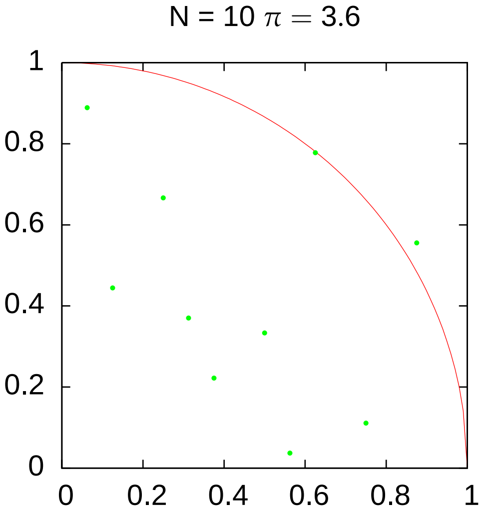


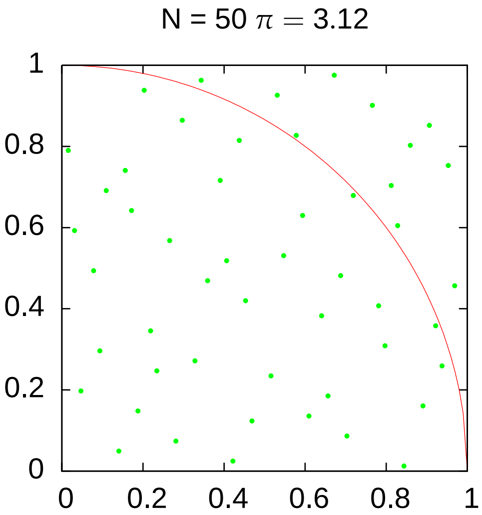

N = 1250 $\pi = 3.13$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

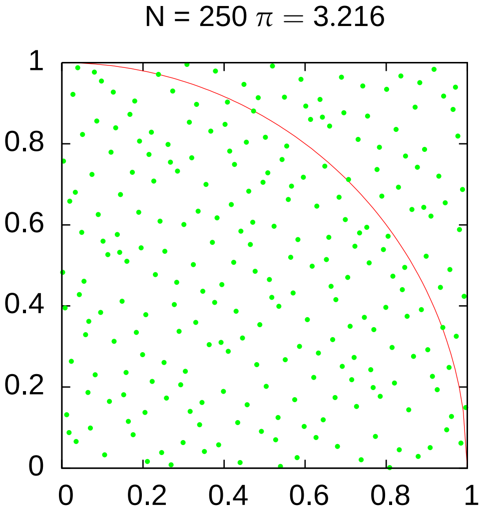
Erweiterung auf mehrere Dimensionen

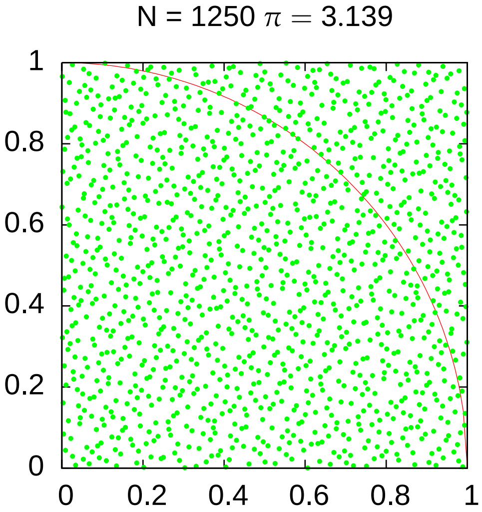
Van der Corpus Sequenz auf mehrere Dimensionen erweitern

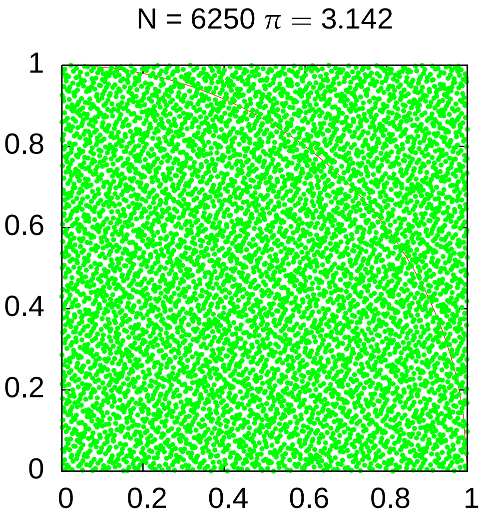
- Halton Sequenz!
- für jede Dimension eine andere VdC Sequenz mit aufeinanderfolgenden Primzahlen $b = 2, 3, 5, 7, 11, \cdots$ als Basen
- π bestimmen!


N = 10 $\pi = 3.6$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


N = 50 $\pi = 3.12$
0
0.2
0.4
0.6
0.8
1


N = 250 $\pi = 3.216$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


N = 1250 $\pi = 3.139$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


N = 6250 $\pi = 3.142$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

höhere Dimensionen

Die Bestimmung von π mit Halton funktioniert also.

höhere Dimensionen

Die Bestimmung von π mit Halton funktioniert also.

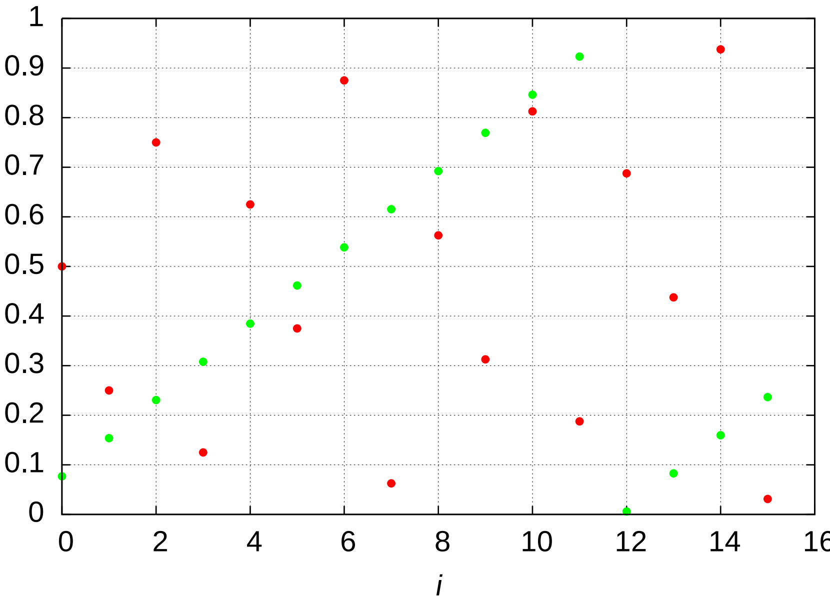
- Allerdings ist dieses Problem auch nur 2-dimensional!

höhere Dimensionen

Die Bestimmung von π mit Halton funktioniert also.

- Allerdings ist dieses Problem auch nur 2-dimensional!
- Um höherdimensionale Probleme zu lösen werden entsprechend größere Primzahlen benötigt

6 Dimensionen: $b_1 = 2$ und $b_6 = 13$



höhere Dimensionen

Die Bestimmung von π mit Halton funktioniert also.

- Allerdings ist dieses Problem auch nur 2-dimensional!
- Um höherdimensionale Probleme zu lösen werden entsprechend größere Primzahlen benötigt
- Allerdings wird dadurch die Diskrepanz der Folgen immer größer, und damit die Konvergenz immer schlechter ...

höhere Dimensionen

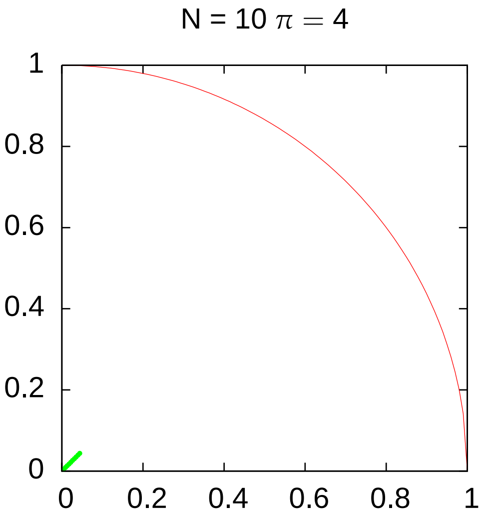
Die Bestimmung von π mit Halton funktioniert also.

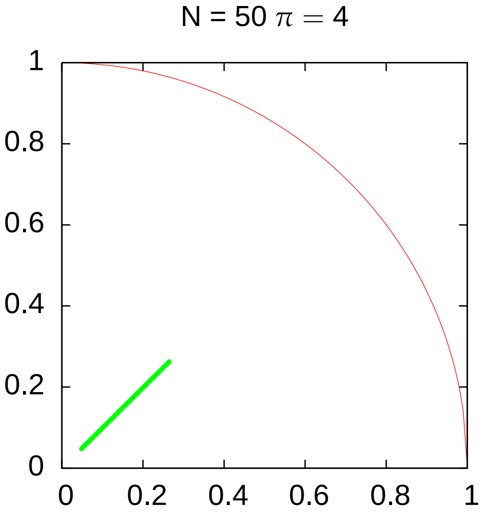
- Allerdings ist dieses Problem auch nur 2-dimensional!
- Um höherdimensionale Probleme zu lösen werden entsprechend größere Primzahlen benötigt
- Allerdings wird dadurch die Diskrepanz der Folgen immer größer, und damit die Konvergenz immer schlechter ...
- Beispiel: 50 Dimensionen
  $b_{49} = 227$ und $b_{50} = 229$ sind dann die größten Basen

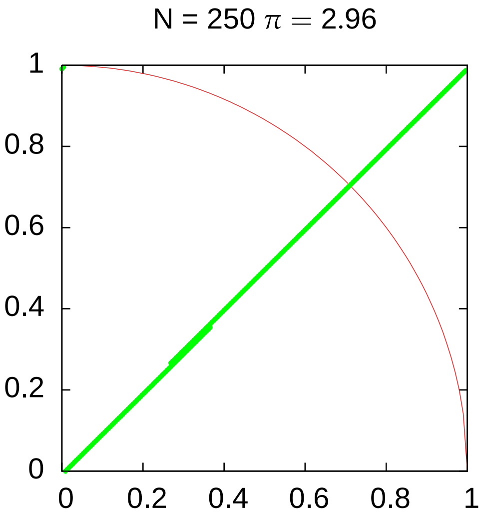
höhere Dimensionen

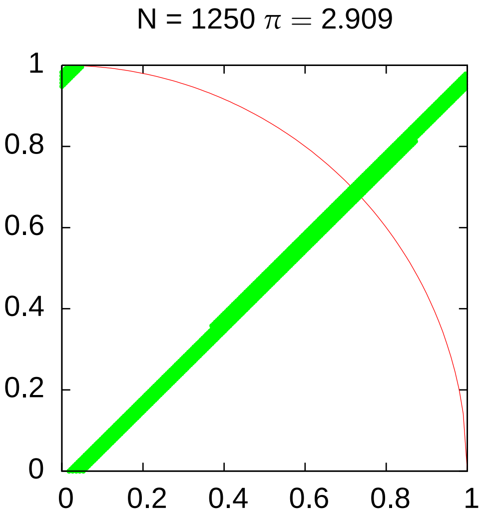
Die Bestimmung von π mit Halton funktioniert also.

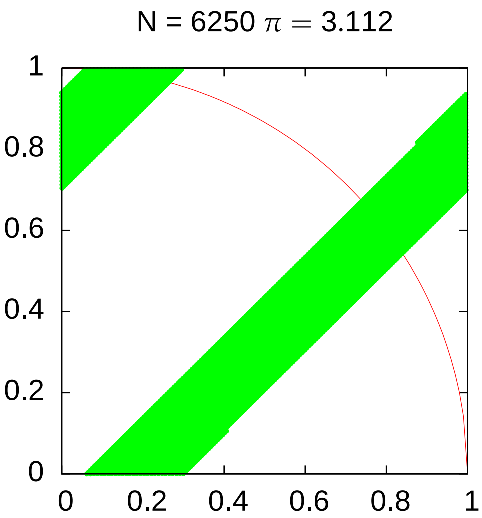
- Allerdings ist dieses Problem auch nur 2-dimensional!
- Um höherdimensionale Probleme zu lösen werden entsprechend größere Primzahlen benötigt
- Allerdings wird dadurch die Diskrepanz der Folgen immer größer, und damit die Konvergenz immer schlechter ...
- Beispiel: 50 Dimensionen
  $b_{49} = 227$ und $b_{50} = 229$ sind dann die größten Basen
- π bestimmen!

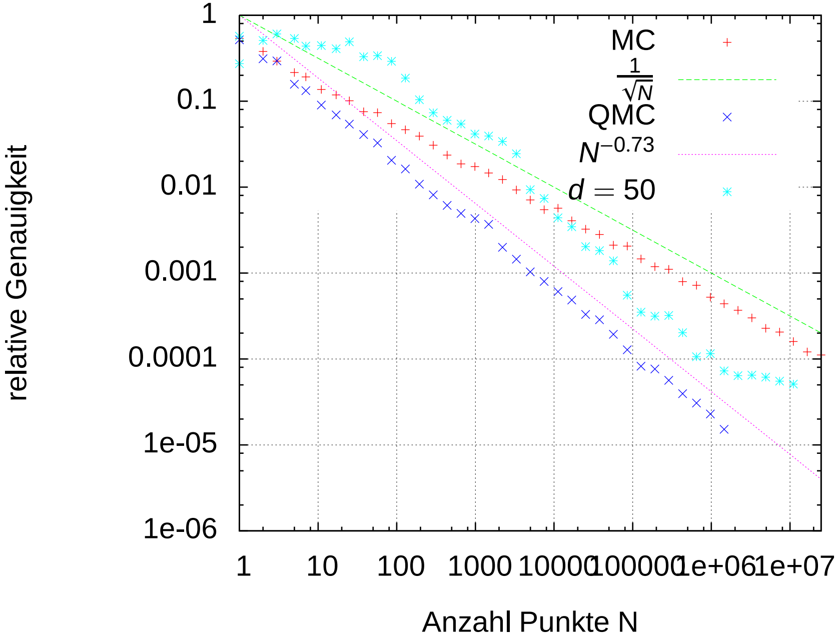

N = 10 $\pi = 4$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


N = 50 $\pi = 4$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


N = 250 $\pi = 2.96$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


N = 1250 $\pi = 2.909$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


N = 6250 $\pi = 3.112$
0
0.2
0.4
0.6
0.8
1
0
0.2
0.4
0.6
0.8
1



MC
$\frac{1}{\sqrt{N}}$
QMC
$N^{-0.73}$
$d = 50$
1
0.1
0.01
0.001
0.0001
1e-05
1e-06
relative Genauigkeit
1
10
100
1000
10000
100000
1e+06
1e+07
Anzahl Punkte N

Nachteile des Halton-Generators

- nicht für hochdimensionale Probleme geeignet

Nachteile des Halton-Generators

- nicht für hochdimensionale Probleme geeignet
- ausserdem nur relativ ineffizient auf Computern zu implementieren, da Berechnungen in Basen $b \neq 2$

Die Lösung - Sobol!

- geringe Diskrepanz auch bei vielen Dimensionen

Die Lösung - Sobol!

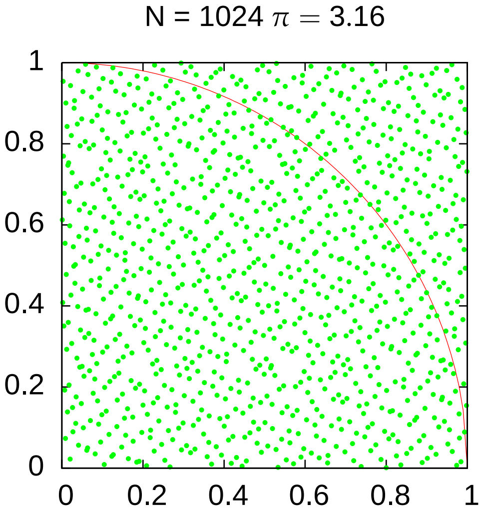
- geringe Diskrepanz auch bei vielen Dimensionen
- Berechnungen nur in einer Basis $b = 2$ nötig

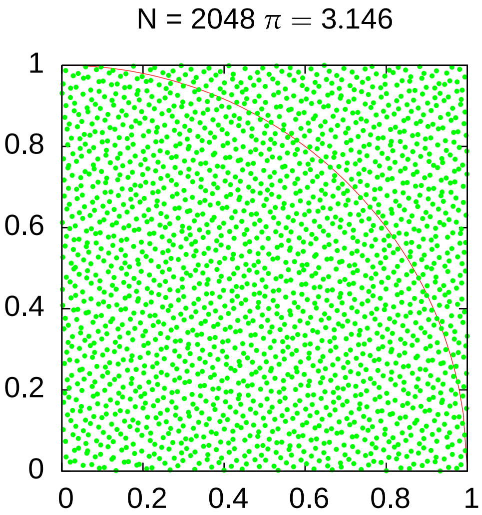
Die Lösung - Sobol!

- geringe Diskrepanz auch bei vielen Dimensionen
- Berechnungen nur in einer Basis $b = 2$ nötig

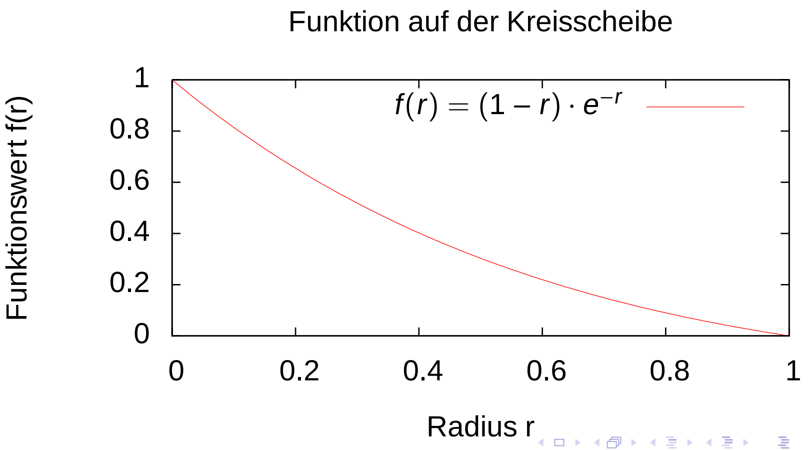
Die Lösung - Sobol!

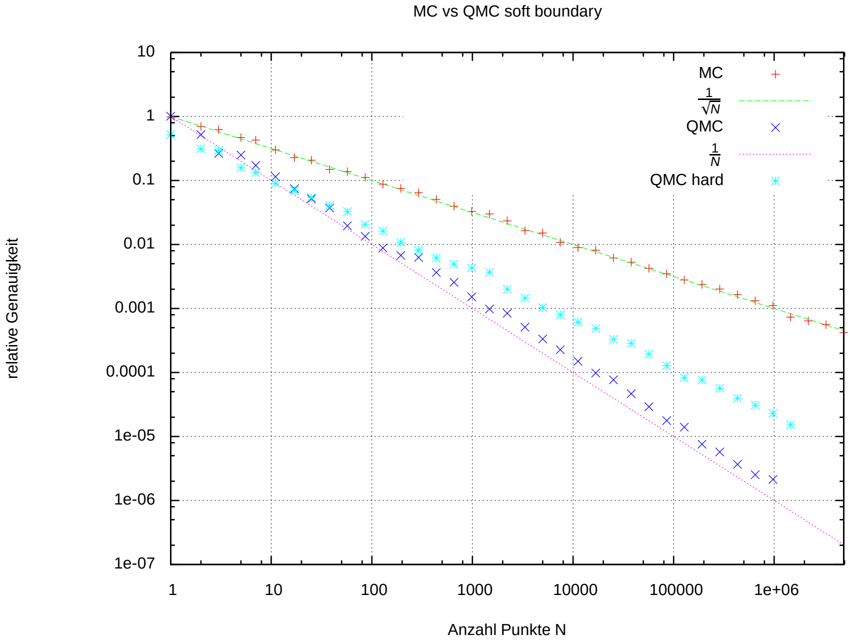
- geringe Diskrepanz auch bei vielen Dimensionen
- Berechnungen nur in einer Basis $b = 2$ nötig
  - daher schnelle Berechnung mit dem Computer
- allerdings: mathematische Basis nichttrivial


N = 1024 $\pi = 3.16$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


N = 2048 $\pi = 3.146$
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

Vergleich MC / QMC bei stetigem Anschluss am Rand





MC vs QMC soft boundary
MC
$\frac{1}{\sqrt{N}}$
QMC
$\frac{1}{N}$
QMC hard
relative Genauigkeit
Anzahl Punkte N
10
1
0.1
0.01
0.001
0.0001
1e-05
1e-06
1e-07
1
10
100
1000
10000
100000
1e+06

Zusammenfassung Quasi-MC

Vorteile

Zusammenfassung Quasi-MC

Vorteile

- wesentlich schnellere Konvergenz für Probleme mittlerer Dimensionalität

Zusammenfassung Quasi-MC

Vorteile

- wesentlich schnellere Konvergenz für Probleme mittlerer Dimensionalität
- besonders bei stetigem Anschluss am Rand!

Zusammenfassung Quasi-MC

Vorteile

- wesentlich schnellere Konvergenz für Probleme mittlerer Dimensionalität
- besonders bei stetigem Anschluss am Rand!

Nachteile

Zusammenfassung Quasi-MC

Vorteile

- wesentlich schnellere Konvergenz für Probleme mittlerer Dimensionalität
- besonders bei stetigem Anschluss am Rand!

Nachteile

- Dimensionsabhängigkeit

Zusammenfassung Quasi-MC

Vorteile

- wesentlich schnellere Konvergenz für Probleme mittlerer Dimensionalität
- besonders bei stetigem Anschluss am Rand!

Nachteile

- Dimensionsabhängigkeit
- schwierige Fehlerabschätzung

nur so ganz am Rande...

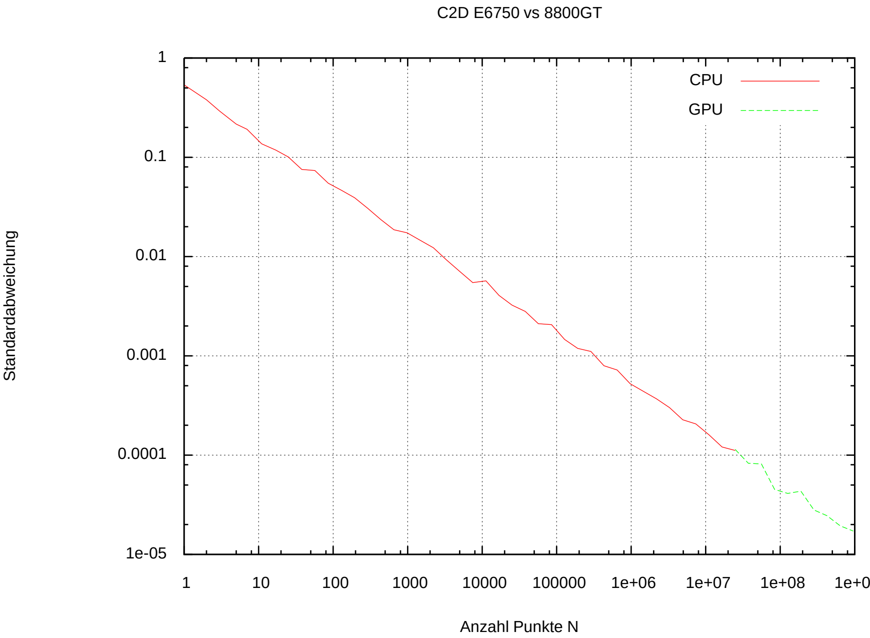
- Monte-Carlo Integration lässt sich sehr gut parallelisieren

nur so ganz am Rande...

- Monte-Carlo Integration lässt sich sehr gut parallelisieren
- gut geeignet für SIMD-Architekturen
  (**S**ingle **I**nstruction, **M**ultiple **D**ata)

nur so ganz am Rande...

- Monte-Carlo Integration lässt sich sehr gut parallelisieren
- gut geeignet für SIMD-Architekturen
  (**S**ingle **I**nstruction, **M**ultiple **D**ata)
  - wie z.B. Grafikkarten!


C2D E6750 vs 8800GT
Standardabweichung
1
0.1
0.01
0.001
0.0001
1e-05
1
10
100
1000
10000
100000
1e+06
1e+07
1e+08
1e+0
Anzahl Punkte N
CPU
GPU

nur so ganz am Rande...

- Monte-Carlo Integration lässt sich sehr gut parallelisieren
- gut geeignet für SIMD-Architekturen
  (**S**ingle **I**nstruction, **M**ultiple **D**ata)
  - wie z.B. Grafikkarten!
- 50x schneller!