

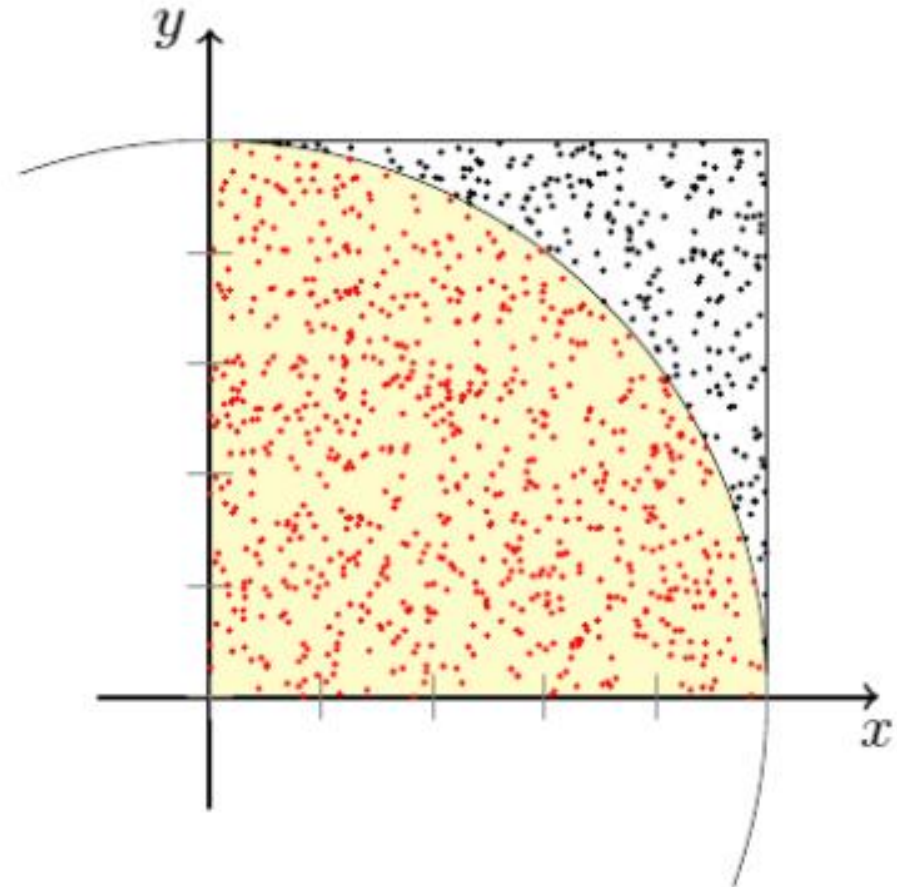


Monte-Carlo-Integration



Monte-Carlo-Integration

- Für eine numerische Integration wird die Monte-Carlo-Methode angewendet. Beispiel: Ein Viertel der Kreiszahl π wird angenähert, indem man innerhalb des Quadrats zufällig gewählter Punkte in den Viertelkreis zeichnet. Aufgrund des Gesetzes der großen Zahlen sinkt nähert man sich dem Ergebnis mit sehr vielen Punkte immer mehr an. π kann nun aufgrund des Verhältnisses der getroffenen zu den insgesamt Punkten mal 4 ermittelt werden.



Aufgabenstellung

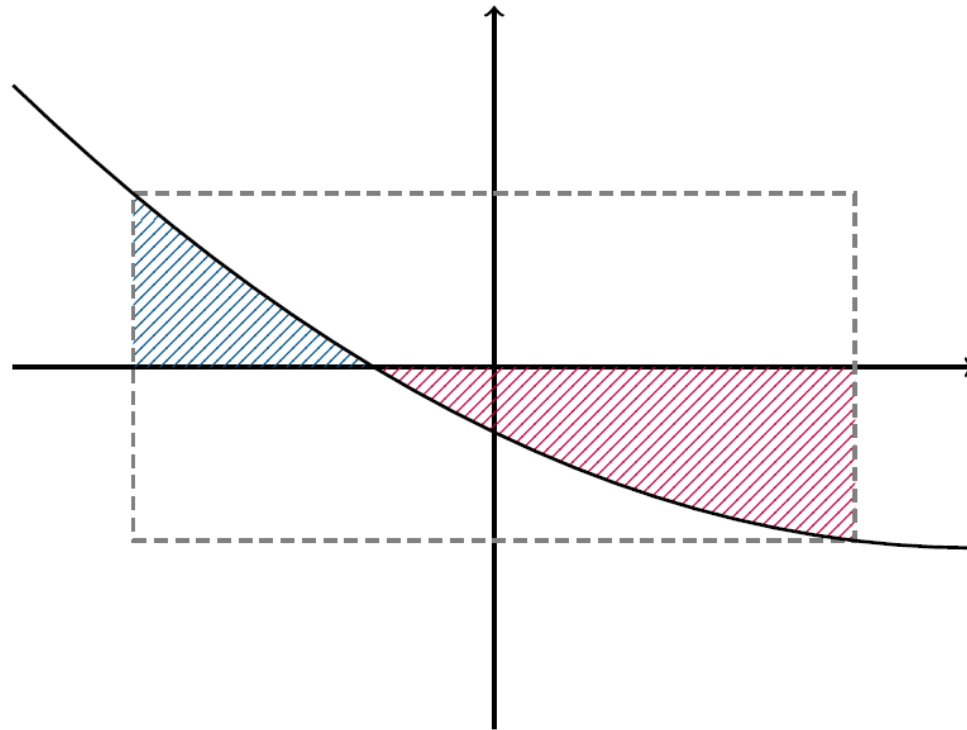
- Implementiere in der Klasse `MonteCarloImpl` das Interface `MonteCarlo`. Die Methode `computeIntegral` soll für eine übergebene Funktion (Klasse `Function`) das Integral im angegebenen Intervall $[a, b]$ mittels Monte-Carlo-Integration (analog zu der Beschreibung in der Vorlesung) parallel berechnen, indem zufällige Punkte ausgewählt und mit dem Funktionswert verglichen werden. Die Methode bekommt die Anzahl der zu verwendenden Threads und die Anzahl der zu betrachtenden Zufallspunkte (`iterations`) übergeben. Gehe davon aus, dass sich die Funktionswerte in dem gegebenen Intervall immer innerhalb der Grenzen $[\min Y, \max Y]$ bewegen.

Hinweise

- Die Zahl `iterations` gibt die Gesamtanzahl an Punkten an, die zufällig gesetzt werden sollen. Das heißt, dass diese Gesamtanzahl an Punkten auf die Threads aufgeteilt werden soll.
- Zum Zählweise der Punkte die „getroffen“ haben muss thread-sicher implementiert werden.
- Die Zufälligkeit muss mithilfe der Klasse `Random` von Java (`java.util.Random`) erzeugt werden. Einfach ein neues Objekt der Klasse `Random` erzeugen (Standardkonstruktor)
-> Methode `nextDouble()` für zufällige Dezimalzahlen von 0-1
Hier darf `Math.random()` nicht genutzt werden, denn dieser Methodenaufruf ist synchronisiert und somit könnte die parallele Implementierung langsamer sein als die sequentielle.

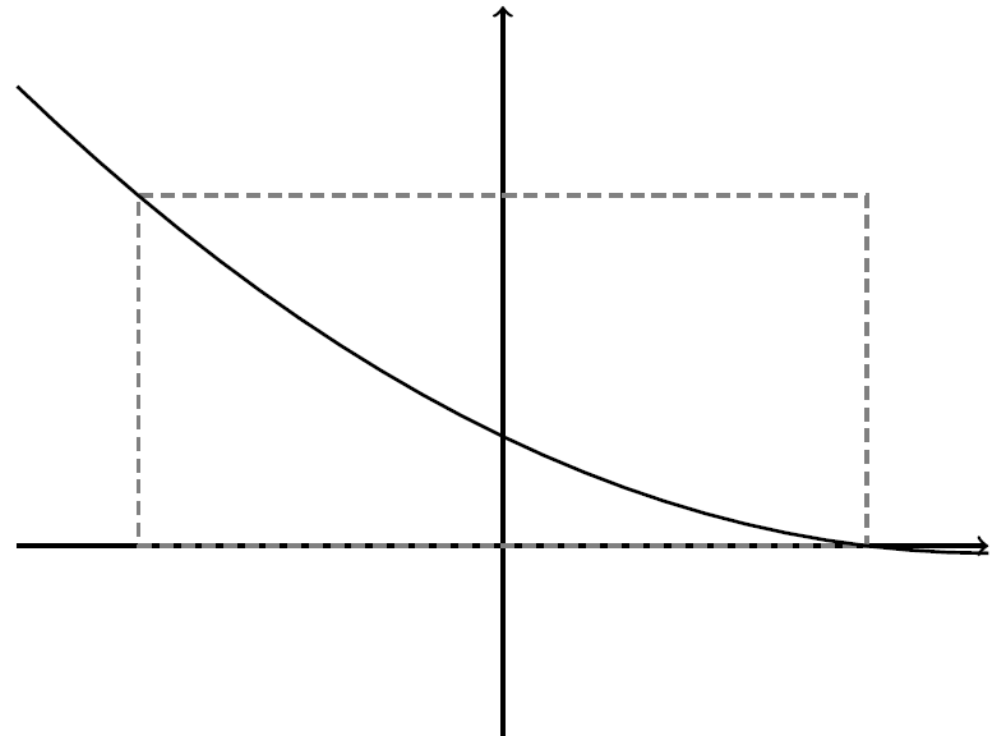
Hilfestellung

- Das Integral einer Funktion entspricht der Fläche zwischen dem Graphen der Funktion und der x-Achse. Hierbei zählen die Flächenstücke unterhalb der x-Achse negativ. Das Integral gibt somit die Flächenbilanz an: $\text{Integral} = \text{blaue Fläche} - \text{rote Fläche}$



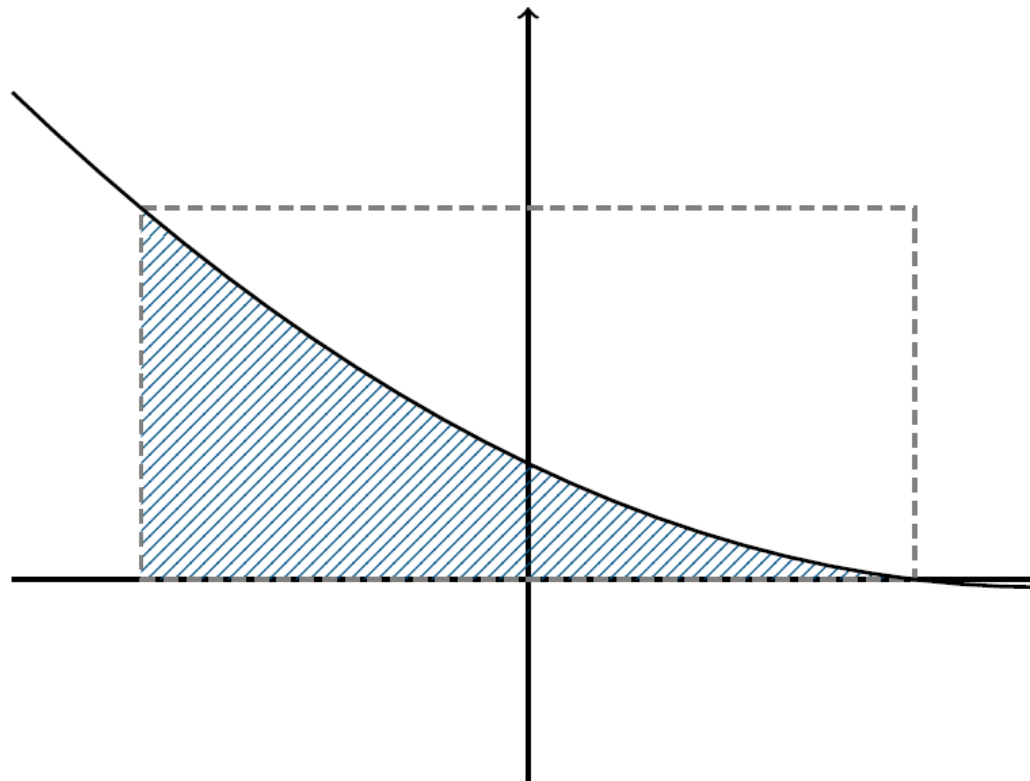
Hilfestellung

- Um das Ganze zu vereinfachen kann folgendermaßen vorgegangen werden: Zunächst wird allerdings das gesamte Rechteck gegeben durch die Intervalle $[a,b]$ und $[\min Y, \max Y]$ für die Zufallskoordinaten herangezogen, als sähe die Funktion so aus:
- Man „verschiebt“ somit die y-Achse entsprechend nach **oben oder unten** (hier: nach unten)



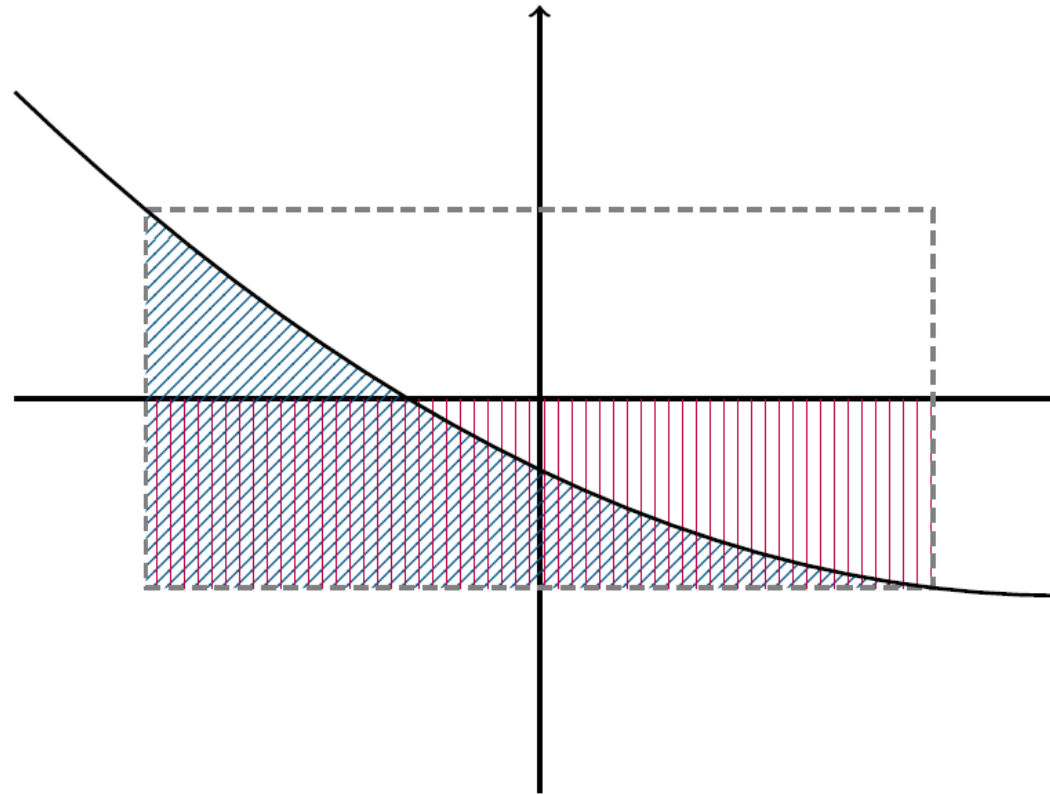
Hilfestellung

- Damit wird als Ergebnis (durch die parallele Berechnung) die Größe dieser Fläche bestimmt. -> „getroffene“ Punkte / Gesamtanzahl Punkte in Abhängigkeit der Größe der Intervalle $[a,b]$ und $[\min X, \min Y]$



Hilfestellung

- Das ist allerdings zu viel bzw. zu wenig. Daher muss das rot gestreifte Rechteck wieder abgezogen oder addiert (Fallunterscheidung anhand von $\min Y$) werden, um das Ergebnis zu korrigieren.



Hilfestellung

- Das Ergebnis ist dann das Integral der Ausgangsfunktion. Blaue Fläche minus rote Fläche.

