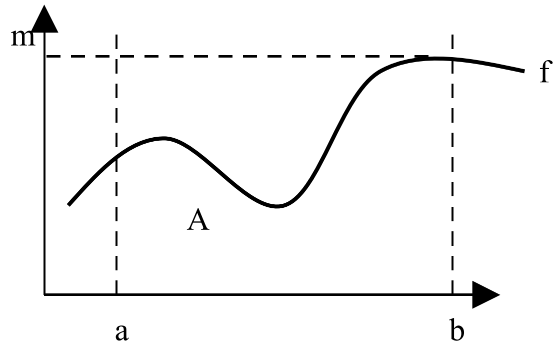
Monte-Carlo-Methode zur Approximation von Flächeninhalten (Integralen) .

Gegeben sind eine Funktion f und ein Intervall [a ; b], wobei b > a , f ≥ 0 und f stetig über dem gesamten Intervall vorausgesetzt werden (siehe Grafik). Das Maximum m von f über [a;b] sei berechenbar.



Das Ziel ist, den Inhalt A der zwischen G_f und x-Achse gelegenen Fläche durch „Zufallsregen" zu approximieren. Die Erzeugung des Zufallsregens ist eine stochastische Methode, die auch „Simulation" oder „Monte-Carlo-Methode" genannt wird.

<u>Vorgehensweise:</u>

Der Flächeninhalt A_m des durch a, b und m bestimmten Rechtecks wird berechnet.
Es gilt: $A_m = (b - a) \cdot m$
Es werden N Punkte $P(x_p;y_p)$ in diesem Rechteck per Zufallsgenerator erzeugt.
Diejenigen Punkte, die zwischen G_f und x-Achse landen, werden gezählt. Ihre Anzahl sei Z.
Sie sind ein Maß für den gesuchten Flächeninhalt A. Geht man nämlich von einer Gleichverteilung der Zufallspunkte aus, so verhalten sich die Punkte näherungsweise so wie die Flächen , d.h. : $\frac{A}{A_m} \approx \frac{Z}{N}$

Die Fläche A lässt sich daher durch $\frac{Z}{N} \cdot A_m$ bzw. $\frac{Z}{N} \cdot (b - a) \cdot m$ approximieren.
Dazu bedarf es allerdings sehr vieler Zufallspunkte, d.h. das N muss sehr groß gewählt werden, um vernünftige Ergebnisse zu erhalten .

<u>Es ergibt sich der folgende Algorithmus:</u>

1. Eingabe von f(x), a, b, N
2. Setze Z = 0 ; bisher noch kein Punkt gezählt
3. Wiederhole N-mal
   Setze x = a + (b – a) · Zufallszahl ; Zufallszahl zwischen 0 und 1 !
   Setze y = m · Zufallszahl
   Falls y ≤ f(x) gilt, dann erhöhe den Zähler Z um 1 ; y_p liegt unterhalb von f
4. Setze A = $\frac{Z}{N} \cdot (b - a) \cdot m$
5. Ausgabe von A

Dies lässt sich in ein Computerprogramm umsetzen.
Der Befehl zum Erzeugen einer Zufallszahl ist in vielen Computersprachen „random" oder „rand"(TI83) .
Falls wird meist mit „If " übersetzt.
Schwierig ist in der Regel die Maximumbestimmung. Hierfür bietet der TI83 jedoch eine fertige Funktion, die wie folgt angewandt wird: Y_1(fMax(Y_1,X,A,B)) STO M
<u>Achtung:</u> Der TI83 verwendet nur Großschreibung bei Variablen; daher muss die Fläche A aus dem Algorithmus ersetzt werden, etwa durch F .

TI83-BASIC-Programm MONTE (Monte-Carlo-Methode):
(Die Funktion $f \geq 0$ muss unter Y_1 eingegeben worden sein !)

```
Prompt A,B,N
Y1(fMax(Y1,X,A,B)) → M
997 → rand : 0 → Z
For(I,1,N)
A+(B-A)∗rand → X
M∗rand → Y
If Y≤Y1(X)
Z+1 → Z
End
Z/N∗(B-A)∗M → F
Disp "FLAECHE",F
```

Beispiele:

1) $f(x) = x^2$; [0 ;1] A = 1/3

N	100	200	500	1000	5000
A ≈	0,29	0,34	0,342	0,324	0,3304

2) $f(x) = \sqrt{9 - x^2}$; [0 ;3] $A = 2{,}25\pi \approx 7{,}069$

N	100	200	500	1000	5000
A ≈	6,21	7,34	7,146	7,182	7,025

Die Ergebnisse sind nicht besonders gut, wie man sieht. Bei N=5000 benötigt der TI83 Silver Edition etwa 8 min. !!

Jetzt noch das Programm MONTE2 mit grafischer Darstellung:

```
Prompt A,B,N
Y1(fMax(Y1,X,A,B)) → M
PlotsOff :FnOff :ClrDraw
95/63→V
If abs(B-A-M)>0.05
1→V
AV→Xmin :BV→Xmax
0→Ymin :M→Ymax
M/10→Yscl :(B-A)/10→Xscl
DrawF  Y1
Line(A,0,A,M) :Line(B,0,B,M)
Line(A,M,B,M)
997 → rand : 0 → Z
```

```
For(I,1,N)
A+(B-A)∗rand → X :M∗rand →
Y
Pt-On(X,Y)
If Y≤Y1(X)
Z+1 → Z
End
Z/N∗(B-A)∗M → F
Text(25,10, "A=",F)
```

Beispiel:

$f(x) = \sqrt{9 - x^2}$; [0 ; 3]