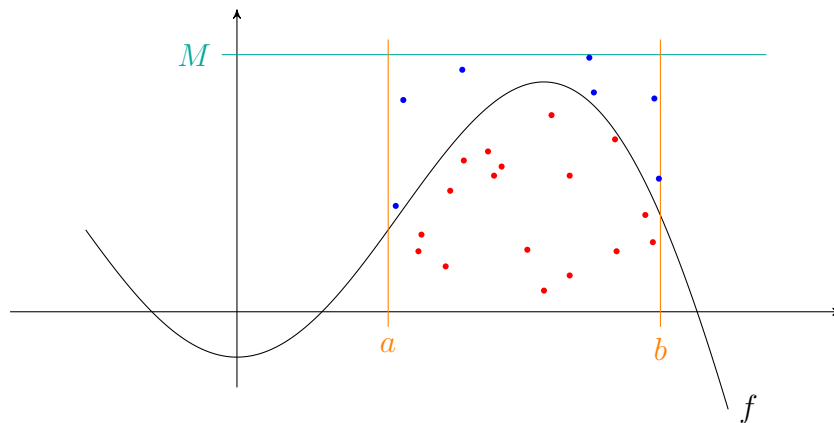


Schreiben Sie eine PYTHON-Funktion, die numerisch integriert, indem sie *zufällig* Punkte auswählt. Gehen Sie dabei folgendermaßen vor:

- (i) Ihre Funktion bekommt als Argumente den Integranden  $f$ , die Intervallgrenzen  $a$  und  $b$  und eine positive natürliche Zahl  $n$ .
- (ii) Der Einfachheit halber gehen wir davon aus, dass  $f$  auf  $[a, b]$  keine negativen Werte annimmt.
- (iii) Sie ermitteln zunächst zufällig  $\lceil \sqrt{n} \rceil$  Funktionswerte im Intervall  $[a, b]$ , vergrößern den größten so ermittelten Wert um zehn Prozent und nennen das Ergebnis  $M$ . ( $M$  ist dann eine grobe Approximation für eine obere Schranke von  $f$  auf  $[a, b]$ .)
- (iv) Nun wählen Sie zufällig  $n$  Punkte im Rechteck  $[a, b] \times [0, M]$  aus und ermitteln jeweils, ob der Punkt unterhalb von  $f$  liegt.
- (v) Berechnen Sie jetzt anhand des Verhältnisses von „Treffern“ zu Versuchen einen Näherungswert für das Integral.
- (vi) Verwenden Sie die Funktion `random.uniform` als Zufallszahlengenerator.



Zum Überprüfen Ihrer Funktion können Sie z.B. diese Integrale verwenden:

$$\int_0^\pi \sin x \, dx = 2 \qquad \int_0^1 e^x \, dx = e - 1$$

Wie gut sind Ihre Approximationen? Wie hängt die Qualität des Ergebnisses von  $n$  ab?