Całkownie

Ola Niedziela 329764

10.06.2022

Metody całkowania

Całki można sobie wyobrazić jako sumy nieskończenie wielu nieskończenie małych wartości, takich jak np. wartość funkcji pomnożona przez nieskończenie małą różniczkę jej zmiennej: f(x)dx (co znajduje odzwierciedlenie w podejściu Riemanna). Jest to określenie nieścisłe i nieformalne, choć używane w początkach rachunku całkowego przez G.W. Leibniza. Dziś ma ono znaczenie jedynie poglądowe i historyczne, natomiast poszczególne rodzaje całek są definiowane ściśle.

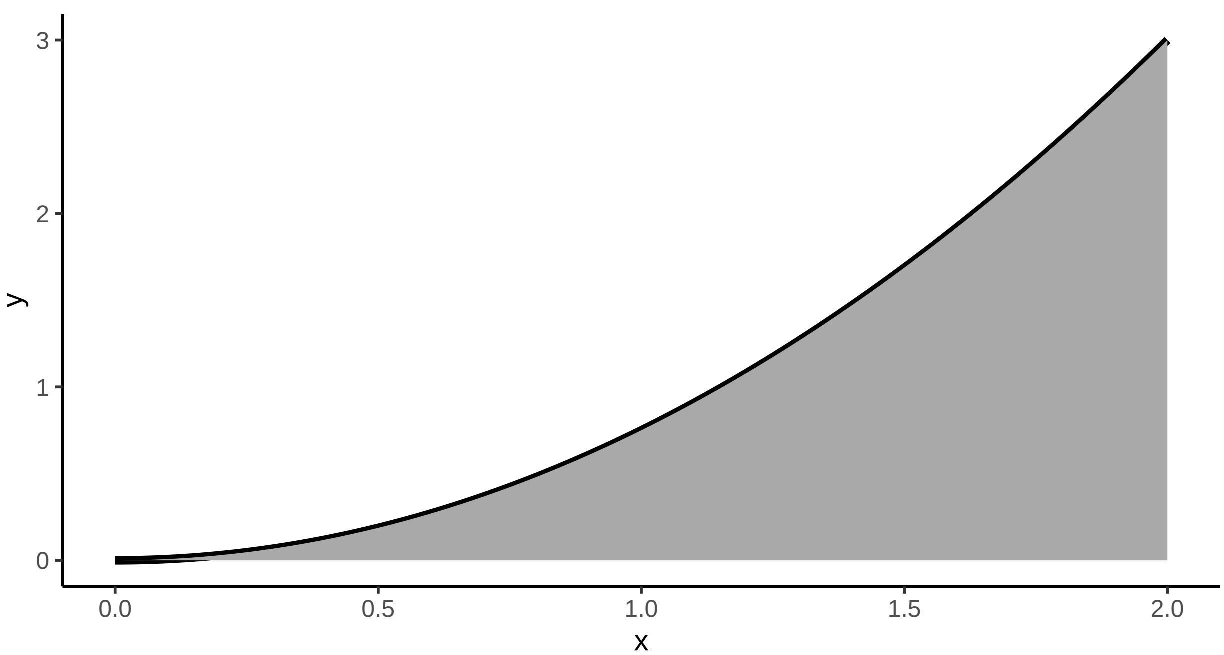
W tym raporcie chcę się przyjrzeć trzem metodom całkowania oraz ich dokładności. W tym celu zbadamy dwie funkcje: wielomian oraz funkcję wymierną. Skorzystam z poniższych metod całkowania:

1. Metoda prostokątów
2. Metoda trapezów
3. Metoda Monte Carlo

Funkcja 1.

Policzymy całkę z funkcji $f_1(x) = \frac{3}{4}x^2$ na przedziale [0,2]:

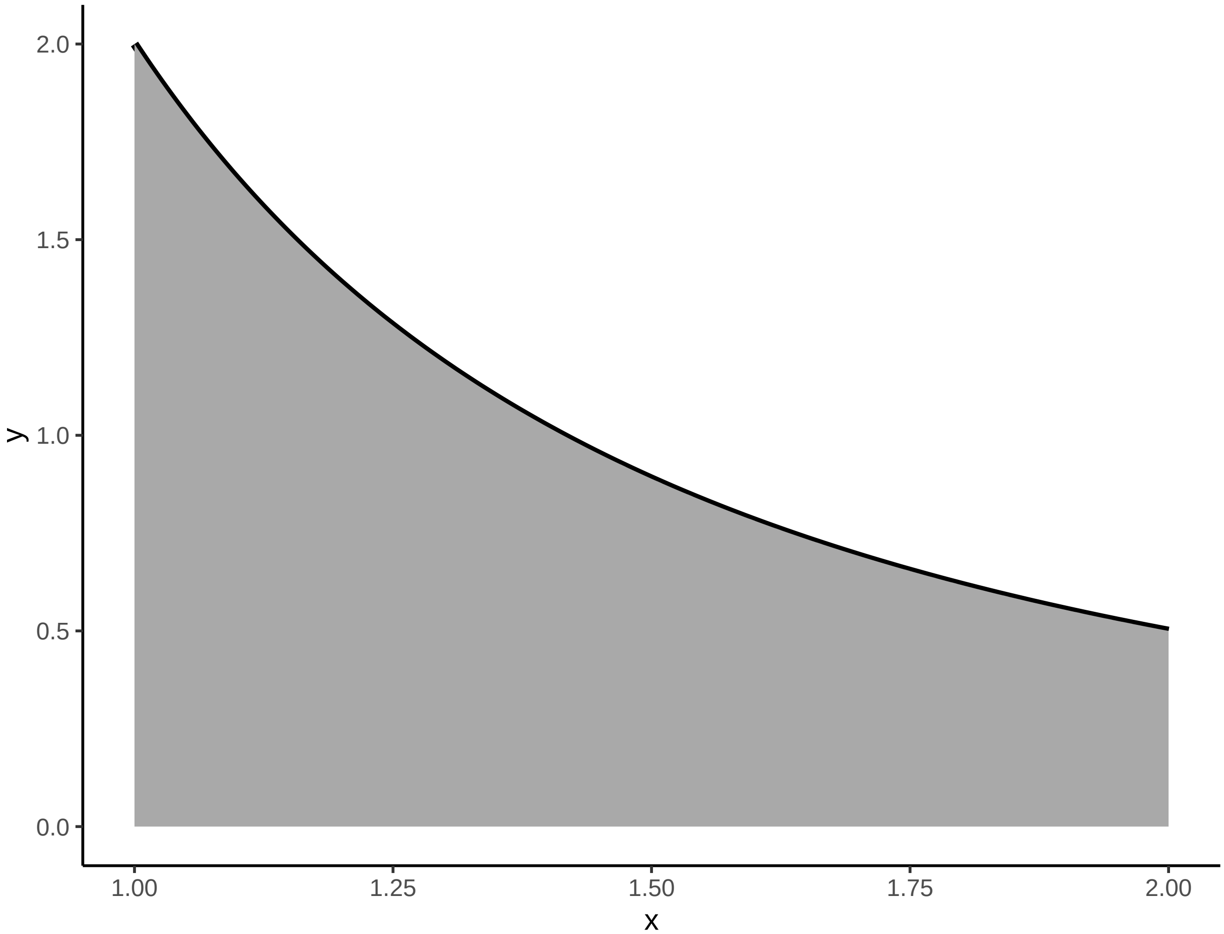
$$\int_0^2 \frac{3}{4}x^2\,dx = \frac{3}{4}\cdot\frac{1}{3}x^3\Big|_0^2 = \frac{1}{4}\cdot 2^3 - \frac{1}{4}\cdot 0^3 = 2 - 0 = 2$$



Funkcja 2.

Policzymy całkę z funkcji $f_2(x) = \frac{2}{x^2}$ na przedziale [1,2]:

$$\int_1^2 \frac{2}{x^2}\,dx = -\frac{2}{x}\bigg|_1^2 = -\frac{2}{2} - (-\frac{2}{1}) = -1 + 2 = 1$$



Policzyliśmy całkę przy użyciu twierdzenia Newtona - Leibniza, znanego również pod nazwą zasadnicze twierdzenie rachunku różniczkowego i całkowego. W dalszej części mojego raportu policzę powyższe całki za pomocą wcześniej wymienionych metod.

Metoda prostokątów

W metodzie prostokątów (ang. rectangular integration) korzystamy z definicji całki oznaczonej Riemanna , w której wartość całki interpretowana jest jako suma pól obszarów pod wykresem krzywej w zadanym przedziale całkowania $[x_p, x_k]$. Sumę tę przybliżamy przy pomocy sumy pól odpowiednio dobranych prostokątów:

Wybierzemy n takich prostokątów, że ich podstawą będzie część przedziału po którym całkujemy, czyli odcinek $[x_{i-1}, x_i]$, gdzie x_i to kolejne punkty podziału, a $i = 1, 2, \ldots, n$ oraz $x_0 = x_p, x_n = x_k$. Wysokością prostokąta będzie odcinek o długości $f(\frac{x_i + x_{i-1}}{2})$, czyli wartość funkcji w środkowym punkcie odcinka podziału.

$$\int_{x_p}^{x_k} f(x)\,dx \approx \sum_{i=1}^{n} (\frac{x_k - x_p}{n}) \cdot f(\frac{x_{i-1} + x_i}{2})$$

Funkcja 1

```
fun1 <- function(x){3/4 * x^2} #nasza funckja
n1 <- 10 #ilość prostokątów
xp1 <- 0 #początek przedziału
xk1 <- 2 #koniec przedziału
len1 <- (xk1 - xp1)/n1 #długość każdego z prostokątów
wek1 <- seq(xp1,xk1, by = len1) #wektor punktów podziału
area1 <- rep(0, n1) #wektor do którego będziemy zapisywać pola prostokątów
for (i in 1:n1){
  area1[i] <- len1 * fun1((wek1[i]+wek1[i+1])/2)
}
intpr1 <- sum(area1) #nasza całka, czyli suma pól prostokątów
```

Funkcja 2

```
fun2 <- function(x){2 / x^2} #nasza funckja
n2 <- 10 #ilość prostokątów
xp2 <- 1 #początek przedziału
xk2 <- 2 #koniec przedziału
len2 <- (xk2 - xp2)/n2 #długość każdego z prostokątów
wek2 <- seq(xp2,xk2, by = len2) #wektor punktów podziału
area2 <- rep(0, n2) #wektor do którego będziemy zapisywać pola prostokątów
for (i in 1:n2){
  area2[i] <- len2 * fun2((wek2[i+1]+wek2[i])/2)
}
intpr2 <- sum(area2) #nasza całka, czyli suma pól prostokątów
```

Po wykonaniu kodu otrzymujemy dla pierwszej funkcji:

$$\int_0^2 \frac{3}{4}x^2 \, dx \approx 1.995$$

oraz dla drugiej:

$$\int_1^2 \frac{2}{x^2} \, dx \approx 0.9985473$$

Metoda trapezów

Opisana wcześniej metoda prostokątów nie jest zbyt dokładna, ponieważ pola użytych w niej prostokątów źle odwzorowują powierzchnię pola pod krzywą (dokładność odwzorowania rośnie wraz ze wzrostem liczby prostokątów). Dużo lepszym rozwiązaniem jest zastosowanie zamiast nich trapezów o wysokości $[x_{i-1}, x_i]$ i podstawach równych odpowiednio wartości funkcji w punktach krańcowych. Sama zasada nie zmienia się. Mamy wtedy:

$$\int_{x_p}^{x_k} f(x) \, dx \approx \sum_{i=1}^{n} \frac{x_k - x_p}{n} \cdot \frac{f(x_{i-1}) + f(x_i)}{2}$$

Funkcja 1

```
fun1 <- function(x){3/4 * x^2} #nasza funckja
n1 <- 10 #ilość trapezów
xp1 <- 0 #początek przedziału
```

```
xk1 <- 2 #koniec przedziału
len1 <- (xk1 - xp1)/n1 #długość każdego z trapezów
wek1 <- seq(xp1,xk1, by = len1) #wektor punktów podziału
area1 <- rep(0, n1) #wektor do którego będziemy zapisywać pola trapezów
for (i in 1:n1){
  area1[i] <- len1 * (fun1(wek1[i])+fun1(wek1[i+1]))/2
}
intr1 <- sum(area1) #nasza całka, czyli suma pól trapezów
```

Funkcja 2

```
fun2 <- function(x){2/(x^2)} #nasza funckja
n2 <- 10 #ilość trapezów
xp2 <- 1 #początek przedziału
xk2 <- 2 #koniec przedziału
len2 <- (xk2 - xp2)/n2 #długość każdego z trapezów
wek2 <- seq(xp2,xk2, by = len2) #wektor punktów podziału
area2 <- rep(0, n2) #wektor do którego będziemy zapisywać pola trapezów
for (i in 1:n2){
  area2[i] <- len2 * (fun2(wek2[i])+fun2(wek2[i+1]))/2
}
intr2 <- sum(area2) #nasza całka, czyli suma pól trapezów
```

Po wykonaniu kodu otrzymujemy dla pierwszej funkcji:

$$\int_0^2 \frac{3}{4}x^2\,dx \approx 2.01$$

oraz dla drugiej:

$$\int_1^2 \frac{2}{x^2}\,dx \approx 1.0029103$$

Metoda Monte Carlo

Metoda ta polega na zliczaniu punktów wygenerowanych w spoób losowy, w pewnym prostokącie, które znajdują się wewnątrz obszaru ograniczonego wykresem funkcji. Wzór wygląda następująco:

$$\int_{x_p}^{x_k} f(x)\,dx \approx \frac{n_f}{n} \cdot P_{prostokąta}$$

,gdzie n_f to ilość punktów "pod wykresem funkcji", a n to liczba wszystkich punktów.

Funkcja 1

```
fun1 <- function(x){3/4 * (x^2)}
xp1 <- 0
xk1 <- 2
yp1 <- fun1(xp1)
yk1 <- fun1(xk1)
ymin1 <- min(c(yp1,yk1))
ymax1 <- max(c(yp1,yk1))
n2 <- 1000 #ilość punktów
wek11 <- runif(n1, xp1, xk1)
wek12 <- runif(n1, ymin1, ymax1)
```

```r
P1 <- abs((xk1 - xp1)*(yk1 - yp1))
nf1 <- 0
for(i in 1:n1){
  if(fun1(wek11[i]) >= wek12[i] ){
    nf1 <- nf1 + 1
  }
}
intmc1 <- nf1/n1 * P1
```

Funkcja 2

```r
fun2 <- function(x){2/(x^2)}
xp2 <- 1
xk2 <- 2
yp2 <- fun2(xp2)
yk2 <- fun2(xk2)
ymin2 <- min(c(yp2,yk2))
ymax2 <- max(c(yp2,yk2))
n2 <- 1000 #ilość punktów
wek21 <- runif(n2, xp2, xk2)
wek22 <- runif(n2, ymin2, ymax2)
P2 <- abs((xk2 - xp2)*(yk2 - yp2))
nf2 <- 0
for(i in 1:n2){
  if(fun1(wek21[i]) >= wek22[i] ){
    nf2 <- nf2 + 1
  }
}
intmc2 <- nf2/n2 * P2
```

Po wykonaniu kodu otrzymujemy dla pierwszej funkcji:

$$\int_0^2 \frac{3}{4}x^2\,dx \approx 2.4$$

oraz dla drugiej:

$$\int_1^2 \frac{2}{x^2}\,dx \approx 1.089$$

Podsumowanie

W tabeli poniżej przedstawiam zestawienie wyników całek liczonych poszczególnymi metodami:

	Metoda prostokątów	Metoda trapezów	Metoda Monte Carlo	Wartość rzeczywista
Funkcja 1	1.995	2.01	2.4	2
Funkcja 2	0.9985473	1.0029103	1.089	1

Wnioski

Metoda trapezów jest zdecydowanie dokładniejsza niż metoda prostokątów, jednak są one dosyć podobne, jeżeli chodzi o algorytm postępowania. Przy wzięciu raptem 10 punktów podziału otrzymaliśmy wartości całki bliskie oczekiwanym nam wartościom. Natomiast korzystając z metody Monte Carlo musimy wziąć zdecydowanie więcej punktów (ja wzięłam 1000), aby otrzymać dokładny wynik (i tak jest on mniej dokładny, niż pozostałe metody), co przekłada się na czas wykonywania kodu.