

RPiS - Egzamin część I

Karol Kęciński

April 15, 2021

Spis treści

1	Standardowy rozkład normalny	1
2	Dystrybuanta standardowego rozkładu normalnego	1
3	Złożony wzór trapezów	3
4	Metoda Romberga	3
5	Realizacja	4

1 Standardowy rozkład normalny

Standardowy rozkład normalny ma gęstość określoną wzorem:

$$f(x) = \frac{1}{\sqrt{2 \cdot \pi}} e^{\left(-\frac{x^2}{2}\right)}$$

2 Dystrybuanta standardowego rozkładu normalnego

Dystrybuanta standardowego rozkładu normalnego jest określona wzorem:

$$\Phi(t) = \int_{-\infty}^t \frac{1}{\sqrt{2 \cdot \pi}} e^{\left(-\frac{x^2}{2}\right)} dx$$

Ta całka nie ma przedstawienia za pomocą funkcji elementarnych. Skorzystamy z przekształceń aby uzasadnić zależność $\Phi(t)$ od $G(t)$, gdzie $G(t)$ dana jest wzorem*:

$$G(t) = \int_0^t e^{\left(-\frac{x^2}{2}\right)} dx$$

(*) oryginalny wzór na $G(t)$ [nazwijmy go $G_{org}(t)$] w treści zadania był całką od $-\infty$ zamiast od 0, ale jak pokażę niżej, taka definicja $G(t)$ ułatwi nam obliczenia. Wobec tego w dalszej części dokumentu przez $G(t)$ rozumie się podaną powyższym wzorem całkę.

Lemat 1. $\Phi(t) = 1 - \Phi(-t)$.

Dowód.

$$1 - \Phi(-t) = \int_{-\infty}^{+\infty} f(x) dx - \int_{-\infty}^{-t} f(x) dx = \int_{-t}^{+\infty} f(x) dx$$

Z parzystości funkcji $f(x)$:

$$\int_{-t}^{+\infty} f(x) dx = \int_{-\infty}^t f(x) dx$$

A zatem:

$$1 - \Phi(-t) = \int_{-t}^{+\infty} f(x) dx = \int_{-\infty}^t f(x) dx = \Phi(t)$$

□

Lemat 2. $\Phi(0) = \frac{1}{2}$.

Dowód. Z lematu 1:

$$\Phi(t) = 1 - \Phi(-t)$$

A zatem:

$$\Phi(0) = 1 - \Phi(0)$$

$$2 \cdot \Phi(0) = 1$$

$$\Phi(0) = \frac{1}{2}$$

□

Zauważmy teraz, że:

a) Dla $t \geq 0$

$$\begin{aligned} \Phi(t) &= \int_{-\infty}^0 f(x) dx + \int_0^t f(x) dx = \frac{1}{2} + \int_0^t f(x) dx = \\ &= \frac{1}{2} + \frac{1}{\sqrt{2 \cdot \pi}} G(t) \end{aligned}$$

b) Dla $t < 0$

$$\Phi(t) = \int_{-\infty}^0 f(x) dx - \int_0^{-t} f(x) dx =$$

Z parzystości $f(x)$:

$$\begin{aligned} &= \int_{-\infty}^0 f(x) dx - \int_0^t f(x) dx = \\ &= \frac{1}{2} - \frac{1}{\sqrt{2 \cdot \pi}} G(abs(t)) \end{aligned}$$

Nasz problem zredukował się do obliczenia wartości $G(t)$

$$G(t) = \int_0^t e^{-\frac{x^2}{2}} dx \tag{1}$$

a następnie pomnożenia otrzymanej wartości przez stałą $\frac{1}{\sqrt{2 \cdot \pi}}$.

3 Złożony wzór trapezów

Całkowanie numeryczne to metoda numeryczna polegająca na przybliżonym obliczaniu całek oznaczonych. Aby uzyskać przybliżoną wartość całki dzieli się przedział całkowania na podprzedziały. Ostateczny wynik jest sumą oszacowań całek w poszczególnych podprzedziałach.

Możemy przybliżyć wartość całki w przedziale $[a, b]$ obliczając pole trapezu o wierzchołkach a , b , $f(a)$, $f(b)$. Jeżeli podzielimy przedział całkowania na n podprzedziałów o długości h i w każdym z nich przybliżymy wartość całki obliczając pole trapezu, to suma tych pól będzie przybliżoną wartością całki.

$$\int_a^b f(x) dx \approx \sum_{i=0}^{n-1} \frac{h}{2} (f(x_i) + f(x_{i+1}))$$

gdzie

$$h = \frac{b-a}{n}, \quad x_i = a + ih$$

Upraszczając powyższe równanie otrzymamy:

$$\begin{aligned} \sum_{i=0}^{n-1} \frac{h}{2} (f(x_i) + f(x_{i+1})) &= \frac{h}{2} \sum_{i=0}^{n-1} (f(x_i) + f(x_{i+1})) = \\ &= \frac{a+b}{n} \sum_{i=0}^n {}'' f(x_i) \end{aligned}$$

gdzie symbol \sum'' oznacza, że pierwszy i ostatni wyraz sumy powinno się podzielić przez 2.

4 Metoda Romberga

Metoda Romberga to jedna z metod całkowania numerycznego. Można ją opisać rekurencyjnie

$$\begin{cases} R_{0,i} & : T_{2^i} = h_i \cdot \sum_{k=0}^{2^i-1} \left(\frac{f(x_k) + f(x_{k+1})}{2} \right) \\ R_{m,i} & : \frac{4^m \cdot R_{m-1,i+1} - R_{m-1,i}}{4^m - 1} \end{cases}$$

gdzie $R_{0,i} = T_{2^i}$ jest wzorem trapezów dla podziału przedziału na 2^i podprzedziałów. Dla zaoszczędzenia kosztów związanych z obliczaniem wyrazów $R_{0,i}$ skorzystamy z zależności rekurencyjnej:

$$R_{0,n} = \frac{1}{2} R_{0,n-1} + h_n \sum_{k=1}^{2^{n-1}} f(a + (2k-1)h_n)$$

5 Realizacja

Program obliczający wartość $\Phi(t)$ został zaimplementowany w języku Python.

Funkcja pomocnicza `romberg_method()` wyznacza kolejne elementy $R_{m,n}$ zapisywane jako elementy dwuwymiarowej listy `Romberg_storage[m][n]`. Korzystamy z nich do obliczenia kolejnych wartości $R_{m,n}$ dopóki różnica między $R_{m-1,0}$ a $R_{m,0}$ nie stanie się mniejsza niż $1 \cdot 10^{-8}$. Będzie to oznaczało, że osiągnęliśmy dokładność do ośmiu cyfr.

```
def Romberg(t):
    def romberg_method(m, n):
        if m != 0:
            new_el = (((4**m) * Romberg_storage[m-1][n+1]) - Romberg_storage[m-1][n]) / ((4 ** m) - 1)
            if n == 0:
                Romberg_storage.append([])
                pointer = Romberg_storage[m]
                pointer.append(new_el)
            elif n == 0:
                Romberg_storage[0][0] = ((G(0)+G(t)) * (t - 0)) / 2
            else:
                sum = 0
                for i in range(1, (2**(n-1))+1):
                    sum += G(0 + ((2 * i) - 1) * h)
                Romberg_storage[0].append((Romberg_storage[0][n-1] / 2) + (sum * h))

    Romberg_storage = [[0 for i in range(1)] for j in range(1)]
    h = t / 2

    romberg_method(0,0)
    romberg_method(0,1)
    romberg_method(1,0)

    m = 1

    while abs(Romberg_storage[m][0] - Romberg_storage[m-1][0]) > 0.00000001:
        h = h / 2
        m = m + 1

        for i in range(m + 1):
            romberg_method(0+i, m-i)

    return Romberg_storage[m][0]
```

Funkcja zwraca w wyniku ostatni obliczony wynik $R_{m,0}$, który stanowi wystarczająco dokładną aproksymację $\Phi(t)$.

Na potrzeby obliczeń w funkcji $Romberg(t)$, zostały zdefiniowane funkcje pomocnicze: $G(x)$ oraz $fi(t)$.

```
def G(x):  
    return m.exp(-(x**2)/2)  
  
def fi(t):  
    return Romberg(t) / m.sqrt(2 * m.pi)
```

Następnie utworzyłem nowy, lekko zmodyfikowany program tak, by generował wykres funkcji $\Phi(t)$:

