

Висша математика

втора част

Лабораторно упражнение No 6

Програмиране на крайни суми в Mathematica

1. Теоретична постановка на задачата

В това упражнение ще разгледаме различни подходи за сумиране чрез средствата на езика Mathematica. С цел представяне на конкретни програмни решения ще се фокусираме върху Римановите интегрални суми

$$I_n = \sum_{i=1}^n f(\xi_i) \Delta x_i$$

на функцията $f(x)$, чрез които се дефинира определеният интеграл

$$\int_a^b f(x) dx = \lim_{H \rightarrow 0} \sum_{i=1}^n f(\xi_i) \Delta x_i.$$

Да си припомним означенията свързани с конкретна Риманова интегрална сума

$[a, b]$ интеграционен интервал;

x_i делящи точки (наричат се още възли);

ξ_i е точка от i -тия подинтервал;

n брой на подинтервалите;

H максимална дължина на подинтервал.

В предното упражнение предположихме, че всички подинтервали имат равни дължини. Нека означим $h_i = \Delta x_i$, $i = 1, 2, \dots, n$ и вектор $h(h_1, h_2, \dots, h_n)$. Тук ще конструираме Риманови интегрални суми на базата на подинтервали чиито дължини h_i се определят чрез генератор на случайни числа. При направените означения делящите точки се получават от

$$x_i = a + \frac{L}{l} \sum_{k=1}^i h_k, \quad i = 1, 2, \dots, n, \quad x_0 = a \quad (1)$$

където $L = b - a$ и $l = \sum_{k=1}^n h_k$. Отново предполагаме, че всяка точка ξ_i е среда на подинтервал.

От теоретична гледна точка номерацията на възлите е от 0 до n . Дефинирайки делящите точки като координати на вектор се нуждаем от номерация на възлите от 1 до $n + 1$. Затова формула (1) се променя в

$$x_i = a + \frac{L}{l} \sum_{k=1}^{i-1} h_k, \quad i = 1, 2, \dots, n, n + 1.$$

Тук трябва да отбележим, че една сума се счита равна на нула, ако горната граница на сумационния индекс е по-малка от долната.

2. Различни програмни решения за представяне на сумата

$$I_n = \sum_{i=1}^n f(\xi_i) \Delta x_i$$

Предполагаме, че дялящите точки са записани във вектор x , а точките, в които пресмятаме функцията в x_i .

Първи начин

В Mathematica съществува оператор за сумиране Sum, който точно съответства на Римановата интегрална сума I_n . Чрез този оператор имаме:

$$RIS = \text{Sum}[f[xi[[i]]]h[[i]], \{i, 1, n\}];$$

Втори начин

Записваме функционалните стойности $f(\xi_i)$ във вектор F . Знакът за скаларното произведение на два вектора е $\cdot$. Тогава Риманова интегрална сума I_n се представя, чрез скаларно произведение на векторите F и h по следния начин:

$$F = \text{Table}\left[f\left[xi[[i]]\right], \{i, 1, n\}\right];$$

$$RIS = F \cdot h;$$

Трети начин (Почти класическо решение)

Ще реализираме сумата чрез оператор за цикъл с фиксирани граници. Даваме начална стойност на сумата $S = 0$. Към текущата стойност на S добавяме произведението отнасящо се за i -тия подинтервал.

$$S = 0;$$

$$\text{For}[i = 1, i \leq n, i++,$$

$$S = S + f[xi[[i]]]h[[i]]];$$

$$RIS = S;$$

Четвърти начин (Класическо решение)

При това решение данните са записани в масиви. Масивът е структура от данни, която много се различава от вектора. Два масива не могат да се умножават скаларно, два вектора обаче могат. Всички координати на даден вектор трябва да бъдат дефинирани. При масивите могат да се дефинират само някои елементи.

$$S = 0;$$

$$\text{For}[i = 1, i \leq n, i++,$$

$$S = S + f[xi[i]]h[i];$$

$$RIS = S;$$

3. Конкретно програмно решение

```
(* Пресмятане на Риманова интегрална сума
за конкретен интервал и подинтегрална функция *)
(* Дължините на подинтервали се определят
посредством генератор на случайни числа *)
(* Начално зареждане с данни *)
Clear[a, b, x, xi, n, f, RIS];
n = 100;
a = 0; b = Pi;
L = b - a;
h = Table[Random[], {i, 1, n}];
(* Сума на всички дължини *)
l = Sum[h[[i]], {i, 1, n}];
(* Делящи точки (възли) *)
(*  $\frac{L}{l}$  представлява мащабиращ коефициент *)
h =  $\frac{L}{l}$  h;
x = Table[a + Sum[h[[k]], {k, 1, i - 1}], {i, 1, n + 1}];
(* Точки  $\xi_i$   $i=1,2,\dots, n$  *)
(* в които се пресмята подинтегралната функция *)
xi = Table[(x[[i]] + x[[i + 1]]) / 2, {i, 1, n}];
(* Дефиниция на подинтегралната функция *)
f[x_] := x * Cos[x];
(* Пресмятане на Римановата интегрална сума *)
(* Сумирането се осъществява чрез оператора Sum *)
RIS = Sum[f[xi[[i]]] h[[i]], {i, 1, n}];
(* Пресмятане на определения интеграл *)
Int = Integrate[f[t], {t, 0, Pi}];
(* Изчисляване на абсолютна грешка получена при
заместване на интеграла с Римановата интегрална сума *)
Err = Abs[Int - RIS];
(* Блок за извеждане на резултати *)
(* Оператор N[v,k] показва че променливата
v ще се представи чрез десетична дроб с k цифри *)
Print["Стойност на Римановата интегрална сума In=", N[RIS, 14]];
Print["Стойност на определения интеграл Int[f[x],{x,a,b}]", N[Int]];
Print["Брой на подинтервалите=", n];
Print["Абсолютна грешка e_h=", N[Err]];

Стойност на Римановата интегрална сума In=-1.99984
Стойност на определения интеграл Int[f[x],{x,a,b}]=-2.
Брой на подинтервалите=100
Абсолютна грешка e_h=0.000156226
```

4. Задачи за изпълнение

В точка 3 е представено конкретно програмно решение на задачата за пресмятане на Риманова интегрална сума. Програмата е написана, като е използван първият начин за сумиране.

Напишете още три програми, във всяка от които се употребява различен начин на сумиране, като използвате описаните в точка 2.