

Отчёт по лабораторной работе 6

Елизавета Александровна Гайдамака

Содержание

Цель работы	3
Задание	4
Теоретическое введение	5
Выполнение лабораторной работы	6
Выводы	11

Цель работы

Целью данной работы является ознакомление с инструментами работы с пределами, последовательностями и интегралами в Octave.

Задание

- Частичные суммы
- Сумма ряда
- Вычисление интегралов
- Аппроксимирование суммами

Теоретическое введение

Octave - полноценный язык программирования поддерживающий множество типов циклов и условных операторов. Однако поскольку это векторный язык, многие вещи, которые можно было бы сделать с помощью циклов, можно векторизовать. Под векторизованным кодом мы понимаем следующее: вместо того чтобы писать цикл для многократной оценки функции мы сгенерируем вектор входных значений а затем оценим функцию с использованием векторного ввода. В результате получается код который легче читать и понимать и он выполняется быстрее благодаря эффективным алгоритмам для матричных операций.

Выполнение лабораторной работы

Посчитаем предел функции

$$f(x) = \left(1 + \frac{1}{n}\right)^n.$$

```

Command Window
@ (n) (1 + 1 ./ n) .^ n

>> k = [0:1:9]'
k =
    0
    1
    2
    3
    4
    5
    6
    7
    8
    9

>> format long
>> n = 10 .^ k
n =
         1
        10
       100
      1000
     10000
    100000
   1000000
  10000000
 100000000
1000000000

>> f(n)
ans =
2.0000000000000000
2.5937424601000002
2.704813829421529
2.716923932235520
2.718145926824356
2.718268237197528
2.718280469156428
2.718281693980372
2.718281786395798
2.718282030814509

>> format
>> |

```

Рис. 1: Рис.1

Посчитаем сумму от 2 до бесконечности для функции, а так же последовательность ее частичных сумм

$$a_n = \frac{1}{n(n+2)}.$$

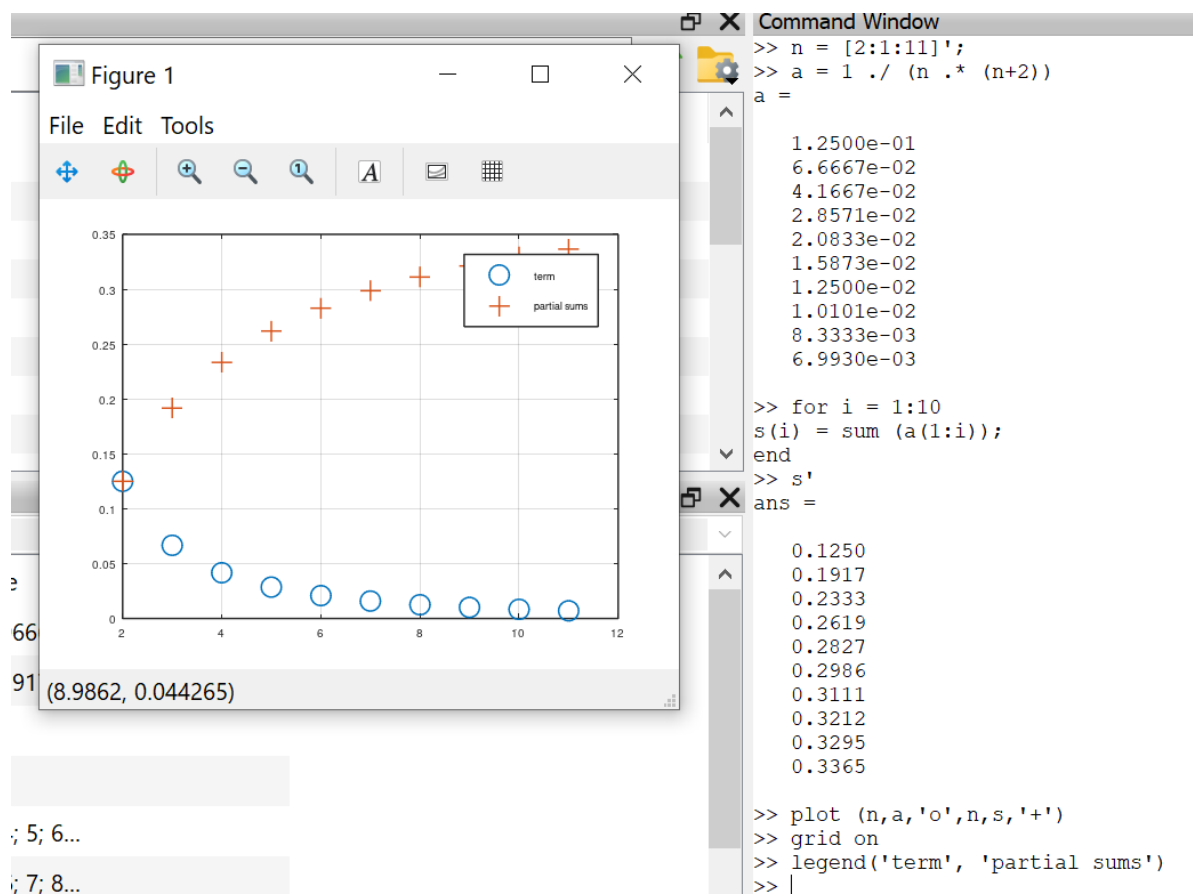


Рис. 2: Рис.2

Найдем сумму от 1 до 1000 функции

$$\frac{1}{n}.$$

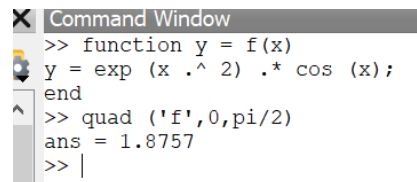
```
Command Window
>> n = [1:1:1000];
>> a = 1 ./ n;
>> sum (a)
ans = 7.4855
>> |
```

Рис. 3: Рис.3

Вычислим определенный интеграл

$$\int_0^{\pi/2} e^{x^2} \cos(x) dx.$$

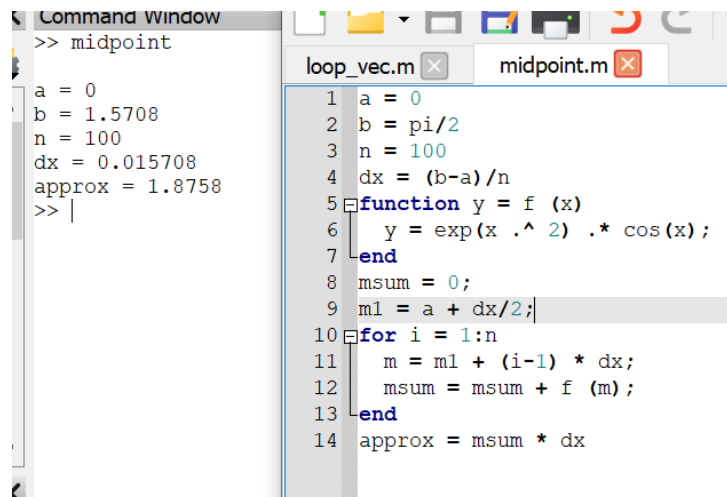
Будем использовать команду quad.



```
Command Window
>> function y = f(x)
y = exp (x .^ 2) .* cos (x);
end
>> quad ('f',0,pi/2)
ans = 1.8757
>> |
```

Рис. 4: Рис.4

Теперь вычислим этот же интеграл по правилу средней точки, напомним скрипт.



```
Command Window
>> midpoint
a = 0
b = 1.5708
n = 100
dx = 0.015708
approx = 1.8758
>> |

loop_vec.m
midpoint.m
1 a = 0
2 b = pi/2
3 n = 100
4 dx = (b-a)/n
5 function y = f (x)
6     y = exp(x .^ 2) .* cos(x);
7 end
8 msum = 0;
9 m1 = a + dx/2;
10 for i = 1:n
11     m = m1 + (i-1) * dx;
12     msum = msum + f (m);
13 end
14 approx = msum * dx
```

Рис. 5: Рис.5

Изменим код так, чтобы он был векторизованный.

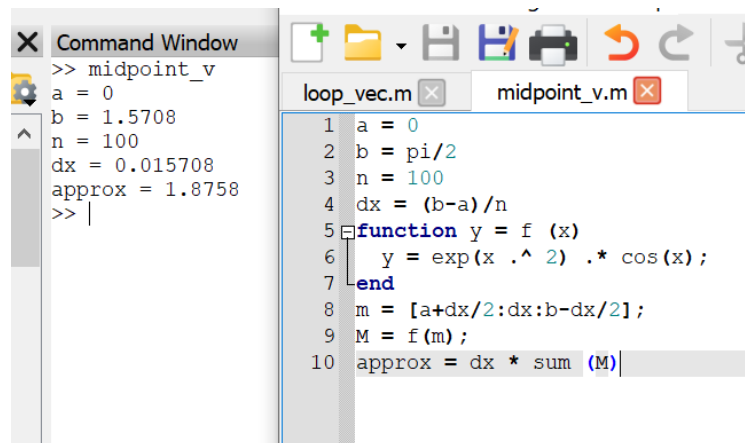


Рис. 6: Рис.6

Сравним время выполнения двух вариантов скриптов.

```
>> tic; midpoint; toc
a = 0
b = 1.5708
n = 100
dx = 0.015708
approx = 1.8758
Elapsed time is 0.00454593 seconds.
>> tic; midpoint_v; toc
a = 0
b = 1.5708
n = 100
dx = 0.015708
approx = 1.8758
Elapsed time is 0.00233197 seconds.
>>
```

Рис. 7: Рис.7

Выводы

Благодаря данной работе я ознакомилась с с инструментами работы с пределами, последовательностями и интегралами в Octave.