

Отчет по лабораторной работе №6

Пределы, последовательности и ряды

Ким Илья Владиславович

Содержание

Цель работы	4
Задание	5
Выполнение лабораторной работы	6
Выводы	17

Список иллюстраций

0.1	Анонимная функция и вычисление предела 1	7
0.2	Анонимная функция и вычисление предела 2	8
0.3	Определение вектора n и вычисление членов 1	9
0.4	Определение вектора n и вычисление членов 2	10
0.5	Определение вектора n и вычисление членов 3	11
0.6	Задание	11
0.7	Задание	12
0.8	Задание 2	12
0.9	Задание 2	13
0.10	Задание 3	13
0.11	Задание 3	14
0.12	Задание 4	15
0.13	Задание 4	16

Цель работы

Научиться использовать Octave с пределами, последовательностями и рядами.

Задание

1. Сделайте отчет по лабораторной работе в формате Markdown.
2. В качестве отчета просьба предоставить отчеты в 3 форматах: pdf, docx и md(в архиве, поскольку он должен содержать скриншоты, Markfile и т.д)

Выполнение лабораторной работы

Пределы, последовательности и ряды

1. Анонимная функция

```

>> f=@(n) (1 + 1 ./n) .^n
f =

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

>> k=[0:1:9]
k =

    0    1    2    3    4    5    6    7    8    9

>> k = [0:1:9]'
k =

    0
    1
    2
    3
    4
    5
    6
    7
    8
    9

>> format long

```

Рис. 0.1: Анонимная функция и вычисление предела 1

```

>> format long
>> n=10 .^ k
n =

         1
        10
       100
      1000
     10000
    100000
   1000000
  10000000
 100000000
1000000000

>> f(n)
ans =

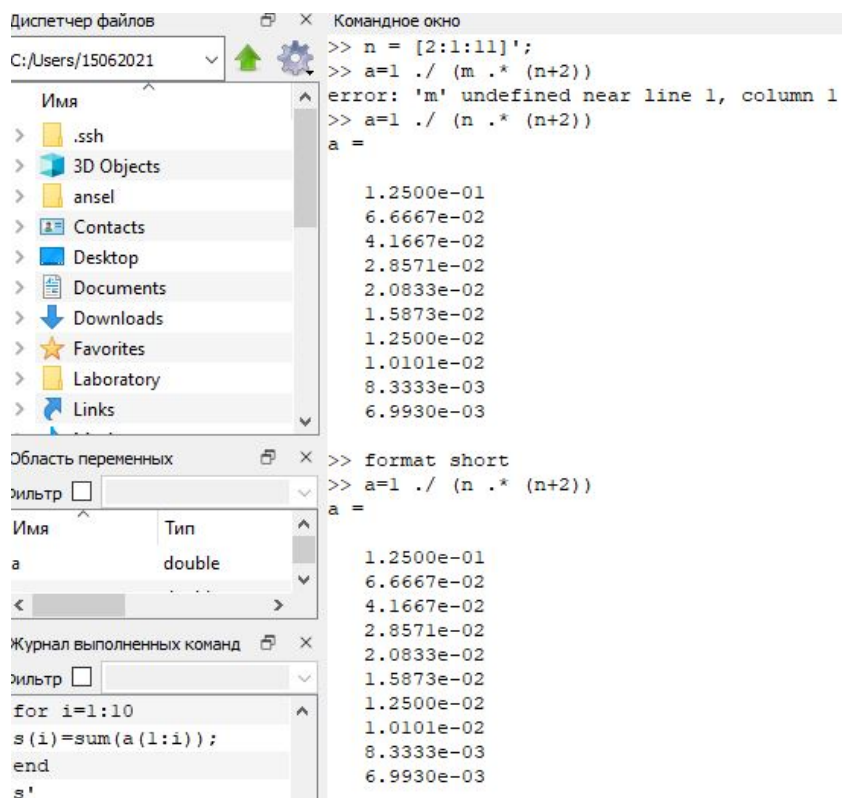
2.0000000000000000
2.593742460100002
2.704813829421529
2.716923932235520
2.718145926824356
2.718268237197528
2.718280469156428
2.718281693980372
2.718281786395798
2.718282030814509

>> format
>> |

```

Рис. 0.2: Анонимная функция и вычисление предела 2

2. Частичные суммы



The screenshot shows the MATLAB environment with the Command Window and Variable Editor. The Command Window displays the following commands and output:

```
>> n = [2:1:11]';  
>> a=1 ./ (m .* (n+2))  
error: 'm' undefined near line 1, column 1  
>> a=1 ./ (n .* (n+2))  
a =  
1.2500e-01  
6.6667e-02  
4.1667e-02  
2.8571e-02  
2.0833e-02  
1.5873e-02  
1.2500e-02  
1.0101e-02  
8.3333e-03  
6.9930e-03
```

The Variable Editor shows the variable `a` as a double array with the same values as the Command Window output.

The Command Window also shows the following commands and output:

```
>> format short  
>> a=1 ./ (n .* (n+2))  
a =  
1.2500e-01  
6.6667e-02  
4.1667e-02  
2.8571e-02  
2.0833e-02  
1.5873e-02  
1.2500e-02  
1.0101e-02  
8.3333e-03  
6.9930e-03
```

The Command Window also shows the following commands and output:

```
for i=1:10  
s(i)=sum(a(1:i));  
end  
s'
```

Рис. 0.3: Определение вектора `n` и вычисление членов `1`

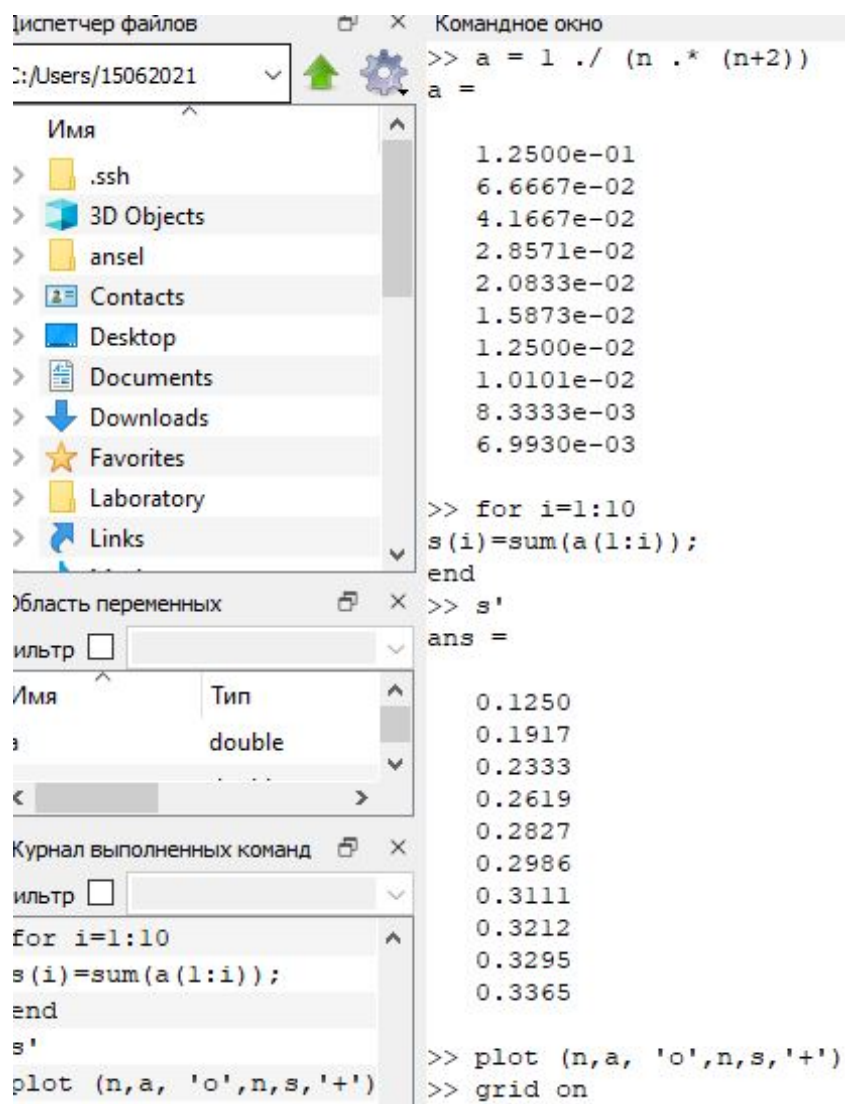


Рис. 0.4: Определение вектора n и вычисление членов 2

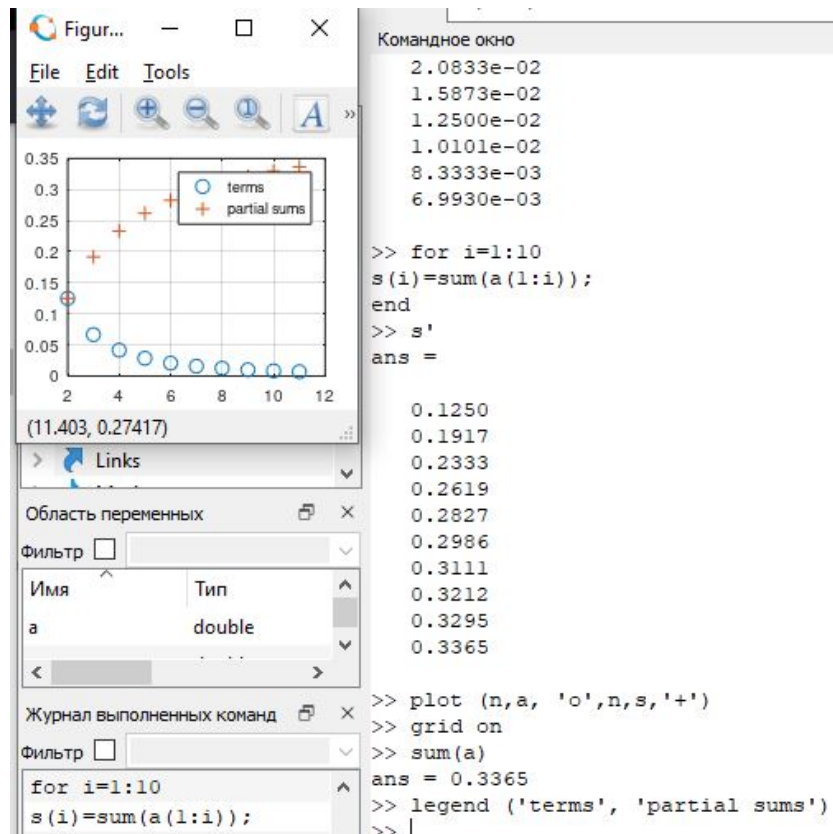


Рис. 0.5: Определение вектора n и вычисление членов z

3. Сумма ряда

Найдём сумму первых 1000 членов гармонического ряда:

$$\sum_{n=1}^{1000} \frac{1}{n}.$$

Нам нужно только сгенерировать члены как ряда вектор, а затем взять их сумму.

Рис. 0.6: Задание

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

Рис. 0.7: Задание

Численное интегрирование

4. Вычисление интегралов

Octave имеет несколько встроенных функций для вычисления определённых интегралов. Мы будем использовать команду `quad` (сокращение от слова квадратура).

Вычислим интеграл:

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

Синтаксис команды – `quad('f', a, b)`. Нам нужно сначала определить функцию.

Рис. 0.8: Задание 2

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

Рис. 0.9: Задание 2

5. Аппроксимирование суммами

Введите код в текстовом файле и назовите его `midpoint.m`.

```
% file 'midpoint.m'
% calculates a midpoint rule approximation of
% the integral from 0 to pi/2 of f(x) = exp(x^2) cos(x)
% — traditional looped code
% set limits of integration, number of terms and delta x
a = 0
b = pi/2
n = 100
dx = (b-a)/n
% define function to integrate
function y = f(x)
y = exp(x.^2).*cos(x);
end
msum = 0;
% initialize sum
m1 = a + dx/2; % first midpoint
% loop to create sum of function values
for i = 1:n
m = m1 + (i-1) * dx; % calculate midpoint
msum = msum + f(m); % add to midpoint sum
end
% midpoint approximation to the integral
approx = msum * dx
```

Он должен быть помещён в ваш рабочий каталог, а затем его можно запустить, набрав `midpoint` в командной строке.

Рис. 0.10: Задание 3

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

Рис. 0.11: Задание 3

Традиционный код работает хорошо, но поскольку Octave является векторным языком, также можно писать векторизованный код, который не требует каких-либо циклов.

Создадим вектор x -координат средних точек. Затем мы оцениваем f по этому вектору средней точки, чтобы получить вектор значений функции. Аппроксимация средней точки – это сумма компонент вектора, умноженная на Δx .

```
% file 'midpoint_v.m'
% calculates a midpoint rule approximation of
% the integral from 0 to pi/2 of f(x) = exp(x^2) cos(x)
% — vectorized code
% set limits of integration, number of terms and delta x
a = 0
b = pi/2
n = 100
dx = (b-a)/n
% define function to integrate
function y = f (x)
y = exp(x.^2) .* cos(x);
end
% create vector of midpoints
m = [a+dx/2:dx:b-dx/2];
% create vector of function values at midpoints
M = f(m);
% midpoint approximation to the integral
approx = dx * sum (M)
```

Рис. 0.12: Задание 4

```
>> midpoint_v
a = 0
b = 1.5708
n = 100
dx = 0.015708
approx = 1.8758
>> tic; midpoint; toc
a = 0
b = 1.5708
n = 100
dx = 0.015708
approx = 1.8758
Elapsed time is 0.00348496 seconds.
>> tic; midpoint_v; toc
a = 0
b = 1.5708
n = 100
dx = 0.015708
approx = 1.8758
Elapsed time is 0.00198817 seconds.
>> |
```

Рис. 0.13: Задание 4

Выводы

Научился использовать Octave с пределами, последовательностями и рядами.