

# **Отчёт по лабораторной работе №6**

**Дисциплина: Научное программирование**

Полиенко Анастасия Николаевна, НПМмд-02-23

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
3.1	Пределы, последовательности и ряды . . . . .	6
3.2	Численное интегрирование . . . . .	13
<b>4</b>	<b>Выводы</b>	<b>18</b>

## Список иллюстраций

3.1	Анонимная функция . . . . .	6
3.2	Индексная переменная . . . . .	7
3.3	Значения $n$ . . . . .	8
3.4	Значения функции . . . . .	9
3.5	Члены ряда . . . . .	10
3.6	Цикл . . . . .	11
3.7	Последовательность частичных сумм . . . . .	12
3.8	График . . . . .	13
3.9	Сумма ряда . . . . .	13
3.10	Вычисление интеграла . . . . .	14
3.11	Код <code>midpoint</code> . . . . .	15
3.12	Результат работы <code>midpoint</code> . . . . .	15
3.13	Код <code>midpoint_v</code> . . . . .	16
3.14	Результат работы <code>midpoint_v</code> . . . . .	16
3.15	Сравнение . . . . .	17

# 1 Цель работы

Изучить работу с пределами, последовательностями и рядами и численным интегрированием в GNU Octave.

## 2 Задание

1. Изучить работу с пределами, последовательностями и рядами
2. Изучить численное интегрирование

## 3 Выполнение лабораторной работы

### 3.1 Пределы, последовательности и ряды

1. Рассмотрим предел

$$\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n$$

Определим анонимную функцию (рис. 3.1).

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

Рис. 3.1: Анонимная функция

Создаём индексную переменную (рис. 3.2).

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

Рис. 3.2: Индексная переменная

С помощью индексной переменной определим значения  $n$  как степени 10 (рис. 3.3).

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

          1
         10
        100
       1000
      10000
     100000
    1000000
   10000000
  100000000
 1000000000
10000000000
```

Рис. 3.3: Значения  $n$

И посчитаем значения функции  $f(n)$  (рис. 3.4). Можем наблюдать, как они сходятся к числу  $e$ .



```
>> f(n)
ans =

2.0000000000000000
2.5937424601000002
2.704813829421529
2.716923932235520
2.718145926824356
2.718268237197528
2.718280469156428
2.718281693980372
2.718281786395798
2.718282030814509
```

Рис. 3.4: Значения функции

2. Посчитаем частичные суммы для ряда

$$\sum_{n=2}^{\infty} a_n, \text{ где } a_n = \frac{1}{n(n+2)}$$

Создаём индексную переменную и считаем члены ряда (рис. 3.5).

```

>> n = [2:1:11]'
n =

     2
     3
     4
     5
     6
     7
     8
     9
    10
    11

>> 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

```

Рис. 3.5: Члены ряда

В цикле считаем последовательность частичных сумм (рис. 3.6) и выведем их (рис. 3.7).

```
>> for i = 1:10
s(i) = sum(a(1:i))
end
s = 0.1250
s =
    0.1250    0.1917
s =
    0.1250    0.1917    0.2333
s =
    0.1250    0.1917    0.2333    0.2619
s =
    0.1250    0.1917    0.2333    0.2619    0.2827
s =
    0.1250    0.1917    0.2333    0.2619    0.2827    0.2986
s =
    0.1250    0.1917    0.2333    0.2619    0.2827    0.2986    0.3111
s =
    0.1250    0.1917    0.2333    0.2619    0.2827    0.2986    0.3111    0.3212
s =
    0.1250    0.1917    0.2333    0.2619    0.2827    0.2986    0.3111    0.3212    0.3295
s =
    0.1250    0.1917    0.2333    0.2619    0.2827    0.2986    0.3111    0.3212    0.3295    0.3365
```

Рис. 3.6: Цикл

```
>> s'  
ans =  
  
0.1250  
0.1917  
0.2333  
0.2619  
0.2827  
0.2986  
0.3111  
0.3212  
0.3295  
0.3365
```

Рис. 3.7: Последовательность частичных сумм

Выведем члены последовательности и частичные суммы на график (рис. 3.8).

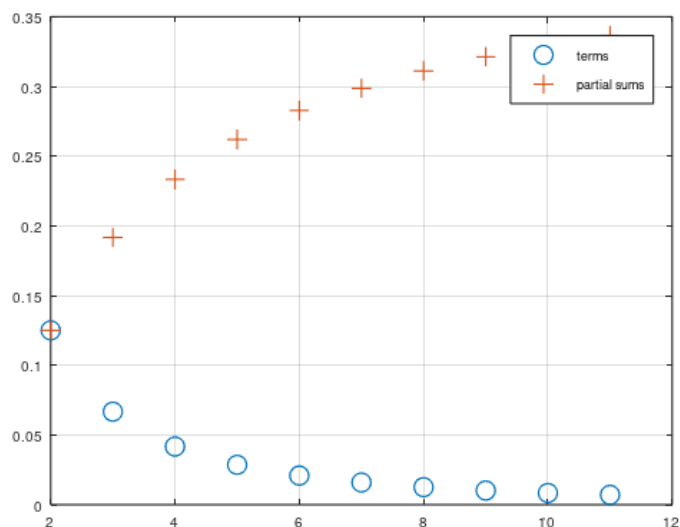


Рис. 3.8: График

### 3. Посчитаем сумму ряда

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

Создаём индексную переменную, генерируем на основе неё вектор значений и считаем их сумму (рис. 3.9).

```
n = [1:1:1000]
a = 1 ./ n
sum(a)

>> sum(a)
ans = 7.4855
>>
```

Рис. 3.9: Сумма ряда

## 3.2 Численное интегрирование

1. Создадим функцию и вычислим определённый интеграл с помощью встроенной команды `quad` (рис. 3.10).

```

>> function y = f(x)
y = exp(x.^2) .* cos(x)
end
>> quad('f',0,pi/2)
y = 1.3103
y = 1.0002
y = 0.2267
y = 1.0056
y = 0.9042
y = 1.0319
y = 1.4191
y = 1.1003
y = 1.5288
y = 1.2269
y = 1.3991
y = 1.0000
y = 0.039792
y = 1.0015
y = 0.5458
y = 1.0149
y = 1.2115
y = 1.0595
y = 1.5188
y = 1.1560
y = 1.4792
ans = 1.8757

```

Рис. 3.10: Вычисление интеграла

2. Напишем код для аппроксимирования суммами с циклами (рис. 3.11) и выведем результат (рис. 3.12); и векторами (рис. 3.13) и выведем результат

(рис. 3.14).

```
a = 0;  
b = pi/2;  
n = 100;  
dx = (b-a)/n;  
  
function y = f(x)  
    y = exp(x.^2) .* cos(x);  
end  
  
msum = 0;  
m1 = a + dx/2  
  
for i = 1:n  
    m = m1 + (i-1) * dx;  
    msum = msum + f(m);  
end  
  
approx = msum * dx
```

Рис. 3.11: Код midpoint

```
>> midpoint  
m1 = 7.8540e-03  
approx = 1.8758  
|
```

Рис. 3.12: Результат работы midpoint

```

a = 0;
b = pi/2;
n = 100;
dx = (b-a)/n;

function y = f(x)
    y = exp(x.^2) .* cos(x);
end

m = [a + dx/2:dx:b - dx/2];
M = f(m)
approx = dx * sum (M)

```

Рис. 3.13: Код midpoint\_v

```

>> midpoint_v
M =

Columns 1 through 9:
    1.000031    1.000278    1.000771    1.001512    1.002499    1.003734    1.005217    1.006947    1.008926

Columns 10 through 18:
    1.011154    1.013632    1.016359    1.019336    1.022564    1.026045    1.029777    1.033762    1.038001

Columns 19 through 27:
    1.042493    1.047241    1.052244    1.057502    1.063017    1.068788    1.074816    1.081101    1.087643

Columns 28 through 36:
    1.094441    1.101496    1.108807    1.116374    1.124194    1.132267    1.140591    1.149165    1.157985

Columns 37 through 45:
    1.167048    1.176362    1.185891    1.195662    1.205659    1.215876    1.226306    1.236941    1.247771

Columns 46 through 54:
    1.258788    1.269979    1.281332    1.292833    1.304466    1.316214    1.328057    1.339973    1.351940

Columns 55 through 63:
    1.363930    1.375916    1.387864    1.399740    1.411505    1.423117    1.434529    1.445691    1.456547

Columns 64 through 72:
    1.467034    1.477087    1.486632    1.495589    1.503870    1.511380    1.518015    1.523660    1.528192

Columns 73 through 81:
    1.531474    1.533360    1.533688    1.532283    1.528953    1.523491    1.515670    1.505243    1.491941

Columns 82 through 90:
    1.475473    1.455519    1.431733    1.403739    1.371125    1.333447    1.290218    1.240910    1.184949

Columns 91 through 99:
    1.121709    1.050510    0.970612    0.881210    0.781428    0.670312    0.546827    0.409843    0.258133

Column 100:
    0.090360

approx = 1.8758

```

Рис. 3.14: Результат работы midpoint\_v



Сравним два алгоритма по времени (рис. 3.15).

```
>> tic; midpoint; toc
m1 = 7.8540e-03
approx = 1.8758
Elapsed time is 0.00720716 seconds.
>> tic; midpoint_v; toc
M =

Columns 1 through 9:
    1.000031    1.000278    1.000771    1.001512    1.002499    1.003734    1.005217    1.006947    1.008926

Columns 10 through 18:
    1.011154    1.013632    1.016359    1.019336    1.022564    1.026045    1.029777    1.033762    1.038001

Columns 19 through 27:
    1.042493    1.047241    1.052244    1.057502    1.063017    1.068788    1.074816    1.081101    1.087643

Columns 28 through 36:
    1.094441    1.101496    1.108807    1.116374    1.124194    1.132267    1.140591    1.149165    1.157985

Columns 37 through 45:
    1.167048    1.176352    1.185891    1.195662    1.205659    1.215876    1.226306    1.236941    1.247771

Columns 46 through 54:
    1.258788    1.269979    1.281332    1.292833    1.304466    1.316214    1.328057    1.339973    1.351940

Columns 55 through 63:
    1.363930    1.375916    1.387864    1.399740    1.411505    1.423117    1.434529    1.445691    1.456547

Columns 64 through 72:
    1.467034    1.477087    1.486632    1.495589    1.503870    1.511380    1.518015    1.523660    1.528192

Columns 73 through 81:
    1.531474    1.533360    1.533688    1.532283    1.528953    1.523491    1.515670    1.505243    1.491941

Columns 82 through 90:
    1.475473    1.455519    1.431733    1.403739    1.371125    1.333447    1.290218    1.240910    1.184949

Columns 91 through 99:
    1.121709    1.050510    0.970612    0.881210    0.781428    0.670312    0.546827    0.409843    0.258133

Column 100:
    0.090360

approx = 1.8758
Elapsed time is 0.019454 seconds.
```

Рис. 3.15: Сравнение

## 4 Выводы

Научилась работе с системами линейных алгебраических уравнений в Octave.