

#РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ ##Факультет физико-математических и естественных наук ###Кафедра прикладной информатики теории вероятностей

ОТЧЕТ ПОЛАБОРАТОРНОЙ РАБОТЕ №6

ТЕМА: Задача на собственные значения

#####дисциплина: Научное программирование Студент: Хиссен Али Уэддей Группа: НПМмд-02-20 Ст. билет № 10322090306 Постановка задачи #####ВЫИСЛЕНИЕ ПРИДЕЛЫ **Литинг 1** Включеине журналирование сессии

>> diary on

вычислим предел данной последовательность в octave

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

Литинг 2

выводим функцию в octave

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

Литинг 3 Далее создаем вектор столбца который индикс начинается от 0 до 9 и потом возьмем n как $n=10^k$

```
octave:2> k=[0:10]
k =

    0
    1
    2
    3
    4
    5
    6
    7
    8
    9
   10

octave:3> format long
octave:4> n=10 .^ k
n =

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

Литинг 4

затем оценим значение $f(n)$ при увеличении n

```
octave:5> f(n)
ans =

 2.0000000000000000
 2.593742460100002
 2.704813829421529
 2.716923932235520
 2.718145926824356
 2.718268237197528
 2.718280469156428
 2.718281693980372
 2.718281786395798
 2.718282030814509
 2.718282053234788
```

не трудно заметить что при увеличении n , предел сходиться к конечному значению который приблизительно равно 2,7128

###вычисление частичные суммы

Литинг 5

выислим частичные суммы для следующее вырожение

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

для заданных значеных n получим :

```
octave:7> format long
octave:8> n=[2:1:11]';
octave:9> a =1 ./ (n .*(n+2))
a =

    1.2500000000000000e-01
    6.666666666666667e-02
    4.166666666666666e-02
    2.857142857142857e-02
    2.083333333333333e-02
    1.587301587301587e-02
    1.2500000000000000e-02
    1.010101010101010e-02
    8.333333333333333e-03
    6.993006993006993e-03
```

Литинг 6

вычисление частичных сумм

```
octave:16> for i=1:10
> s(i)=sum(a(1:i))
> end
```

Литинг 7

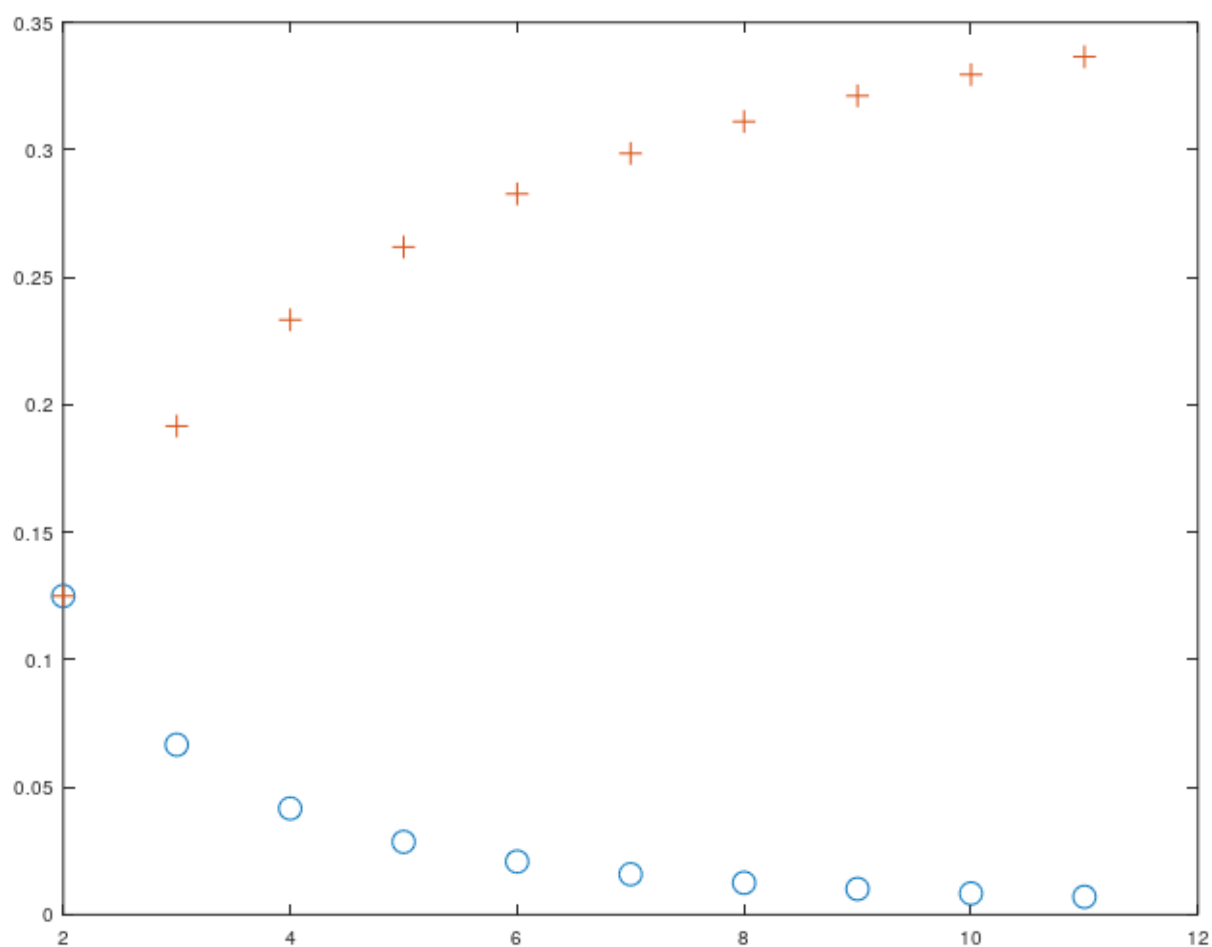
```
octave:17> s'
ans =

0.12500000000000000
0.19166666666666667
0.23333333333333333
0.261904761904762
0.282738095238095
0.29861111111111111
0.31111111111111111
0.321212121212121
0.329545454545455
0.336538461538462
```

Литинг 8

```
octave:18> plot(n,a,'o',n,s,'+')
1 2 3 4 5 6 7 8 9 10 11 12
```

Нарисуем график слагаемые и частичные суммы



###Вычисление суммы ряда вычислим сумму первые 1000 членов гармонического ряда.

```
octave:19> n=[1:1:1000];
octave:20> a=1./n;
octave:21> sum(a)
ans = 7.485470860550343
```

Литинг 9

##Численные интегрирование

вычисление интегралов

Литинг 10

вычислим интеграл от

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

для этого используем команда `quad('f',a,b)`. Определим функцию и затем вычислим интеграл.

```
octave:22> function y=f(x)
> y=exp (x .* cos(x));
> end
octave:23> quad('f',0,pi/2)
ans = 2.291186441180860
```

Аппроксимирование суммами

Литинг 11

для предидущего интеграла, вычислим интеграл по правило средней точки для $n=100$ для этого напишем следующий скрипт

```
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 nsum=0
9 m1=a+dx/2;
10 for i=1:n
11     m=m1 + (i-1) *dx;
12     nsum=nsum + f(m);
13 end
14 approx= nsum * dx
```

результат выполнения скрипта в командной строки

```
>> approx_integral
a = 0
b = 1.5708
n = 100
dx = 0.015708
nsum = 0
approx = 1.8758
```

напишем тот же код но без цикла так как octave является векторным языком

Литинг 12

```
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
9 m1=[a+dx/2:dx:b-dx/2];
10 M= f(m1);
11
12 approx= sum(M)* dx
13
14
```

```
>> approx_integral2
a = 0
b = 1.5708
n = 100
dx = 0.015708
approx = 1.8758
>> |
```

Литинг 13

сравним результаты измеряя время выполнения двух скриптов

```
>> tic;approx_integral;toc
a = 0
b = 1.5708
n = 100
dx = 0.015708
nsum = 0
approx = 1.8758
Elapsed time is 0.030364 seconds.
>> tic;approx_integral2;toc
a = 0
b = 1.5708
n = 100
dx = 0.015708
approx = 1.8758
Elapsed time is 0.00323391 seconds.
```

можно заметить что при использовании второй метод, скрипт работает достаточно быстро, т.е 10 быстрее.

Литинг 14

diary off

вывод: так мы познакомились с способом вычисления пределов,интегралов, частичные суммы ,суммы и аппроксимации в octave.