

Лабораторная работа №6

Пределы, последовательности, ряды и численное интегрирование в Octave

Демидова Е. А.

02 мая 2003

Российский университет дружбы народов, Москва, Россия

Информация

- Демидова Екатерина Алексеевна
- студентка группы НКНбд-01-21
- Российский университет дружбы народов
- <https://github.com/eademidova>



Введение

Научиться работать с пределами, последовательностями, рядами и выполнять численное интегрирование в Octave.

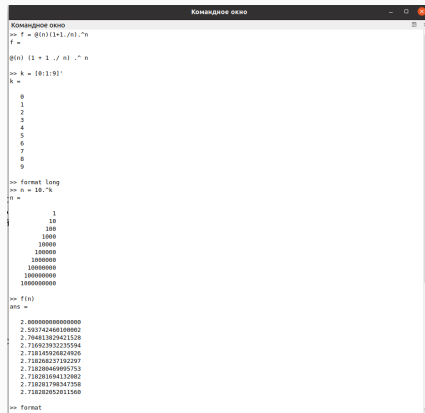
- Оценить предел.
- Найти частичные суммы.
- Найти сумму ряда.
- Вычислить интеграл встроенной функцией.
- Вычислить интеграл по правилу средней точки.

Выполнение лабораторной работы

Рассмотрим предел:

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

Предел сходится к значению, которое составляет приблизительно 2,71828...



```
Командное окно
>> 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

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

1
10
100
1000
10000
100000
1000000
10000000
100000000
1000000000

>> f(n)
ans =

2.800000000000000
2.593742460100002
2.704813829421526
2.716923932235594
2.718145926824926
2.718268237192297
2.718280469095753
2.718281694132682
2.718281798347356
2.718282052011560

>> format
```

Рис. 1: Оценка предела

Найдем частичные суммы ряда:

$$\sum_{n=2}^{\infty} \frac{1}{n(n+2)}$$

Частичные суммы

Для получения последовательности частичных сумм используем цикл и функцию `sum()`.

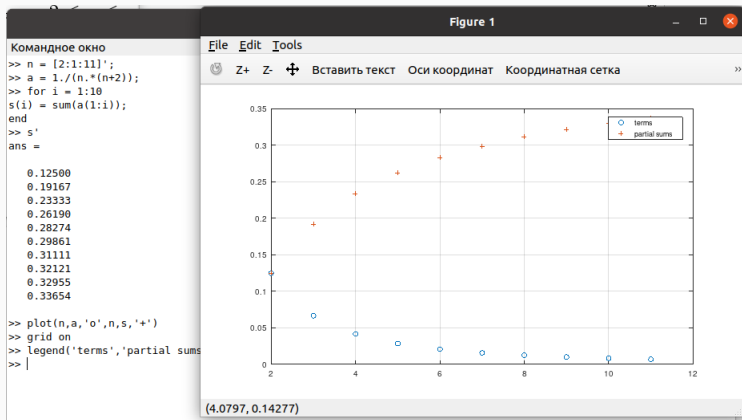


Рис. 2: Частичные суммы

Найдём сумму первых 1000 членов гармонического ряда:

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



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

Рис. 3: Сумма ряда

Вычислим интеграл:

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

Вычисление интеграла. Функция `quad()`

Используем команду `quad('f', a, b)`.

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

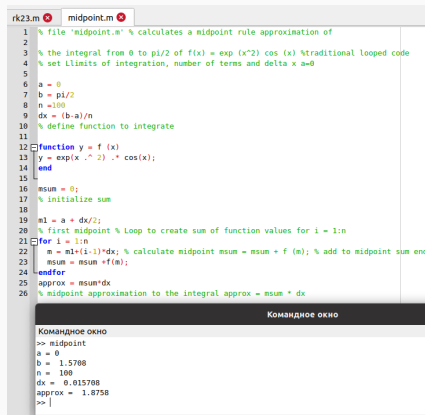
@(x) (exp (x .^ 2) .* cos (x))

>> quad(f, 0, pi/2)
error: 'x' undefined near line 2 column 9
error: called from
    f at line 2 column 3
>> quad('f', 0, pi/2)
ans = 1.8757
```

Рис. 4: Вычисление интеграла. Функция `quad()`

Вычисление интеграла по правилу средней точки.

Напишем скрипт, чтобы вычислить интеграл по правилу средней точки для $n = 100$.



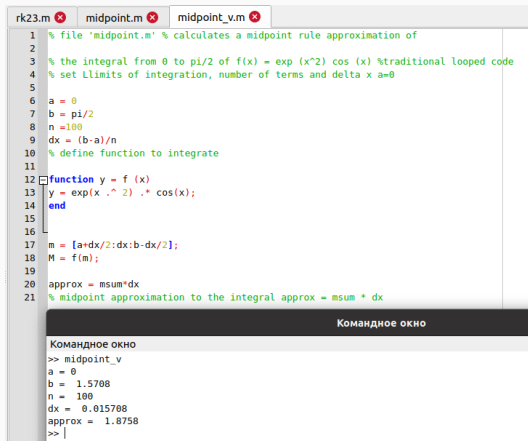
```
rk23.m midpoint.m
1 % file 'midpoint.m' % calculates a midpoint rule approximation of
2
3 % the integral from 0 to pi/2 of f(x) = exp (x^2) cos (x) %traditional looped code
4 % set limits of integration, number of terms and delta x a=0
5
6 a = 0;
7 b = pi/2;
8 n = 100;
9 dx = (b-a)/n;
10 % define function to integrate
11
12 function y = f (x)
13 y = exp(x.^2) .* cos(x);
14 end
15
16 msum = 0;
17 % initialize sum
18
19 m1 = a + dx/2;
20 % first midpoint % Loop to create sum of function values for i = 1:n
21 for i = 1:n
22     m = m1+(i-1)*dx; % calculate midpoint msum = msum + f (m); % add to midpoint sum end
23     msum = msum +f(m);
24 endfor
25 approx = msum*dx
26 % midpoint approximation to the integral approx = msum * dx
```

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

Рис. 5: Вычисление интеграла по правилу средней точки.

Вычисление интеграла по правилу средней точки.

Написали векторизированный код для вычисления интеграла по правилу средней точки.



```
rk23.m x midpoint.m x midpoint_v.m x
1 % file 'midpoint.m' % calculates a midpoint rule approximation of
2
3 % the integral from 0 to pi/2 of f(x) = exp(x^2) cos(x) %traditional looped code
4 % set Llimits of integration, number of terms and delta x a=0
5
6 a = 0
7 b = pi/2
8 n = 100
9 dx = (b-a)/n
10 % define function to integrate
11
12 function y = f(x)
13 y = exp(x.^2) .* cos(x);
14 end
15
16
17 m = [a+dx/2:dx:b-dx/2];
18 M = f(m);
19
20 approx = msum*dx
21 % midpoint approximation to the integral approx = msum * dx
```

Командное окно

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

Рис. 6: Вычисление интеграла по правилу средней точки. Векторизированный код.

Вычисление интеграла по правилу средней точки.

Сравним время выполнения для каждой реализации

```
>> tic;midpoint;toc  
a = 0  
b = 1.5708  
n = 100  
dx = 0.015708  
approx = 1.8758  
Elapsed time is 0.00658202 seconds.  
>> tic;midpoint_v;toc  
a = 0  
b = 1.5708  
n = 100  
dx = 0.015708  
approx = 1.8758  
Elapsed time is 0.000570059 seconds.  
>> |
```

Рис. 7: Сравнение времени реализаций вычисления интеграла по правилу средней точки

Заключение

В результате выполнения работы научились работать с пределами, последовательностями, рядами и выполнять численное интегрирование в Octave.

1. GNU Octave [Электронный ресурс]. Free Software Foundation, 2023. URL: <https://octave.org/>.
2. GNU Octave Documentation [Электронный ресурс]. Free Software Foundation, 2023. URL: <https://docs.octave.org/latest/>.