

Лабораторная работа №6

**Пределы, последовательности, ряды и численное интегрирование в
Octave**

Демидова Екатерина Алексеевна

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лабораторной работы	8
5	Выводы	14
	Список литературы	15

Список иллюстраций

4.1	Оценка предела	9
4.2	Частичные суммы	10
4.3	Сумма ряда	10
4.4	Вычисление интеграла. Функция <code>quad()</code>	11
4.5	Вычисление интеграла по правилу средней точки.	12
4.6	Вычисление интеграла по правилу средней точки. Векторизиро- ванный код.	13
4.7	Сравнение времени реализаций вычисления интеграла по правилу средней точки	13

1 Цель работы

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

2 Задание

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

3 Теоретическое введение

Дадим определение GNU Octave. GNU Octave — свободная программная система для математических вычислений, использующая совместимый с MATLAB язык высокого уровня [1].

На официальном сайте Octave даётся следующая характеристика этого научного языка программирования[]:

- Мощный синтаксис, ориентированный на математику, со встроенными инструментами 2D/3D-графики и визуализации.
- Бесплатное программное обеспечение, работающее на GNU/Linux, macOS, BSD и Microsoft Windows.
- Вставка, совместимая со многими скриптами Matlab

Приведём некоторые примеры использования Octave[2]:

1. Решение систем уравнений с помощью операций линейной алгебры над векторами и матрицами.

```
b = [4; 9; 2] # Column vector
A = [ 3 4 5;
     1 3 1;
     3 5 9 ]
x = A \ b      # Solve the system Ax = b
```

2. Визуализация данных с помощью высокоуровневых графических команд в 2D и 3D.

```
x = -10:0.1:10; # Create an evenly-spaced vector from -10..10
y = sin (x);    # y is also a vector
plot (x, y);
title ("Simple 2-D Plot");
xlabel ("x");
ylabel ("sin (x)");
```

4 Выполнение лабораторной работы

Рассмотрим предел:

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

Оценим это выражение. Для этого определим анонимную функцию f равную выражению, предел которого мы ищем. Затем создадим индексную переменную из целых чисел от 0 до 9 и возьмём степени 10, которые будут входными значениями. Наконец оценим $f(n)$ и получим, что предел сходится к значению, которое составляет приблизительно 2,71828... (рис. [4.1])


```

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 =

 2.000000000000000
 2.593742460100002
 2.704813829421528
 2.716923932235594
 2.718145926824926
 2.718268237192297
 2.718280469095753
 2.718281694132082
 2.718281798347358
 2.718282052011560

>> format

```

Рис. 4.1: Оценка предела

Найдем частичные суммы ряда:

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

Определим индексный вектор от 2 до 11, а затем вычислим члены суммы. Для получения последовательности частичных сумм используем цикл и функцию `sum()`. Затем отобразим слагаемые и частичные суммы на графике (рис. [4.2]).

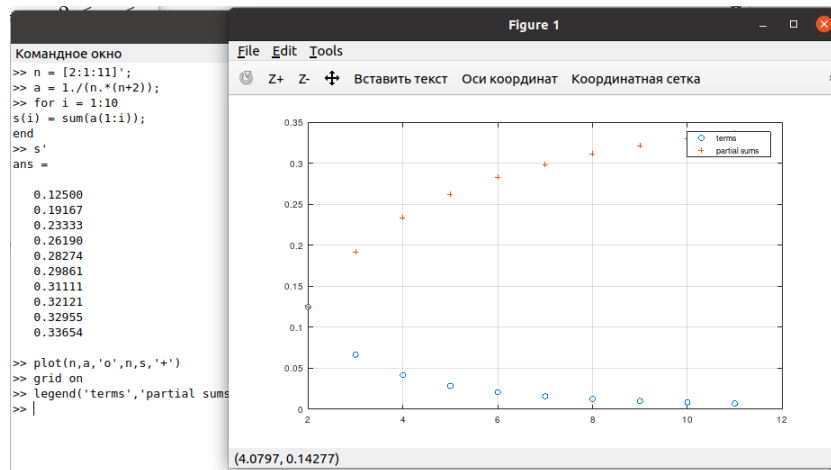


Рис. 4.2: Частичные суммы

Найдём сумму первых 1000 членов гармонического ряда:

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

Для этого сгенерируем члены ряда как вектор и возьмём их сумму (рис. [4.3]).



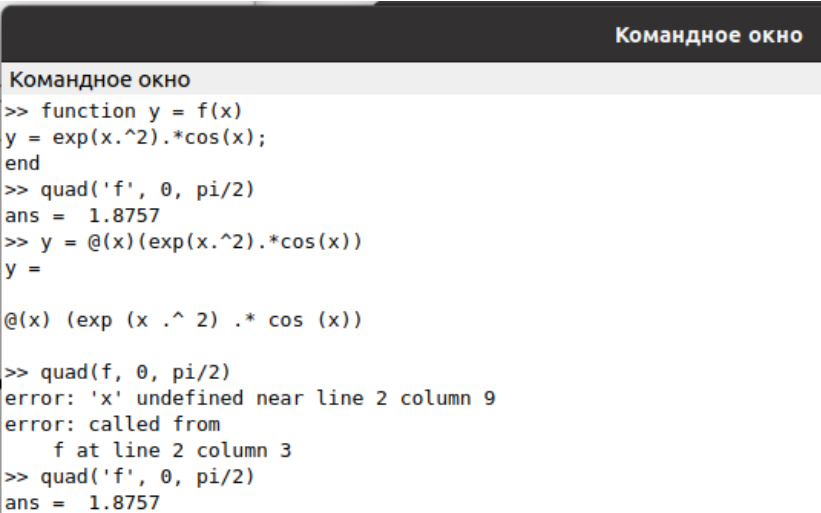
Рис. 4.3: Сумма ряда

Вычислим интеграл:

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

Для этого используем команду `quad('f', a, b)`. Зададим подынтегральную функ-

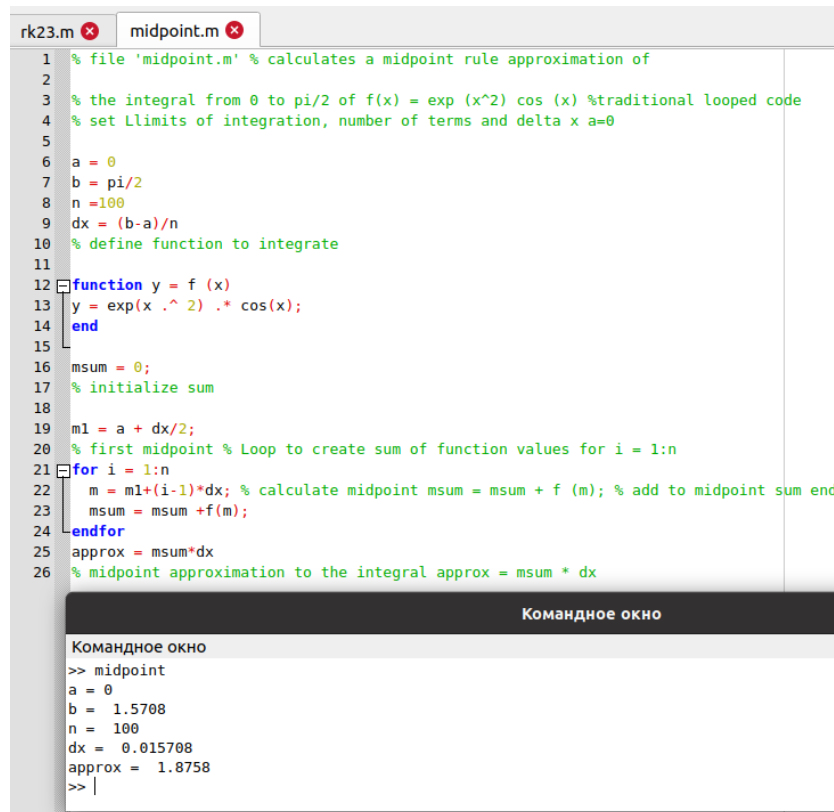
цию используя конструкцию `function...end` и анонимную функцию. (рис. [4.4])



```
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. 4.4: Вычисление интеграла. Функция `quad()`

Напишем скрипт, чтобы вычислить интеграл по правилу средней точки для $n = 100$. Будем использовать цикл, который добавляет значение функции к промежуточной сумме в каждой итерации. В конце сумма умножается на Δx (рис. [4.5])

The image shows a MATLAB script editor with two tabs: 'rk23.m' and 'midpoint.m'. The 'midpoint.m' tab is active, displaying a script that calculates the midpoint rule approximation of an integral. The script includes comments in green, function definitions in blue, and calculations in black. Below the script, a 'Командное окно' (Command Window) shows the output of the 'midpoint' function call, displaying the values of 'a', 'b', 'n', 'dx', and 'approx'.

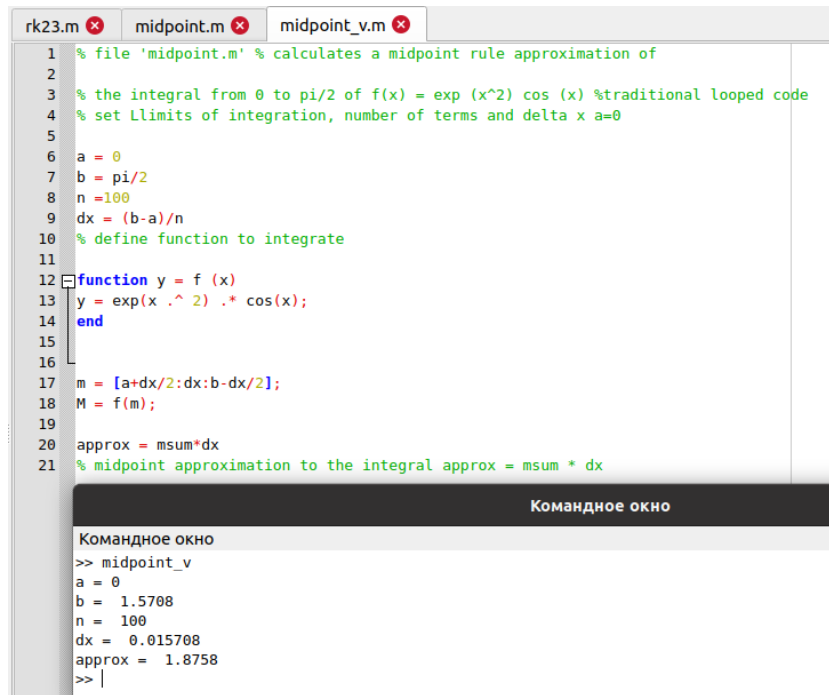
```
1 % file 'midpoint.m' % calculates a midpoint rule approximation of
2
3 % the integral from 0 to pi/2 of f(x) = exp (x^2) cos (x) %traditional looped code
4 % set limits of integration, number of terms and delta x a=0
5
6 a = 0
7 b = pi/2
8 n = 100
9 dx = (b-a)/n
10 % define function to integrate
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

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

Рис. 4.5: Вычисление интеграла по правилу средней точки.

Напишем векторизированный код для вычисления интеграла по правилу средней точки. Создадим вектор x -координат средних точек. Затем оценим f по этому вектору средней точки, чтобы получить вектор значений функции. Аппроксимация средней точки - это сумма компонент вектора, умноженная на Δx (рис. [4.6]).



```
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 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
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

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

Рис. 4.6: Вычисление интеграла по правилу средней точки. Векторизированный код.

Результаты были получены одинаковые. Сравним время выполнения для каждой реализации(рис. [4.7]).

```
>> 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. 4.7: Сравнение времени реализаций вычисления интеграла по правилу средней точки

5 Выводы

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

Список литературы

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/>.