

Лабораторная работа №6

Пределы, последовательности, ряды и интегрирование

Смирнов-Мальцев Егор Дмитриевич

Содержание

Цель работы	4
Задание	5
Теоретическое введение	6
Выполнение лабораторной работы	7
Выводы	11
Список литературы	12

Список иллюстраций

1	Оценка предела	7
2	Частичные суммы	8
3	График частичных суммы	8
4	Сумма гармонического ряда	9
5	Вычисление интеграла. Функция <code>quad()</code>	9
6	Вычисление интеграла по правилу средней точки через цикл.	9
7	Вычисление интеграла по правилу средней точки через векторы.	9
8	Сравнение времени выполнения вычислений интеграла различными функциями.	10

Цель работы

1. Научиться считать пределы,
2. Научиться работать с последовательностями и рядами,
3. Научиться выполнять численное интегрирование.

Задание

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

Теоретическое введение

Интеграл примерно равен сумме по разбиению значений умноженных на длину интервала. Octave - векторно-ориентированный язык. Поэтому стоит использовать вектора вместо циклов при выполнении заданий.

Выполнение лабораторной работы

Оценим предел:

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

Для этого определим функцию f равную этому выражению. Затем создадим вектор из степеней 10. Оценим $f(n)$. (рис. [-@fig:001])

```
octave:1> f = @(n) (1+1./n) .^ n;
octave:2> n = [0:10];
k =
  0
  1
  2
  3
  4
  5
  6
  7
  8
  9
octave:3> format long
octave:4> n = 10 .^ k
n =
  1
  10
  100
  1000
  10000
  100000
  1000000
  10000000
  100000000
octave:5> f(n)
ans =
  2.000000000000000
  2.593742460100002
  2.718281828459045
  2.716923932235594
  2.718281828459045
  2.718281828459045
  2.718281828459045
  2.718281828459045
  2.718281828459045
```

Рис. 1: Оценка предела

Найдем частичные суммы ряда

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

Для получения последовательности частичных сумм используем цикл и функцию `sum()`. Затем отобразим слагаемые и частичные суммы на графике (рис. [-@fig:002], [-@fig:003]).

```

octave> n = [1:11]';
octave> a = 1 ./ (n.^2 (n+2))
a
   1.2000000000000e-01
   6.0000000000000e-02
   4.0000000000000e-02
   2.4074285714285e-02
   1.8181818181818e-02
   1.5873015873015e-02
   1.2000000000000e-02
   1.0101010101010e-02
   8.3333333333333e-03
   6.9306930693069e-03

octave> for t = 1:10
   s(t) = sum(a(1:t));
end
octave> s'
ans =
   0.12500000000000
   0.15166666666667
   0.23333333333333
   0.26190476190476
   0.26738095238095
   0.26861111111111
   0.26738095238095
   0.26190476190476
   0.25000000000000
   0.23333333333333
   0.20000000000000

octave> plot(n,a,'o','n,s','+')
octave> hold on
octave> legend('terms', 'partial sums')

```

Рис. 2: Частичные суммы

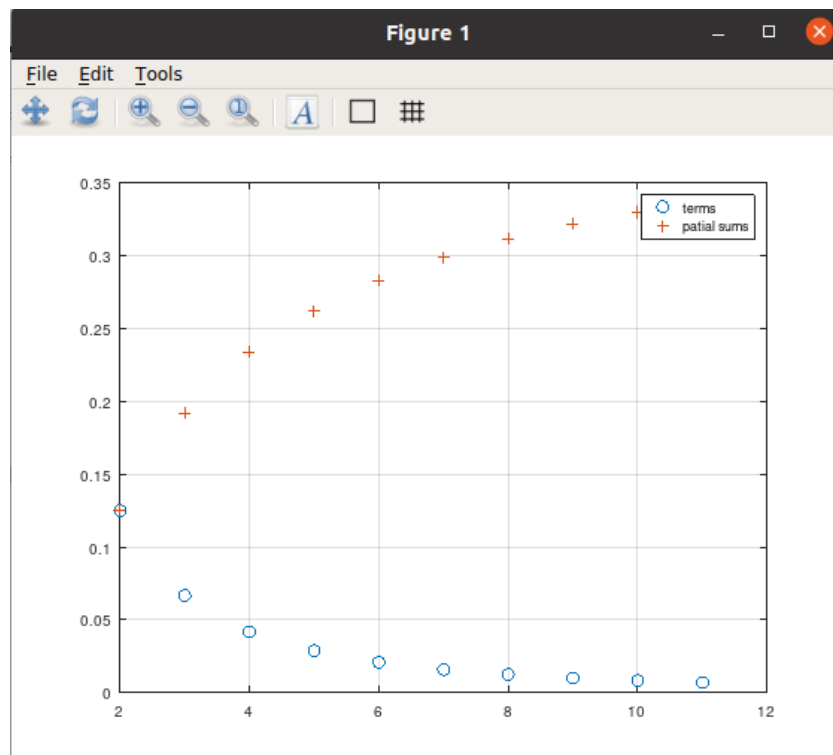


Рис. 3: График частичных суммы

Найдём сумму первых 1000 членов ряда

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

Для этого сгенерируем члены ряда как вектор и возьмём их сумму (рис. [-@fig:004]).


```

octave:12> legend('terms', 'partial sums')
octave:13> n = [1:1:1000];
octave:14> a = 1 ./ n;
octave:15> sum(a)
ans = 7.485470860550343
octave:16> function y=f(x)
> y = exp(x.^2) .* cos(x);

```

Рис. 4: Сумма гармонического ряда

Вычислим интеграл

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

с помощью встроенной функции `quad('f',a,b)` (рис. [-@fig:005])

```

octave:15> sum(a)
ans = 7.485470860550343
octave:16> function y=f(x)
> y = exp(x.^2) .* cos(x);
> end
octave:17> quad('f',0,pi/2)
ans = 1.875665011463391

```

Рис. 5: Вычисление интеграла. Функция `quad()`

Напишем функцию, вычисляющую интеграл по правилу средней точки через цикл. Она расположена в файле `programs/midpoint.m`. Применим ее (рис. [-@fig:006])

```

octave:12> midpoint
a = 0
b = 1.5708
n = 100
dx = 0.015708
approx = 1.8758
octave:13> midpoint_v

```

Рис. 6: Вычисление интеграла по правилу средней точки через цикл.

Напишем такую же функцию через векторы. Новая функция расположена в файле `programs/midpoint_v.m`. Применим ее (рис. [-@fig:007]).

```

octave:13> midpoint_v
a = 0
b = 1.5708
n = 100
dx = 0.015708
approx = 1.8758
octave:14> tic; midpoint; toc

```

Рис. 7: Вычисление интеграла по правилу средней точки через векторы.

Сравним время выполнения для каждой функции (рис. [-@fig:008]).

```

%Elapsed = tic; midpoint; toc
a = 0
b = 1.5708
n = 100
dx = 0.015708
approx = 1.0728
Elapsed time is 0.00377393 seconds.
%Elapsed = tic; midpoint_v; toc
a = 0
b = 1.5708
n = 100
dx = 0.015708
approx = 1.0728
Elapsed time is 0.00048399 seconds.
end

```

Рис. 8: Сравнение времени выполнения вычислений интеграла различными функциями.

Выводы

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

Список литературы