ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

Пределы, последовательности и ряды

Аминов Зулфикор Мирзокаримович

Содержание

Ход работы:

Пределы, последовательности и ряды

Рассмотрим предел:

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

рисунка 1

Оценим это выражение с методом анонимной функцией. Это хороший способ быстро определить простую функйию.

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

@(n) (1 + 1 ./ n) .^ n
```

рисунка 2

Далее мы создаём индексную переменную, состоящую из целых чисел от 0 до 9:

```
>> k = [0:1:9]'
k =

   0
   1
   2
   3
   4
   5
   6
   7
   8
   9
```

рисунка 3

Теперь мы возьмём степени 10, которые будут входными значениями, а затем оценим f(n).

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

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

```
>> f (n)
ans =

   2.000000000000000
   2.593742460100002
   2.704813829421529
   2.716923932235520
   2.718145926824356
   2.718268237197528
   2.718280469156428
   2.718281693980372
   2.718281786395798
   2.718282030814509

>> format
```

Частичные суммы

Пусть $a \sum_{n=2}^{\infty} a_n$ -ряд, n-й член равен

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

рисунка 7

Для этого мы определим индксный вектор n от 2 до 11, а затем вычислим члены.

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

рисунка 8

Если мы хотим частичную сумму, нам нужно только написать sum(a). Если мы хотим получить последовательность частичных сумм, нам нужно использовать цикл. Мы будем использоватьцикл for с индексом i от 10. Для каждого i мы получим частичную сумму последовательность a_n от первого слагаемого до i-го слагаемого. На выходе получается 10-элементный вектор этих частичных сумм.

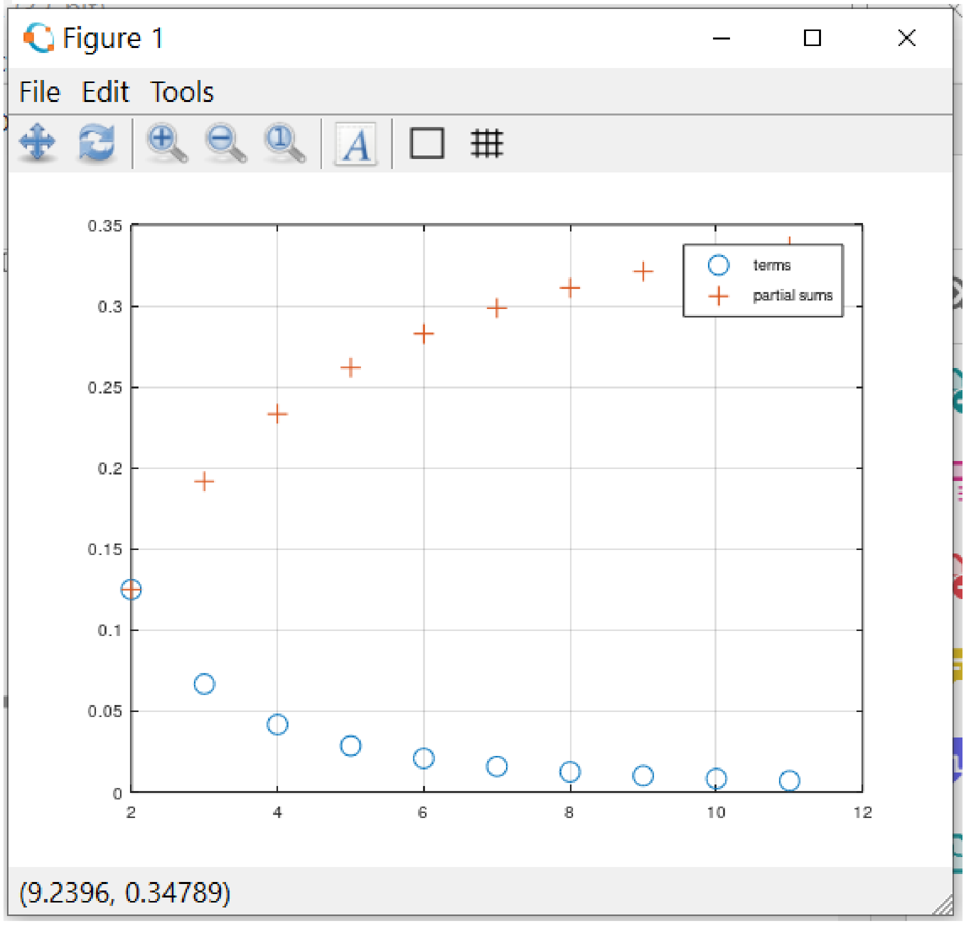
```
>> for i = 1:10
s (i) = sum (a(1:i));
end
>> s'
ans =

   0.1250
   0.1917
   0.2333
   0.2619
   0.2827
   0.2986
   0.3111
   0.3212
   0.3295
   0.3365
```

рисунка 9

Наконец, мы построим слагаемые и частичные суммы для 2<= n <= 11.

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



Сумма ряда

Найдем сумму первых 1000 членов гармонического ряда:

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

рисунка 12

Нам нужно только сгенировать члены как ряда вектор, а затем взять их сумму.

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

рисунка 13

Частичные интегрирование

Вычисление интегралов

С помощью команду quar вычислим интеграл:

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

рисунка 14

Синтаксис команды -quar(‘f’, a, b). Нам нужно сначала определить функйию.

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

рисунка 15

Аппроксимирование суммами

Напишем скрипт, чтобы вычислить интеграл

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

рисунка 16

по правилу средней точки для n = 100.

Введим код в текстовом файле и назавем его midpoint.m.

midpoint.m – Блокнот

Файл Правка Формат Вид Справка

```
% file 'midpoint.m'
% calculates a midpoint rule approximation of
% the integral from 0 to pi/2 of f(x) = exp (x^2) cos (x)
% -- traditional looped code
% set limits of integration, number of terms and delta x
a = 0
b = pi/2
n = 100
dx = (b-a)/n
% define function to ontegrate
function y = f (x)
y = exp(x .^2) .* cos(x);
end
msum = 0;
% initialize sum
m1 = a + dx/2; % first midpoint
% loop to create sum of function values
for i = 1:n
m = m1 + (i-1) * dx; % calculate midpoint
msum = msum + f (m); % add to midpoint sum
end
% midpoint approximation to the integral
approx = msum * dx
```

рисунка 17

Набрав midpoint в командной строке запустим скрипт

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

рисунка 18

Создадим вектор x-координат средних точек. Затем мы оцениваем f по этому вектору средней точки, чтобы получить вектор значений функции. Аппроксимация средней точки -это сумма компонент вектора, умноженная на делтаX.

midpoint_v.m – Блокнот

Файл Правка Формат Вид Справка

```
% file 'midpoint_v.m'
% calculates a midpoint rule approximation of
% the integral from 0 to pi/2 of f(x) = exp (x^2) cos (x)
% -- vectorized code
% set limits of integration, number of terms and delta x
a = 0
b = pi/2
n = 100
dx = (b-a)/n
% define function to integrate
function y = f (x)
y = exp(x .^2) .* cos(x);
end
% create vector of midpoints
m = [a+dx/2:dx:b-dx/2];
%create vector of function values at midpoints
M = f(m);
% midpoint approximation to the integral
approx = dx * sum (M)
```

рисунка 19

Запустим его.

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

рисунка 20

Сравниваем результаты и сравниваем время выполнения для каждой реализации.

```
>> tic; midpoint; toc
a = 0
b = 1.5708
n = 100
dx = 0.015708
approx = 1.8758
Elapsed time is 0.00917101 seconds.
>> tic; midpoint_v; toc
a = 0
b = 1.5708
n = 100
dx = 0.015708
approx = 1.8758
Elapsed time is 0.00212193 seconds.
```

рисунка 21

Вывод

Научился работать в Octave с пределамы, последовательностами и с рядамы.