Отчёт по лабораторной работе №6

Дисциплна: Научное программирование

Живцова Анна, 1132249547

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Изучить способы использования Octave для задач - Оценки предела последовательности

- Поиска частичных сумм рядов
- Численного интегрирования

2 Задание

Используя векторные вычисления Octave, реализовать

- Оценку предела последовательности
- Поиск частичных сумм рядов
- Численное интегрирование

3 Теоретическое введение

Octave – полноценный язык программирования, поддерживающий множество типов циклов и условных операторов. Однако, поскольку это векторный язык, многие вещи, которые можно было бы сделать с помощью циклов, векторизовать и получить выигрыш в производительности [1]. В данной работе мы рассмотрим распространенные численные задачи оценки предела последовательности, нахождения частичных сумм и численного интегрирования.

4 Выполнение лабораторной работы

4.1 Оценка предела последовательности

Будем оценивать предел последоательности $a_n = \left(1 + \frac{1}{n}\right)^n$ при $n \to \infty$. Известно, что этот предел равен $e = 2,7182818284$. Найдем значения a_{n_i} при некоторой конечной возрастающей последовательности n_i. (см рис. 4.1).

```
>> f = @(n)(1+1./n).^n;
>> k=[0:1:9]';
>> format long
>> n = 10.^k;
>> f(n)
ans =

   2.000000000000000
   2.593742460100002
   2.704813829421529
   2.716923932235520
   2.718145926824356
   2.718268237197528
   2.718280469156428
   2.718281693980372
   2.718281786395798
   2.718282030814509
```

Рис. 4.1: Оценка предела последовательности

4.2 Частичные суммы

Рассмотрим ряд, n-ый член которого a_n равен $\frac{1}{n(n+2)}$. Найдем и изобразим его частичные суммы (см рис. 4.2). Дополнительно найдем сумму первых 1000 членов гармонического ряда. Она примерно равна 7.485470860550343.

```
>> n = [2:1:11]';
>> a = 1./(n.*(n+2));
>> a
a =

   1.250000000000000e-01
   6.666666666666667e-02
   4.166666666666666e-02
   2.857142857142857e-02
   2.083333333333333e-02
   1.587301587301587e-02
   1.250000000000000e-02
   1.010101010101010e-02
   8.333333333333333e-03
   6.993006993006993e-03

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

   0.125000000000000
   0.191666666666667
   0.233333333333333
   0.261904761904762
   0.282738095238095
   0.298611111111111
   0.311111111111111
   0.321212121212121
   0.329545454545455
   0.336538461538462

>> plot(n,a, 'o', n, s, '+')
>> grid on;
>> legend('terms', 'partial sums')
>> n=[1:1:1000];
>> a = 1./n;
>> sum(a)
ans = 7.485470860550343
```

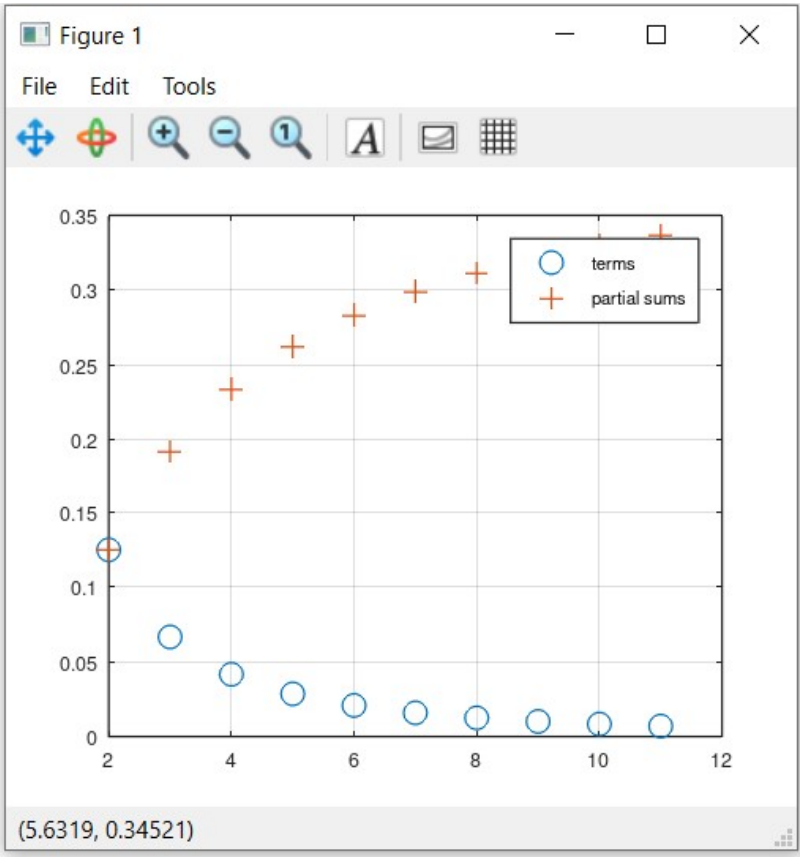

Рис. 4.2: Поиск частичных сумм

4.3 Численное интегрирование

Будем работать с определенным интегралом $\int_0^{\pi/2} e^x \cos(x)dx$. Вычислим его с помощью встроенной функции (см рис. 4.3), а также с помощью двух программ, реализующих метод прямоугольников (см рис. 4.4 и 4.5). Сравним производительность программы, использующей векторные вычисления, и программы, не

использующей векторные вычисления (см рис. 4.6 (см рис. 4.7).

```
>> f=@(x)(exp(x.^2).*cos(x));
>> quad(f,0,pi/2)
ans = 1.875665011463391
```

Рис. 4.3: Численное интегрирование встроенной функцией

```
a = 0
b = pi/2
n = 100
dx = (b - a)/n
function y = f(x)
  y = exp(x.^2).*cos(x);
end
msum = 0;
m1 = a + dx/2;
for i=1:n
  m = m1 + (i-1)*dx;
  msum = msum + f(m);
end
approx = msum*dx
```

Рис. 4.4: Программа для численного интегрирования, использующая цикл

```
a = 0
b = pi/2
n = 100
dx = (b - a)/n
function y = f(x)
  y = exp(x.^2).*cos(x);
end
m = [a+dx/2:dx:b-dx/2];
approx = dx*sum(f(m))
```

Рис. 4.5: Программа для численного интегрирования, использующая векторизацию

```
>> tic; midpoint; toc;
a = 0
b = 1.5708
n = 100
dx = 0.015708
approx = 1.8758
Elapsed time is 0.00576401 seconds.
```

Рис. 4.6: Время выполнения и результат программы, использующей цикл

```
>> tic; midpoint_v; toc;
a = 0
b = 1.5708
n = 100
dx = 0.015708
approx = 1.8758
Elapsed time is 0.00118804 seconds.
```

Рис. 4.7: Время выполнения и результат программы, использующей векторизацию

5 Выводы

В данной работе я научилась эффективно использовать Octave для задач оценки предела последовательности, поиска частичных сумм рядов и численного интегрирования. Также, на примере задачи численного интегрирования, я произвела оценку производительности программы, использующей векторные вычисления, и программы, не использующей векторные вычисления.

6 Список литературы

1. GNU Octave documentation. The Octave Project Developers, 2024.