Презентация по лабораторной работе №6

Дисциплина “Научное программирование”

Живцова А.А.

11 октября 2024

Кафедра теории вероятностей и кибербезопасности, Российский университет дружбы народов имени Патриса Лумумбы, Москва, Россия

Информация

Докладчик

- Живцова Анна Александровна
- студент кафедры теории вероятностей и кибербезопасности
- Российский университет дружбы народов имени Патриса Лумумбы
- zhivtsova_aa@pfur.ru
- https://github.com/AnnaZhiv

Вводная часть

Актуальность

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.

Объект и предмет исследования

- Задача оценки предела последовательности
- Задача поиска частичных сумм рядов
- Задача численного интегрирования
- Векторизация
- Octave

Цели

Изучить способы использования Octave для задач

- Оценки предела последовательности
- Поиска частичных сумм рядов
- Численного интегрирования

Задачи

Используя векторные вычисления Octave, реализовать

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

Материалы и методы

- Язык научного программирования Octave
- Среда программирования GNUoctave

Выполнение работы

Оценка предела последовательности

Оценим предел последоательности $a_n = \left(1 + \frac{1}{n}\right)^n$ при $n \to \infty$.

```
>> 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
```

Частичные суммы

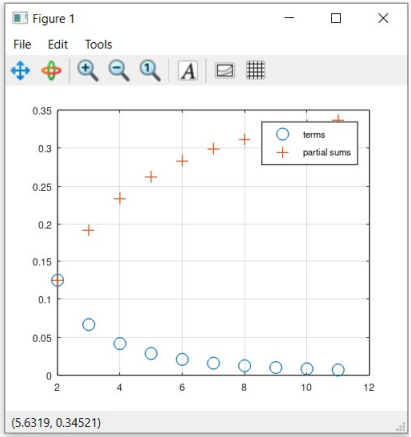
Найдем и изобразим частичные суммы ряда $a_n \frac{1}{n(n+2)}$. Дополнительно найдем сумму первых 1000 членов гармонического ряда. Она примерно равна 7.485470860550343.

```
>> 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
```



Вычисляем интеграл $\int_0^{\pi/2} e^x \cos(x) dx$ с помощью двух программ, реализующих метод прямоугольников.

Цикл

```
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
```

Векторизация

```
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))
```

Результаты

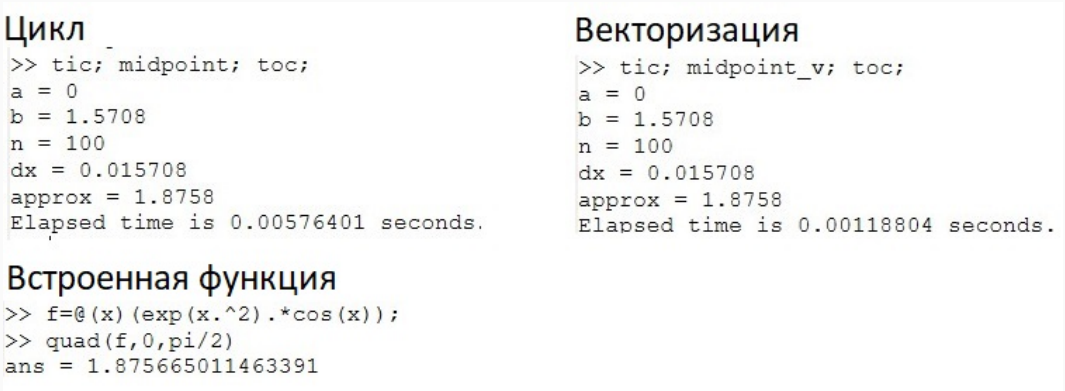


Рис. 4: Сравнение выполнения программ с векторизацие и без

Выводы

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.