Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Кафедра информатики и прикладной математики

Лабораторная работа №1
Дисциплина «Символьные вычисления»

Вариант 13

Выполнил:
Съестов Дмитрий Вячеславович
Группа Р3317

Преподаватель:
Кореньков Юрий Дмитриевич

Санкт-Петербург
2018

Задание

Сравнить характеристики и выделить особенности методов символьного и численного интегрирования с помощью пакета символьных вычислений по варианту.

Порядок выполнения:

- Изучить средства интегрирования данного пакета символьных вычислений (Таблица 1)
- Рассмотреть различия и сходства в способах применения символьного и численного интегрирования
- Выполнить численное интегрирование заданной функции (Таблица 2)
- Выполнить символьное интегрирование заданной функции (Таблица 2)
- Сравнить результаты, сделать выводы, оформить отчет по работе и продемонстрировать преподавателю для защиты

Матпакет: **SageMath**

$$\int \frac{dx}{(x^2+1)^2} \qquad \int \frac{x^2 dx}{(x^2+1)^2} \qquad \int \frac{dx}{e^x+1}$$

Исходный код программы:

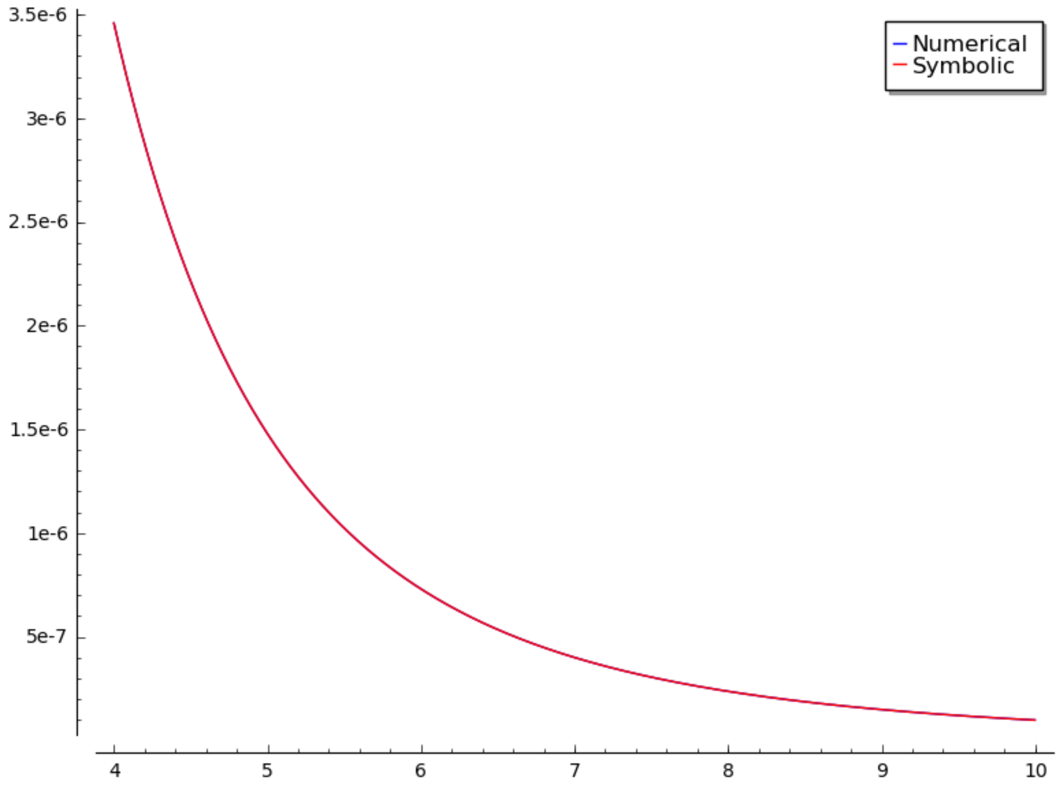
```
from sage.calculus.integration import numerical_integral
from sage.symbolic.integration.integral import definite_integral, indefinite_integral

def numerical(f, a, b, step):
   x = a
   lst = []
   while x + step < b:
      dy, _ = numerical_integral(f, x, x + step)
      lst.append((x, dy))
      x += step
   return lst

def symbolic(f, a, b, step):
   x = a
   lst = []
   while x + step < b:
      dy = RR(f(x + step) - f(x))
      lst.append((x, dy))
      x += step
   return lst

x = SR.var('x')
```

```
functions = [
    SR('1 / (x^2 + 1)^2'),
    SR('x^2 / (1 + x^2)^2'),
    SR('1 / (exp(x) + 1)')
]

ranges = [
    (4, 10),
    (-9, 0),
    (2, 10)
]

for i, f in enumerate(functions):
    print('Function {0}: {1}'.format(i+1, f))
    antiderivative = f.integral(x)
    print('Antiderivative: ' + str(antiderivative) + '\n')

    f_plot = plot(f, (-10, 10), color='green', legend_label=str(f))
    g_plot = plot(antiderivative, (-10, 10), color='orange', legend_label=str(antiderivative))
    show(f_plot + g_plot)

    a, b = ranges[i]
    num_result = numerical(f, a, b, 0.01)
    sym_result = symbolic(f, a, b, 0.01)

    num_plot = line(num_result, color='blue', legend_label='Numerical')
    sym_plot = line(sym_result, color='red', legend_label='Symbolic')
    show(num_plot + sym_plot)

    delta = [(s[0], s[1] - n[1]) for n, s in zip(num_result, sym_result)]
    delta_plot = line(delta, color='purple', legend_label='Difference')
    show(delta_plot)
```

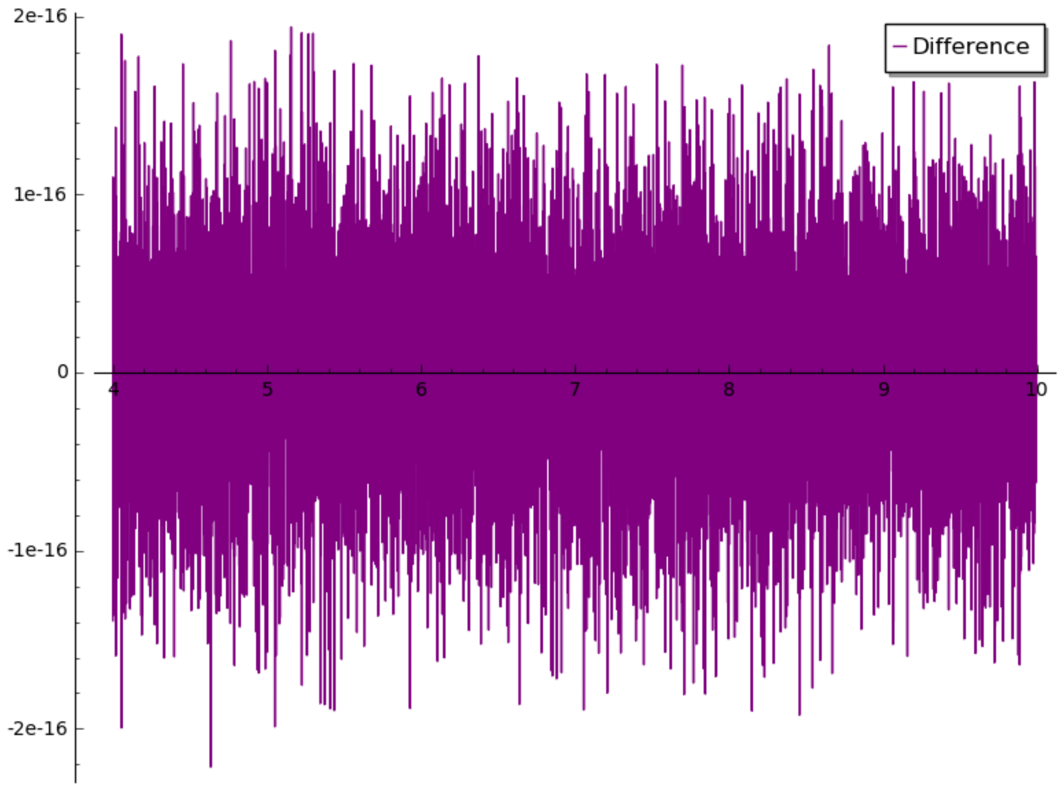
Результат:

```
Function 1: (x^2 + 1)^(-2)
Antiderivative: 1/2*x/(x^2 + 1) + 1/2*arctan(x)

Function 2: x^2/(x^2 + 1)^2
Antiderivative: -1/2*x/(x^2 + 1) + 1/2*arctan(x)

Function 3: 1/(e^x + 1)
Antiderivative: x - log(e^x + 1)
```

(x^2 + 1)^(-2)

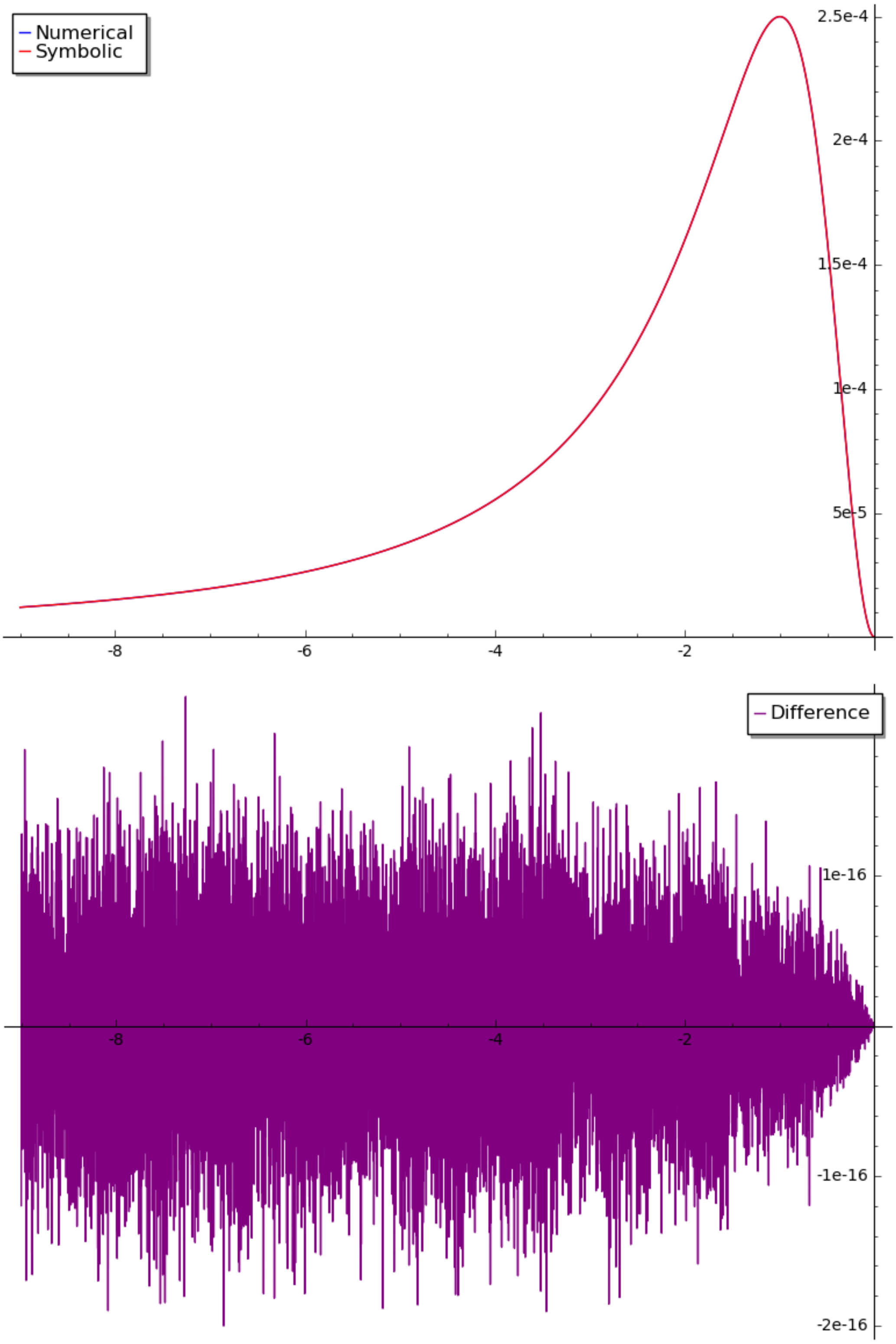


3.5e-6
3e-6
2.5e-6
2e-6
1.5e-6
1e-6
5e-7
4
5
6
7
8
9
10
Numerical
Symbolic

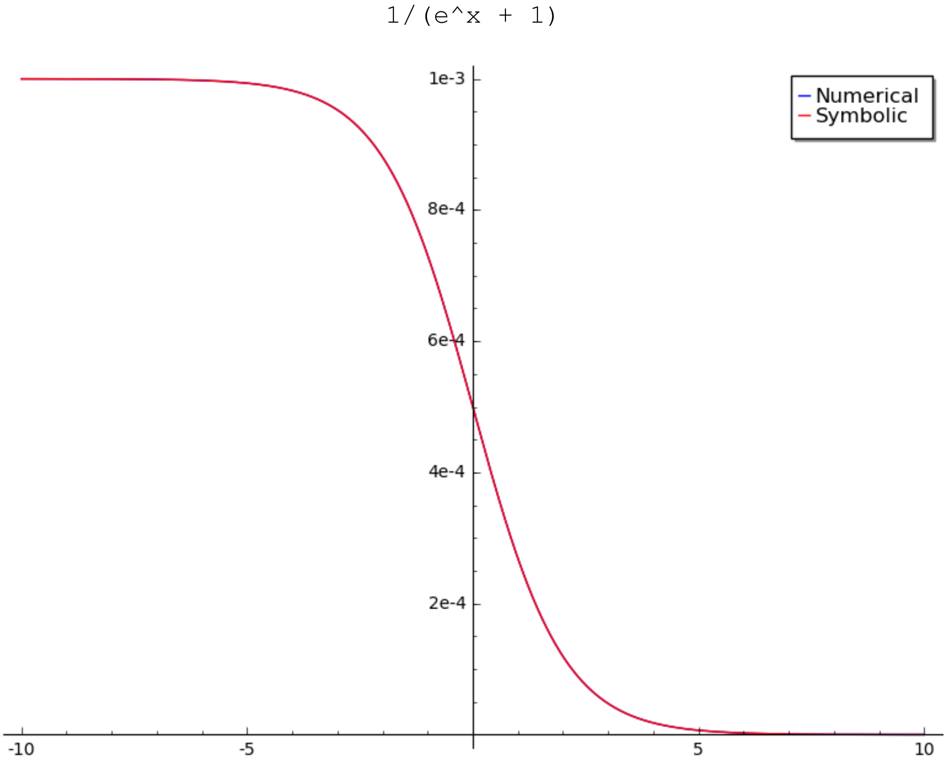


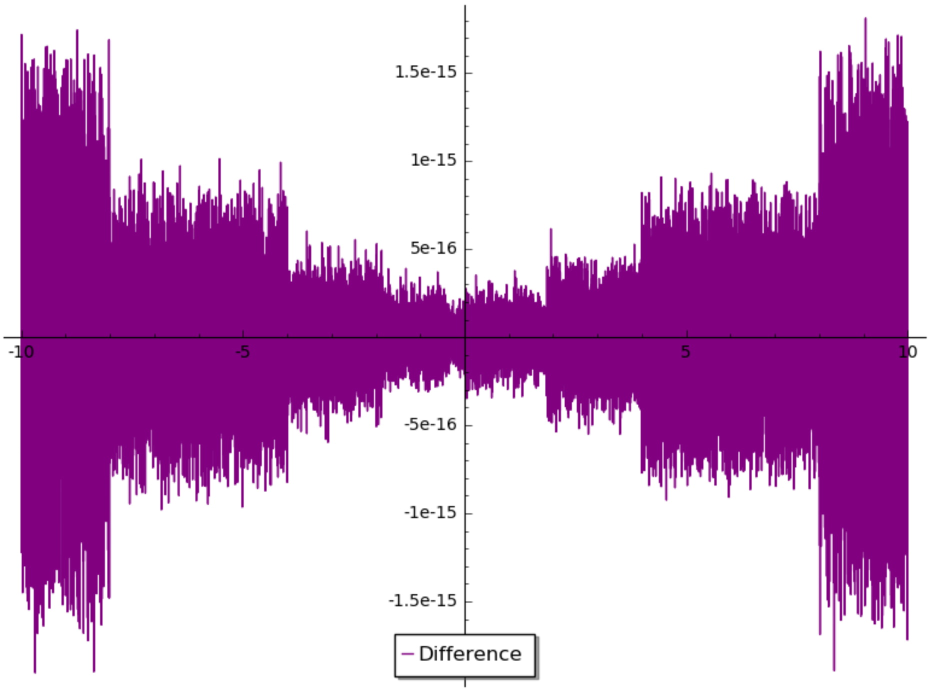
2e-16
1e-16
0
-1e-16
-2e-16
4
5
6
7
8
9
10
Difference

x^2/(x^2 + 1)^2



Numerical
Symbolic
2.5e-4
2e-4
1.5e-4
1e-4
5e-5
-8
-6
-4
-2
Difference
1e-16
-1e-16
-2e-16
-8
-6
-4
-2





Вывод

Sage по умолчанию использует адаптивное интегрирование. Это означает, что точность подстраивается под данный участок функции, и поэтому погрешность может сильно изменяться на разных участках интегрирования (см. последний график)

Преимущество Sage по сравнению с другими системами компьютерной алгебры в том, что он является свободным ПО и может использоваться отдельно в виде библиотек для Python.