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 P3317

**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**

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)

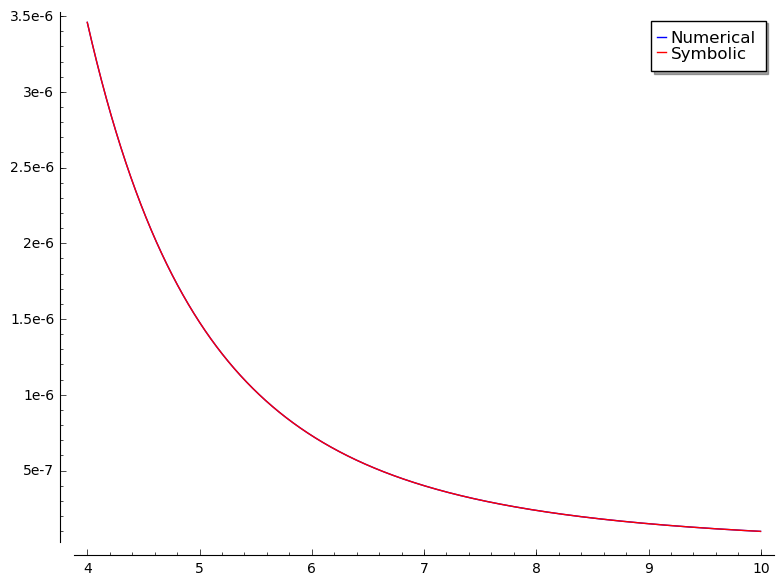
Function 2: x^2/(x^2 + 1)^2

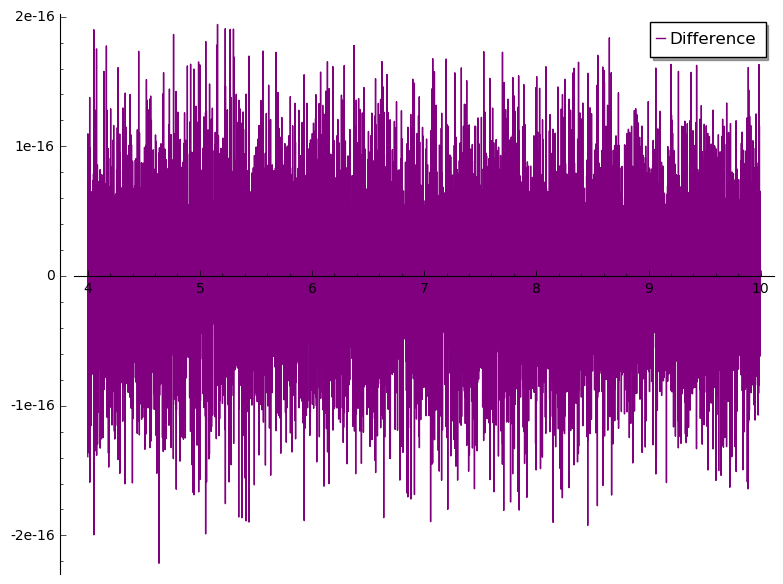
Antiderivative: -1/2\*x/(x^2 + 1) + 1/2\*arctan(x)

Function 3: 1/(e^x + 1)

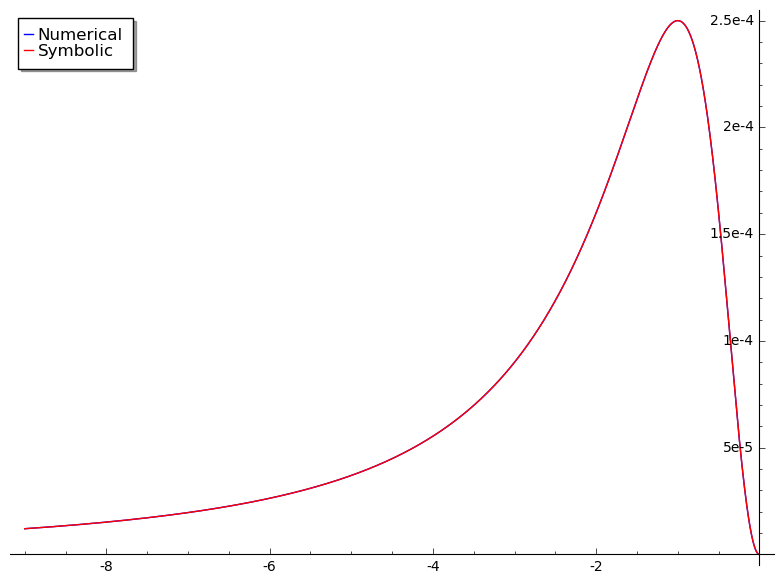
Antiderivative: x - log(e^x + 1)

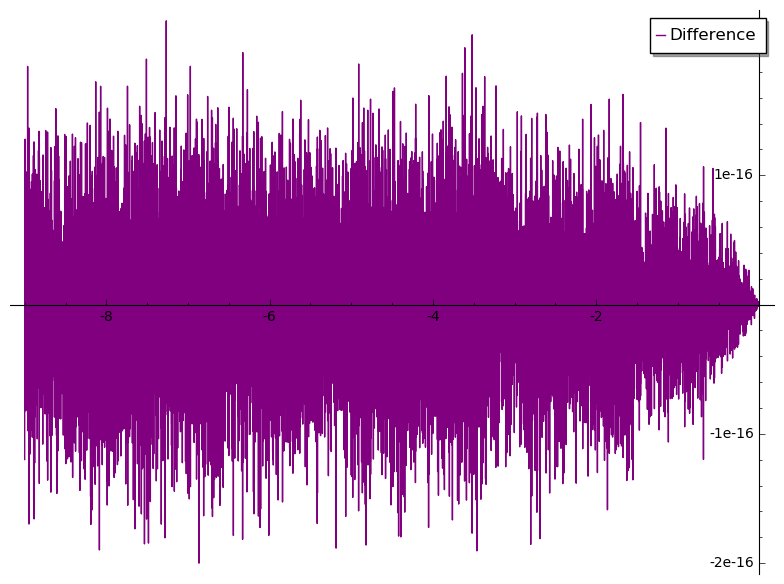
(x^2 + 1)^(-2)



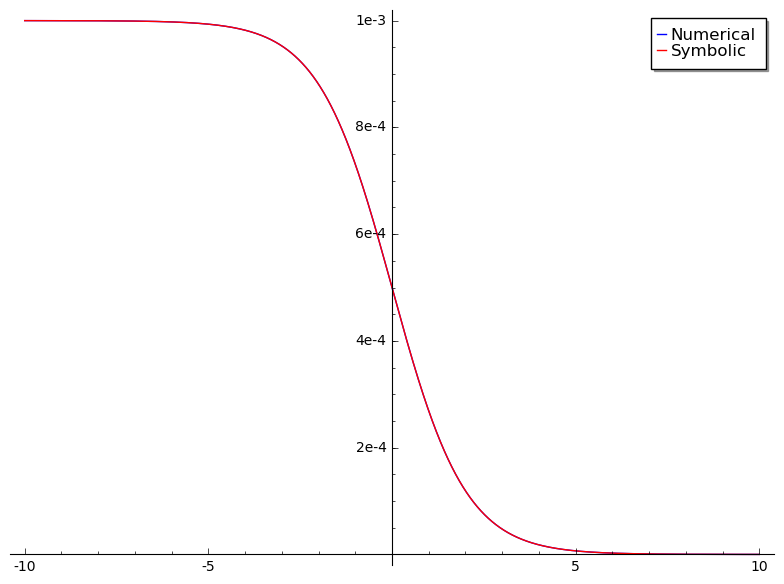


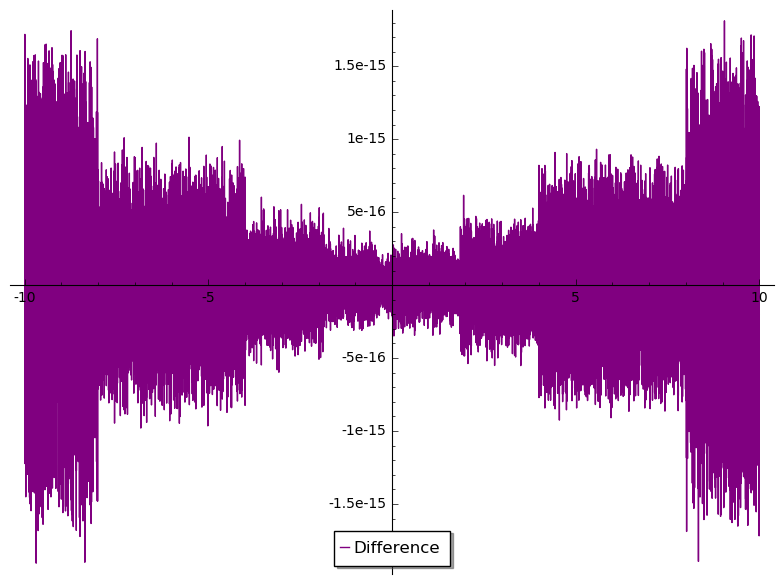
x^2/(x^2 + 1)^2





1/(e^x + 1)

****

****

**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.