

УДК 621.311

ИСПОЛЬЗОВАНИЕ ПРИКЛАДНЫХ МАТЕМАТИЧЕСКИХ ПАКЕТОВ ДЛЯ ВЫЧИСЛЕНИЯ ОПРЕДЕЛЁННОГО ИНТЕГРАЛА И ЕГО ПРИЛОЖЕНИЙ

Юреть Н.Г., Политыко А.В.

Научный руководитель – Бадак Б.А., старший преподаватель кафедры «Высшая математика»

Определенный интеграл является одним из важнейших понятий математического анализа, который находит широкое применение в различных областях науки и техники. В настоящее время существует множество прикладных математических пакетов, которые позволяют вычислять определенные интегралы. Эти пакеты предоставляют широкий спектр функций для работы с интегралами, что значительно упрощает и ускоряет процесс вычислений. В работе рассмотрены некоторые из наиболее популярных прикладных математических пакетов, такие как Mathcad и Python[1], а также примеры использования этих пакетов для вычисления определенных интегралов и их приложений.

В Mathcad для вычисления определенного интеграла используется функция `int`. Синтаксис этой функции имеет вид: `int(f(x), a, b)`. Например, для вычисления интеграла $\int_0^1 x^2 dx$ в Mathcad можно использовать следующий код: `int(x^2, 0, 1)`.

В Python для вычисления определенного интеграла можно использовать различные библиотеки, одна из таких `sympy`. Библиотека `sympy` предоставляет функцию `integrate`, которая позволяет вычислить определенный интеграл с помощью символического метода. Синтаксис этой функции следующий: `integrate(f, x, a, b)`.

Например, для вычисления интеграла в Python можно использовать следующий код:

```
 $\int_0^1 x^2 dx$   
from sympy import integrate, symbols  
x = symbols('x')  
result = integrate(x**2, x, 0, 1)
```

При сравнении результатов были выявлены следующие отличия: Python менее производительный, чем Mathcad, особенно для задач, требующих высокой точности. Оба приложения показали высокий уровень точности, так же в настройках программы этот уровень можно изменить. Mathcad имеет простой и понятный интерфейс, удобный для начинающих пользователей. Python требует навыков программирования, что может сделать его менее удобным для начинающих пользователей. Mathcad имеет ограниченный набор функций для работы с интегралами. Python предлагает широкий спектр библиотек для работы с

интегралами, включая численные методы, символическое интегрирование и визуализацию.

Использование интегралов может быть применено в производственных задачах. Рассмотрим пример такой *задачи*: есть производственная линия, на которой изготавливаются детали. Необходимо определить общие затраты на производство, включая расходы на энергию и материалы.

Предположим, у нас есть формула для расчета затрат на производство одной детали, которая зависит от времени производства и расхода материалов. Давайте представим, что эта формула имеет вид:

$$\int_0^T (E(t) + M(t))dt$$

где:

- T - время производства одной детали,
- $E(t)$ - функция, описывающая затраты на энергию в зависимости от времени t ,
- $M(t)$ - функция, описывающая затраты на материалы в зависимости от времени t .

Пример решения задачи через Python с библиотекой sympy:

```
from sympy import symbols, integrate
# Определяем символы
t = symbols('t')
# Предположим, у нас есть функции затрат на энергию E(t) и на материалы M(t)
# Для примера возьмем некоторые выражения, сравнимые с реальными функциями
E_t = 2*t # Пример функции затрат на энергию
M_t = 3*t**2 # Пример функции затрат на материалы
# Общие затраты на производство одной детали
total_costs = integrate(E_t + M_t, (t, 0, T))
# Предположим, что время производства одной детали равно 10 часам
T = 10
# Вычисляем общие затраты на производство детали
result = total_costs.subs(T, 10)
# Вычисляем общие затраты
print("Общие затраты на производство детали:", result)
Вывод: 1100
```

Пример решения задачи в Mathcad через функцию int:

1. Определение функций:
 $E(t) := 2 \cdot t$
 $M(t) := 3 \cdot t^2$
2. Задание времени производства:
 $T := 10$
3. Вычисление общих затрат:
Затраты := int($E(t) + M(t)$, t , 0, T)

Вывод 1100

Используя Mathcad и библиотеку SymPy в Python для символьного вычисления интегралов, мы видим, что оба инструмента предоставляют удобные способы решения аналитических задач. Mathcad предлагает интуитивно понятный интерфейс и прямое решение задач в символьной форме, что делает его привлекательным для быстрого и простого решения простых задач. С другой стороны, SymPy в Python предоставляет более гибкий подход и может быть частью более крупного программного проекта, благодаря своей интеграции с другими библиотеками и инструментами.

Mathcad предоставляет простой и интуитивно понятный интерфейс, а также прямое решение символьных задач, что делает его эффективным инструментом для быстрого анализа и решения простых математических задач. С другой стороны, библиотека SymPy в Python предоставляет более гибкий подход, позволяя проводить символьные вычисления в рамках более крупных программных проектов. Таким образом, использование прикладных математических пакетов и библиотек символьного вычисления обеспечивает удобные и эффективные инструменты для анализа и моделирования математических задач.

Литература

1. Смирнов В.И. «Введение в численные методы»– Санкт-Петербург: БХВ-Петербург, 2019.