

Вычислительная математика

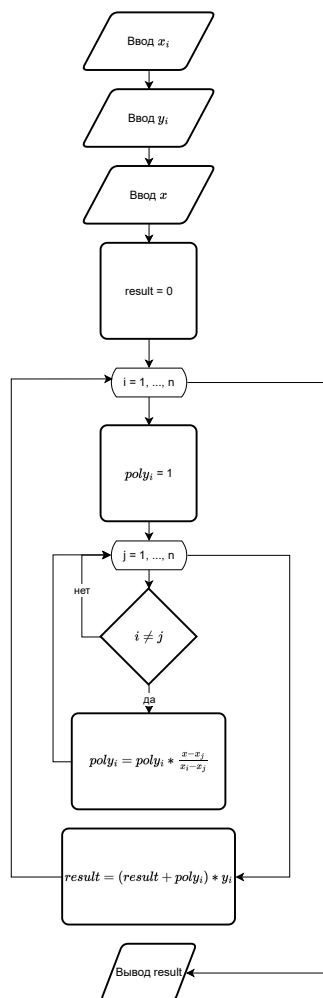
04.03.2024

Отчет по лабораторной работе 1

Описание численного метода

Метод интерполяции Лагранжа - метод аппроксимации, который используется для нахождения полинома, который приближает заданный набор данных. Метод основан на построении интерполяционного полинома в форме Лагранжа, который проходит через все заданные точки. Полином состоит из суммы произведений значения функции в точке x на базисный полином в этой точке. Полином Лагранжа для x_1, x_2, \dots, x_n и y_1, y_2, \dots, y_n точек выражается, как $L(x) = \sum_{i=1}^n y_i * l_i(x)$, где $l_i(x)$ - i -й базисный полином. $l_i(x)$ определяется, как $l_i(x) = \prod_{j=1, j \neq i}^n \frac{x-x_j}{x_i-x_j}$, при этом $i \neq j$. при этом, базисный полином равен 1 в точке x_i , а в точках x_j он равен нулю.

Блок-схема



Листинг кода

```
1 def interpolate_by_lagrange(x_axis, y_axis, x): # x_axis, y_axis - interpolation nodes, x -  
    interpolation point  
2     result = 0 # result of interpolation  
3  
4     for i in range(len(x_axis)):  
5         poly_i = 1 # Lagrange multiplier  
6  
7         for j in range(len(x_axis)):  
8             if i != j: # to avoid zero division  
9                 poly_i *= (x - x_axis[j]) / (x_axis[i] - x_axis[j])  
10  
11         result += poly_i * y_axis[i] # final interpolation result for this iteration
```

```
12     return result
```

Листинг 1: Python example

Примеры работы программы

Стандартный случай

stdin:

```
1 3
2 -1.0 0.0 1.0
3 -1.0 0.0 1.0
4 0.5
```

stdout:

```
1 0.5
```

Программа отработала штатно, выдав значение интерполированной функции

Граничные условия

stdin:

```
1 2
2 -1.0 1.0
3 -1.0 1.0
4 -1.0
```

stdout:

```
1 -1.0
```

Программа вывела значение '-1.0' в точности, так как интерполяция в точке одного из узлов возвращает значение этого узла

Нарушение уникальности узлов

stdin:

```
1 3
2 0.0 0.0 0.0
3 0.0 1.0 2.0
4 0.5
```

Программа не сможет корректно отработать, так как нарушает условие интерполяции Лагранжа ($\forall i, j x_i \neq x_j$)

Интерполяция вне диапазона

stdin:

```
1 4
2 -1.0 -0.5 0.5 1.0
3 1.0 0.5 -0.5 -1.0
4 2.0
```

stdout:

```
1 -2.0
```

Программа посчитала значение интерлированной функции в точке 2.0, однако эта точка лежит за пределом интерполирования. Из-за этого результат может быть неточным.

Одна точка

stdin:

```
1 1
2 0.0
```

3	1.0
4	0.0

stdout:

1	1.0
---	-----

Программа вернет значение функции в единственной точке интерполяции.

Выводы

Программа стабильно обрабатывает на данных, если они удовлетворяют условиям возможности интерполяции Лагранжа. Однако она не способна выполнить расчет, если входные данные нарушают эти требования (например, если значения x у узлов интерполяции не уникальны, программа не отработает должным образом).

Метод Лагранжа - это очень простой, но при этом мощный и удобный инструмент для аппроксимации различных функций на основе заданных данных. Он хорошо работает с небольшими, равномерно распределенными данными. Однако из-за его высокой алгоритмической сложности (алгоритмическая сложность интерполяции - $O(n^2)$, где n - количество точек из-за вложенного цикла), возможности проявления феномена Рунге, а так же из-за проблем с численной стабильностью он имеет ограниченное применение. Например, метод интерполяции Ньютона дает возможность добавлять новые точки, не пересчитывая весь полином, а метод кубических сплайнов позволяет делать более гладкую аппроксимацию.