|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  |
| МИНОБРНАУКИ РОССИИ | | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | | |
| **Институт кибернетики** | | | |
| **Кафедра высшей математики** | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **КУРСОВАЯ РАБОТА** | | | | | | | |
| **по дисциплине** | | | | | | | |
| *«* | *Численные методы* | | | | | | *»* |
|  | | | | | | | |
| **Тема курсовой работы** | | | **«** | **Вычисление минимума функции,** | | | |
| **зависящей от интерполяционного многочлена.** | | | | | | | **»** |
|  | | | | | | | |
| Студент группы | | **КМБО-03-16** | | |  | **Антонов Артём Евгеньевич** | |
|  | |  | | |  |  | |
|  | |  | | |  | к.т.н., доцент | |
| Руководитель курсовой работы | | | | |  | **Сенявин Михаил Маркович** | |
|  | | | | |  |  | |

|  |  |  |
| --- | --- | --- |
| Работа представлена к защите | «\_\_»\_\_\_\_\_\_\_201\_\_\_ г. |  |
| *(подпись студента)* |
|  |  |  |
| «Допущен к защите» | «\_\_»\_\_\_\_\_\_\_201\_\_\_ г. |  |
| *(подпись руководителя)* |

Москва 2019

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | | | |  | | |
| МИНОБРНАУКИ РОССИИ | | | | | | | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | | | | | | | |
| **Институт кибернетики** | | | | | | | | |
| **Кафедра высшей математики** | | | | | | | | |
|  | | | **Утверждаю** | | | |
|  | | | Заведующий  кафедрой\_\_\_\_\_\_\_\_\_\_\_\_\_*Ю.И.Худак* | | | |
|  | | | «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_2019 г. | | | |
| **ЗАДАНИЕ** | | | | | | |
| **на выполнение курсовой работы** | | | | | | |
| **по** **дисциплине** «*Численные методы*» | | | | | | |
|  | | | | | | |
| Студент *Антонов А. Е.* Группа *КМБО-03-16* | | | | | | |
|  | | | | | | |
| 1. **Тема: «Вычисление минимума функции, зависящей от интерполяционного многочлена»** | | | | | | |
| 1. **Исходные данные:**   Вычислить минимум функции на отрезке [a, b] с точностью ε. — значения в точках соответственно интерполяционного многочлена, построенного для таблично заданной функции f(x). | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
| 1. **Срок представления к защите курсовой работы:** **до** « » \_\_\_\_\_\_\_\_\_\_\_ 2019 г. | | | | | | |
|  | | | | | | |
| Задание на курсовую  работу выдал | | «\_\_\_»\_\_\_\_\_\_2019 г. | | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* | | *(\_\_\_\_\_\_\_\_\_\_\_\_)* |
| Задание на курсовую  работу получил | | «\_\_\_»\_\_\_\_\_\_2019 г. | | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* | | *(\_\_\_\_\_\_\_\_\_\_\_\_)* |

**Содержание**

Задание курсовой работы………………………………………………..4

Теоретические сведения…………………………………………………5

Решение…………………………………………………………………...8

Оценка точности…………………………………………………………11

Список литературы………………………………………………………13

Приложение………………………………………………………………14

**Задание курсовой работы**

Вычислить минимум функции на отрезке [a, b] с точностью ε. — значения в точках соответственно интерполяционного многочлена, построенного для таблично заданной функции f(x).

Исходные данные:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | a | b | ε |
| 0.042 | 0.588 | 0 | 2 | 0.0001 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| x | 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 |
| f(x) | 1.859 | 1.852 | 1.851 | 1.848 | 1.842 | 1.833 | 1.822 |

**Теоретические сведения**

**Задача интерполяции**

Пусть функция   задана набором точек  на интервале :

Задача интерполяции – найти функцию , принимающую в точках  те же значения . Тогда,условие интерполяции:

)

При этом предполагается, что среди значений  нет одинаковых. Точки  называют узлами интерполяции.

Если   ищется только на отрезке  – то это задача интерполяции, а если за пределами первоначального отрезка, то это задача **э**кстраполяции.

Задача нахождения интерполяционной функции   имеет много решений, так как через заданные точки можно провести бесконечно много кривых, каждая из которых будет графиком функции, для которой выполнены все условия интерполяции. Для практики важен случай интерполяции функции многочленами:

При этом искомый полином называется интерполяционным полиномом**.**

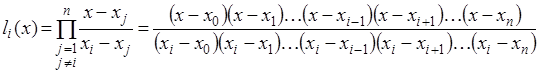
При построении одного многочлена для всего рассматриваемого интервала  для нахождения коэффициентов многочлена необходимо решить систему уравнений, построенную на основе полинома (1.3). Данная система содержит  уравнение, следовательно, с ее помощью можно определить  коэффициент. Поэтому максимальная степень интерполяционного многочлена , и многочлен принимает вид

**Интерполяционный многочлен Лагранжа**

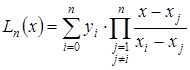
При глобальной интерполяции на всем интервале  строится единый многочлен. Одной из форм записи интерполяционного многочлена для глобальной интерполяции является многочлен Лагранжа:

(1.5)

 где  – базисные многочлены степени *n*:

          (1.6)

То есть многочлен Лагранжа можно записать в виде:

          (1.7)

Многочлен  удовлетворяет условию  Это условие означает, что многочлен равен нулю при каждом  кроме , то есть  – корни этого многочлена. Таким образом, степень многочлена  равна *n* и при  обращаются в ноль все слагаемые суммы, кроме слагаемого с номером , равного .

Погрешность интерполяции методом Лагранжа зависит от свойств функции *,* от расположения узлов интерполяции и точки *x*. Полином Лагранжа имеет малую погрешность при небольших значениях *n* (*n*<20). При больших *n* погрешность начинает расти, что свидетельствует о том, что метод Лагранжа не сходится (то есть его погрешность не убывает с ростом *n*).

Число арифметических операции, необходимых для построения многочлена Лагранжа, пропорционально  и является наименьшим для всех форм записи.

**Погрешность интерполяционной формулы Лагранжа**

При определении значения для функции  с помощью многочлена Лагранжа возникает погрешность или остаточное слагаемое .

Здесь предполагается, что используется глобальный способ интерполяции и что . Последнее предположение требуется для применения соответствующих теорем математического анализа, однако, приведенное ниже соотношение для может использоваться и для сеточных функций.

На основе указанных предположений доказано, что при интерполяции функции , заданной в общем случае на неравномерной сетке интерполяционным многочленом Лагранжа произвольного значения возникает погрешность

, где — многочлен (n+1)-й степени, а . Поскольку точно найти нельзя (из-за неопределенности точки ξ), то при проведении вычислений обычно находятся только приближенные оценки погрешностей интерполяции, которые являются априорными.

Оценка погрешности интерполяции в некоторой произвольной фиксированной точке имеет вид: , где

Оценка максимальной погрешности интерполяции в любой точке , т.е. на всем отрезке:

**Метод золотого сечения нахождения минимума функции**

Пусть задана функция . Тогда для того, чтобы найти неопределённое значение этой функции на заданном отрезке, отвечающее критерию поиска (пусть это будет минимум), рассматриваемый отрезок делится в пропорции золотого сечения в обоих направлениях, то есть выбираются две такие, что

…, г

Таким образом:

и

То есть точка делит отрезок в отношении золотого сечения. Аналогично делит отрезок [ в той же пропорции. Это свойство и используется для построения итеративного процесса.

Шаг 1. Задаются начальные границы отрезка a, b и точность .

Шаг 2. Рассчитывают начальные точки деления: и и значения в них целевой функции .

Если , то

Иначе .

Шаг 3.

Если , то и остановка.

Иначе возврат к шагу 2.

**Решение**

Поставленная задача решается в два этапа.

1. Нахождение значений интерполяционного многочлена, построенного для таблично заданной функции f(x), в точках
2. Вычисление минимума функции .

Рассмотрим первый этап.

Реализуем функцию нахождения значения интерполяционного многочлена Лагранжа в заданной точке.

*# x, y —список точек(x,y) таблично заданной функции,*

*# t — точка, в которой требуется вычислить значение интерполяционного многочлена*

**def** lagranz(x, y, t):

z = 0

**for** j **in** range(len(y)):

nom = 1;

denom = 1

**for** i **in** range(len(y)):

**if** i != j:

nom = nom\*(t - x[i])

denom = denom\*(x[j] - x[i])

z = z + y[j] \* nom / denom

**return** z *# z — значение интерполяционного многочлена в точке t*

Применим эту функцию.

x = np.array([0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6])

f\_x = np.array([1.859, 1.852, 1.851, 1.848, 1.842, 1.833, 1.822])

x1, x2 = 0.042, 0.588

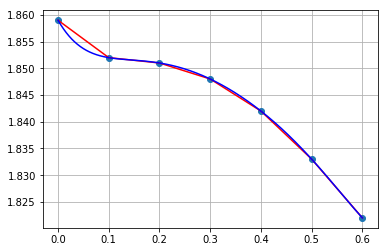
a, b = 0, 2

eps = 0.0001

P\_x1, P\_x2 = lagranz(x, f\_x, x1), lagranz(x, f\_x, x2)

Получим

На рисунке синим изображён график интерполяционного многочлена Лагранжа, а красным — кривая, получившаяся в результате соединения точек отрезками. Можно убедиться, что интерполяционный многочлен построен правильно.



Перейдём ко второму этапу.

Минимум непрерывной точки может быть либо на концах отрезка, либо в точке минимума.

Значения функции на концах отрезка:

.

Реализуем функцию нахождения точки минимума с помощью метода золотого сечения. Для большей эффективности будем использовать вариант алгоритма, при котором на каждой итерации функция будет вычисляться только в одной точке.

*# f — функция, минимум которой хотим найти.*

*# l, r — границы отрезка, на котором ищем минимум.*

*# eps — заданная точность.*

**def** GoldenSectionSearch(f, l, r, eps):

sgr = (math.sqrt(5) - 1) / 2

gr = 1 - sgr

x1, x2 = sgr\*l + gr\*r, gr\*l + sgr\*r

A, B = f(x1), f(x2)

**while** abs(r - l) > eps:

**if** A < B:

r = x2

x2 = x1

x1 = sgr\*l + gr\*r

B = A

A = f(x1)

**else**:

l = x1

x1 = x2

x2 = gr\*l + sgr\*r

A = B

B = f(x2)

x\_ans = (l + r) / 2

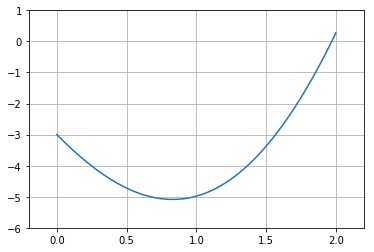
y\_ans = f(x\_ans)

**return** (x\_ans, y\_ans)

В результате работы этой функции получим точку (0.8297606112070215, -5.086268874417563).

Минимум F(x): -5.086268874417563

В правильности можно убедиться, взглянув на график функции.



**Оценка точности**

Поскольку аналитическое выражение для f неизвестно, то найдём конечную разность порядка 6.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| 1.859 |  |  |  |  |  |  |
| 1.852 | -0.007 |  |  |  |  |  |
| 1.851 | -0.001 | 0.006 |  |  |  |  |
| 1.848 | -0.003 | -0.002 | -0.008 |  |  |  |
| 1.842 | -0.006 | -0.003 | -0.001 | 0.007 |  |  |
| 1.833 | -0.009 | -0.003 | 0 | 0.001 | -0.006 |  |
| 1.822 | -0.011 | -0.002 | 0.001 | 0.001 | 0 | 0.006 |

Для

Для

= + + , где — оценка погрешности метода (погрешности интерполирования), — оценка неустранимой погрешности, − оценка округления.

Абсолютную неустранимую ошибку будем считать равной нулю, т. к. полагаем, что в любом узле интерполяции выполняется .

Погрешность округления .

= + + .

Для всей задачи , значит, при нахождении можно производить вычисления с точностью

**Список литературы**

1. Джон Г.Мэтьюз, Куртис Д.Финк. "Численные методы. Использование MATLAB". — М, СПб: "Вильямс", 2001. — 716 с.
2. Галилеев М.М., Гончар Л.И., Грузина Т.Н. Численные методы: учеб. пособие.- СПб.: СПбГИЭУ, 2012. 125 с.
3. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. – 4-е изд. – М.:БИНОМ. Лаборатория знаний, 2006. – 636с
4. В. И. Киреев, А. В. Пантелеев "Численные методы в примерах и задачах", — M, "Высшая школа", 2006. — 480 с.

**Приложение**

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**import** math

*# x, y — списки точек x,y,*

*# t — точка, в которой требуется вычислить значение интерполяционного многочлена*

**def** lagranz(x, y, t):

ans = 0

**for** j **in** range(len(y)):

nom = 1;

denom = 1

**for** i **in** range(len(y)):

**if** i != j:

nom = nom\*(t - x[i])

denom = denom\*(x[j] - x[i])

ans = ans + y[j] \* nom / denom

**return** ans

*# f — функция, минимум которой хотим найти.*

*# l, r — границы отрезка, на котором ищем минимум.*

*# eps — заданная точность.*

**def** GoldenSectionSearch(f, l, r, eps):

sgr = (math.sqrt(5) - 1) / 2

gr = 1 - sgr

x1, x2 = sgr\*l + gr\*r, gr\*l + sgr\*r

A, B = f(x1), f(x2)

num\_of\_iter = 0

**while** abs(r - l) > eps:

num\_of\_iter += 1

**if** A < B:

r = x2

x2 = x1

x1 = sgr\*l + gr\*r

B = A

A = f(x1)

**else**:

l = x1

x1 = x2

x2 = gr\*l + sgr\*r

A = B

B = f(x2)

x\_ans = (l + r) / 2

y\_ans = f(x\_ans)

**print**(num\_of\_iter)

**return** (x\_ans, y\_ans)

x = np.array([0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6])

f\_x = np.array([1.859, 1.852, 1.851, 1.848, 1.842, 1.833, 1.822])

x1, x2 = 0.042, 0.588

a, b = 0, 2

eps = 0.0001

P\_x1, P\_x2 = lagranz(x, f\_x, x1), lagranz(x, f\_x, x2)

**print**('P\_x1 = ', P\_x1, 'P\_x2 = ', P\_x2)

xnew = np.linspace(0, 0.6, 1000)

ynew = [lagranz(x, f\_x, i) **for** i **in** xnew]

plt.plot(x, f\_x, 'o', x, f\_x, 'r', xnew, ynew, 'b')

plt.grid(True)

plt.show()

**def** f(x):

**return** P\_x1 \* x \*\* 2 -2.5 \* P\_x2 \* math.sin(x) - 3

manimum = GoldenSectionSearch(f, a, b, eps)

**print**(manimum)

xx = np.linspace(0, 2, 1000)

yy = [f(x) **for** x **in** xx]

plt.plot(xx, yy)

plt.grid(True)

plt.xlim(-0.2, 2.2)

plt.ylim(-6, 1)

plt.show()