1. **Определение численных методов. Классификация численных методов**
2. **Элементы теории погрешностей. Виды и источники погрешностей**
3. **Абсолютная и относительная погрешности**
4. **Сходимость и устойчивость численных методов**
5. **Классификация численных методов решения нелинейных уравнений**
6. **Общая постановка задачи, локализация корней уравнения**
7. **Уточнение корней методом половинного деления, алгоритм реализации метода**
8. **Программная реализация метода половинного деления, пример**
9. **Уточнение корней методом Ньютона, алгоритм реализации метода**
10. **Программная реализация метода Ньютона, пример**
11. **Уточнение корней методом хорд, алгоритм реализации метода**
12. **Программная реализация метода хорд, пример**
13. **Аппроксимация и интерполяция данных. Практическое применение. Математические определения**
14. **Классификация методов аппроксимации и интерполяции, общий обзор методов**
15. **Аппроксимация по методу наименьших квадратов. Суть метода**
16. **Вывод системы нормальных уравнений для аппроксимации линейной функцией**
17. **Вывод системы нормальных уравнений для аппроксимации полиномом второго порядка**
18. **Интерполяция полиномом Лагранжа, вывод формулы Лагранжа для линейной, квадратичной и кубической интерполяции**
19. **Вывод формулы Лагранжа для глобальной интерполяции**
20. **Программная реализация интерполяции полиномом Лагранжа**
21. **Тригонометрическая интерполяция. Постановка задачи. Вид тригонометрического полинома**
22. **Формулы для расчета коэффициентов тригонометрического полинома и интерполирующей функции**
23. **Определение численных методов. Классификация численных методов**

численные методы – методы в которых решение сложных математических задач сводится к последовательному выполнению большого числа простых арифметических операций.

+ дают хотя бы какое-то решение

Можно оценить ошибку

-решение всегда приближенное

**Классификация численных методов**

Можно привести следующую классификацию численных методов:

♣ методы решения уравнений;

♣ методы решения систем уравнений;

♣ методы вычисления интегралов;

♣ методы аппроксимации и интерполяции;

♣ методы решения дифференциальных уравнений и систем;

♣ методы оптимизации и т.д.

**Классификация численных методов**

Для алгебраических и трансцендентных уравнений пригодны одни и те же методы уточнения приближенных значений действительных корней:

• метод половинного деления (метод дихотомии);

• метод простых итераций;

• метод Ньютона (метод касательных);

• модифицированный метод Ньютона (метод секущих);

• метод хорд и др.

1. **Элементы теории погрешностей. Виды и источники погрешностей**

При решении любой практической задачи необходимо всегда указывать требуемую точность результата.

В связи с этим необходимо уметь:

• 1) зная заданную точность исходных данных, оценивать точность результата (прямая задача теории погрешностей);

• 2) зная требуемую точность результата, выбирать необходимую точность исходных данных (обратная задача теории погрешностей).

**При численном решении задач приходится оперировать двумя видами чисел – точными и приближенными.**

• К точным числам относятся числа, которые дают истинное значение исследуемой величины.

• К приближенным относятся числа, близкие к истинному значению, причем степень близости и определяется погрешностью вычислений.

**Источники погрешностей**

1) погрешность математической модели;

2) погрешность исходных данных (неустранимая погрешность);

3) погрешность численного метода;

4) вычислительная погрешность

**Погрешность математической модели** возникает из-за стремления обеспечить сравнительную простоту ее технической реализации и доступности исследования.

**Погрешность численного метода (погрешность аппроксимации**), связана, например, с заменой интеграла суммой, с усечением рядов при вычислении функций, с интерполированием табличных значений функциональных зависимостей и т.п.

**Вычислительная погрешность** возникает из-за округления чисел, промежуточных и окончательных результатов счета.

1. **Абсолютная и относительная погрешности**

1. **Абсолютная погрешность** — это разность между точным значением числа х и его приближенным значением а: x = | х – а |.

2. **Относительная погрешность** δх = Δx / |а|.

Относительную погрешность часто выражают в процентах: δх = Δx / |а| 100% . •

х – точное неизвестное значение некоторой величины определяется по формуле х=а±x.

1. **Сходимость и устойчивость численных методов**

При анализе вычислительного процесса одним из важнейших критериев является сходимость численного метода.

Он означает близость полученного численного решения к истинному.

**Последовательность** сходится к точному решению х, если при неограниченном возрастании числа итераций n предел этой последовательности существует и равен = x.

**Устойчивость метода**

• Пусть в результате решения задачи по исходному значению x находится значение искомой величины y.

• Пусть исходная величина имеет абсолютную погрешность dx, а решение имеет погрешность dy.

• Задача называется устойчивой по исходному параметру x, если малое приращение исходной величины dx приводит к малому приращению искомой величины dy.

• Другими словами, малые погрешности в исходных величинах приводят к малым погрешностям в результатах расчетов.

Отсутствие устойчивости означает, что даже незначительные погрешности в исходных данных приводят к большим погрешностям в решении или вовсе к неверному результату.

1. **Классификация численных методов решения нелинейных уравнений**

Для алгебраических и трансцендентных уравнений пригодны одни и те же методы уточнения приближенных значений действительных корней:

*метод половинного деления* (*метод дихотомии*);

*метод простых итераций* ;

*метод Ньютона* *(метод касательных)* ;

*модифицированный метод Ньютона* ( *метод секущих* );

*метод хорд* и др.

Для алгебраических и трансцендентных уравнений пригодны одни и те же методы уточнения приближенных значений действительных корней:

*метод половинного деления* (*метод дихотомии*);

*метод простых итераций* ;

*метод Ньютона* *(метод касательных)* ;

*модифицированный метод Ньютона* ( *метод секущих* );

*метод хорд* и др.

1. **Общая постановка задачи, локализация корней уравнения**

Метод половинного деления

Постановка задачи и алгоритм решения

• Дано нелинейное уравнение: f(x)=0 •

Найти корень уравнения, принадлежащий интервалу \*a,b+, с заданной точностью ε. • Для уточнения корня методом половинного деления последовательно осуществляем следующие операции:

1. **Уточнение корней методом половинного деления, алгоритм реализации метода**

Метод половинного деления

1. Делим интервал пополам: •

t=(a+b)/2

1. В качестве нового интервала изоляции принимаем ту половину интервала, на концах которого функция имеет разные знаки

Для этого:

• a) Вычисляем значение функции f(x) в точках a и t.

• b) Проверяем: если f(a)f(t) < 0, то корень находится в левой половине интервала \*a,b+. Тогда отбрасываем правую половину интервала и делаем переприсвоение b=t.

c) Если f(a)f(t) < 0 не выполняется, то корень находится в правой половине интервала \*a,b+. Тогда отбрасываем левую половину и делаем переприсвоение a=t. В обоих случаях мы получим новый интервал \*a,b+ в 2 раза меньший предыдущего.

Процесс, начиная с пункта 1, циклически повторяем до тех пор, пока длина интервала \*a,b+ не станет равной либо меньшей заданной точности.

1. **Программная реализация метода половинного деления, пример**
2. Описать функцию левой части уравнения f(x)
3. . Задать a, b, eps – границы интервала изоляции и точности вычислений
4. •Организовать цикл реализации метода •
5. Вывести полученные результаты

from math import\*

def f(x): return x\*\*2\*sin(x)

a=12

b=13

eps=0.001

while (b-a)>eps :

t=(a+b)/2.0

if f(t)\*f(a)<0:

b=t

else

a=t

print(t)

1. **Уточнение корней методом Ньютона, алгоритм реализации** **метода**

* Выберем начальную точку x0=b (конец интервала изоляции).
* Находим значение функции в этой точке и проводим к ней касательную, пересечение которой с осью Х дает нам первое приближение корня x1.
* В результате итерационный процесс схождения к корню реализуется рекуррентной формулой
* Процесс поиска продолжаем до тех пор, пока не выполнится условие:

1. **Программная реализация метода Ньютона, пример**

1.Описать функцию левой части уравнения f(x)=0 и функции первой производной 2. Задать x1, x2, eps –первое и второе приближения корня и точности вычислений

3. Организовать цикл реализации метода

4 Вывести полученные результаты

x0 = (a + b) / 2

xn = F(x0)

xn1 = xn - F(xn) / F1(xn)

while abs(xn1 - xn) > math.pow(10, -6):

xn = xn1

xn1 = xn - F(xn) / F1(xn)

return xn1

1. **Уточнение корней методом хорд, алгоритм реализации метода**

Входные данные: f(x), f’’(x), a, b, ε.

Если f(a)·f ”(a)>0, то c=a, иначе если f(b)·f ’’(b)>0, то c=b.

Если f(a)·f’’(a)<0, то x1=a, иначе если f(b)·f ’’(b)<0, то x1=b

dx=f(x1)·(x1-c)/(f(x1)-f(c)).

x1=x1-dx

Если |dx|>ε, то идти к 3.

1. **Программная реализация метода хорд, пример**

a =4

b = 3

eps = 0.00001

while abs(b - a) > eps:

a = b - (b - a) \* f(b) / (f(b) - f(a))

b = a + (a + b) \* f(a) / (f(a) - f(b))

1. **Аппроксимация и интерполяция данных. Практическое применение. Математические определения**

**Практическое применение**

Из всех способов задания зависимостей наиболее удобным является аналитический способ задания зависимости в виде функции n=f(Мкр.), P=f(t), y=f(t). В результате проведения эксперимента **табличную функцию.**

Можно ли проинтегрировать или продифференцировать табличную функцию?

При такой постановке задачи моделирования нужно заменить табличную функцию аналитической. Для этой цели используются методы ***аппроксимации и интерполяции*.**

**Математические определения**

**Аппроксимация** –это замена исходной функции f(x) функцией φ(x)так, чтобы отклонение f(x) от φ(x)в заданной области было наименьшим. Функция φ(x)называется аппроксимирующей.

Если исходная функция f(x) задана таблично (дискретным набором точек), то аппроксимация называется ***дискретной***

Если исходная функция f(x) задана аналитически (на отрезке), то аппроксимация называется ***непрерывной или интегральной***

***Интерполяция*** –это замена исходной функции f(x) функцией φ(x)так, чтобы φ(x)точно проходила через точки исходной функции f(x). Интерполяция еще называется ***точечной аппроксимацией***

Точки исходной функции f(x)называются ***узлами интерполяции***

Экстраполяцией называетсяаппроксимация вне заданной области определения исходной функции, т.е **x < x0 и x > xn**

Найдя интерполяционную функцию, мы можем вычислить ее значения между узлами интерполяции, а также определить значение функции за пределами заданного интервала (**провести экстраполяцию**).

1. **Классификация методов аппроксимации и интерполяции, общий обзор методов**

Если интерполяция проводится только для отдельных участков отрезка [а, b] (области определения f(х)), т.е. для mинтерполяционных узлов, где m < n, то интерполяцию называют **локальной**.

Если интерполяционный многочлен ищется для всего интервала области определения x, т.е. для [x0, xn], то интерполяция называется **глобальной.**

Простейшими видами локальной интерполяции является **линейная и квадратичная**

При линейной интерполяции точки заданной функции соединяются линейными отрезками, а при квадратичной –отрезками парабол.

Такая интерполяция еще называется **параболической**.

**Типовые виды глобальной интерполяции**

* Интерполяционный многочлен Лагранжа
* Интерполяционный многочлен Ньютона
* Сплайны

Сплайн представляет собой математическую модель гибкого тонкого стержня из упругого материала, закрепленного в двух соседних узлах интерполяции с заданными углами наклона αи βтак, чтобы потенциальная энергия стержня была минимальна

Сплайн –функция, которая вместе с несколькими производными непрерывна на всем заданном отрезке [a,b], а на каждом частичном отрезке [ xi,xi+1] в отдельности является некоторым алгебраическим многочленом.

Наиболее известными методами аппроксимации являются метод наименьших квадратов, метод многочленов Чебышева, рядов Тейлор и т.д.

**При решении задач аппроксимации часто используются функции регрессии**

**Регрессия**–представление совокупности данных некоторой функцией **f(x)**.

Задачей регрессии является вычисление параметров функции **f(x)**таким образом, чтобы функция приближала последовательность исходных точек с наименьшей погрешностью.

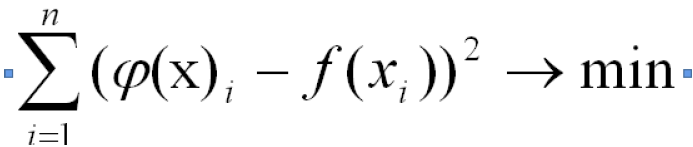
При этом функция **f(x)**называется **уравнением регрессии**.

При регрессии не требуется чтобы функция проходила через все заданные точки, что особенно важно при аппроксимации данных, заведомо содержащих ошибки.

1. **Аппроксимация по методу наименьших квадратов. Суть метода**

Метод наименьших квадратов (МНК), идея которого принадлежит А.Лежандру, а теоретическое обоснование К.Гауссу. В соответствии с этим методом, оценки параметров ai определяют из условия минимума суммы квадратов отклонений измеренных значений yi от соответствующей ординаты рассмотренной кривой

Критерий метода наименьших квадратов

****Основной мерой отклонения аппроксимирующей функции φ(x)от исходной функции f(x) при аппроксимации является величина, равная сумме квадратов разностей между значениями аппроксимирующей и исходной функций

**Алгоритм**

•1. Ввести два одномерных массива xи yперечислением элементов

•2. Вычислить суммы (какие?)

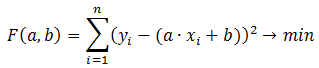
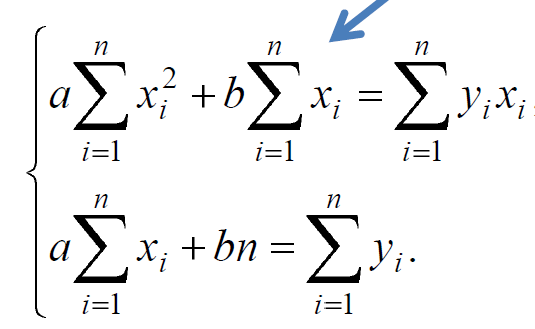
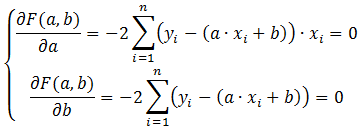
•3. Вычислить aи bпо формулам

•4. Создать одномерный массив аргумента аппроксимирующей функции, например, x1, содержащий числа в пределах изменения x, но их количество должно быть гораздо больше (50-100 чисел)

•5. Рассчитать значения аппроксимирующей функцииy1 для каждого значения x1

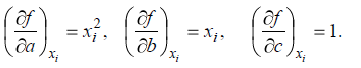
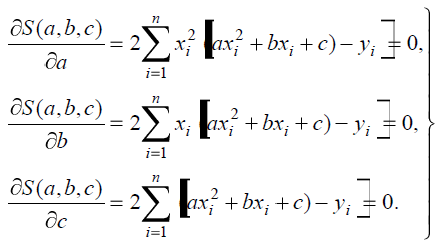
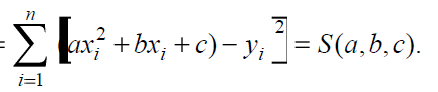
•6. Построить график исходной функции в виде точек, не соединенные отрезками прямых и помеченные маркером и график аппроксимирующей функции

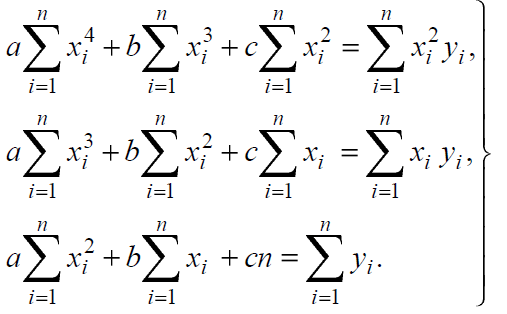
1. **Вывод системы нормальных уравнений для аппроксимации линейной функцией**

y = ax+b.  
Аппроксимация заключается в отыскании коэффициентов a и b уравнения таких, чтобы все экспериментальные точки лежали наиболее близко к аппроксимирующей прямой.  


1. **Вывод системы нормальных уравнений для аппроксимации полиномом второго порядка**

*,*  ,



Аналогично тому, как это сделано для аппроксимации прямой, получим систему нормальных уравнений:

1. **Интерполяция полиномом Лагранжа, вывод формулы Лагранжа для линейной, квадратичной и кубической интерполяции**

многочлен Лагранжа:

**Линейная интерполяция**•При **n=1**формула Лагранжа имеет вид:

**Квадатическая интерполяция** •При **n=2**формула Лагранжа имеет вид:

**Кубическая интерполяция** •При **n=3**формула Лагранжа имеет вид:

1. Вывод формулы Лагранжа для глобальной интерполяции
2. **Программная реализация интерполяции полиномом Лагранжа**

|  |  |
| --- | --- |
| import matplotlib.pyplot as plt  import numpy as np  from math import\*  x=np.array([-1,0,1])  y=np.array([-0.8, 1.6, 2.3])  n = len(x)  xt=0.5  Yt=0 | for j in range(0, n):  L=1  for i in range (0,n):  if i!=j :  L=L\*(xt-x[i])/(x[j]-x[i])  Yt=Yt+y[j]\*L  print(Yt) |

1. **Тригонометрическая интерполяция. Постановка задачи. Вид тригонометрического полинома**

**Тригонометрическим полиномом** от переменной ***x*** называется выражение вида(1)

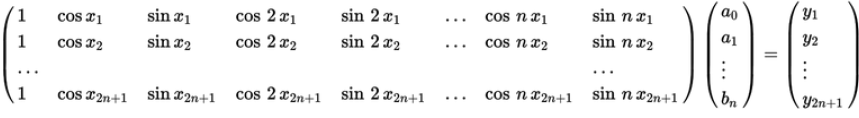
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | X1 | X2 | … | x(2n+1) |
| Y | Y1 | Y2 | … | y(2n+1) |

Постановка задачи. Построить тригонометрический полином порядка n, принимающий значения по следующей таблице

Интерполирующий полином имеет вид

В узловых точках

Для получения коэффициентов интерполирующего полинома нужно решить систему 2n+1 линейных уравнений



1. **Формулы для расчета коэффициентов тригонометрического полинома и интерполирующей функции**

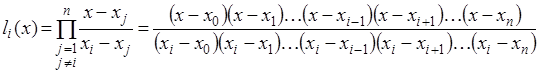
Если в качестве узлов интерполяции взять равноотстоящие

то коэффициенты интерполяционного тригонометрического полинома находятся по формулам:

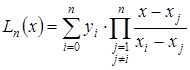
При глобальной интерполяции на всем интервале  строится единый многочлен. Одной из форм записи интерполяционного многочлена для глобальной интерполяции является многочлен Лагранжа:

          (3.11)

где  – базисные многочлены степени *n*:

          (3.12)

То есть многочлен Лагранжа можно записать в виде:

          (3.13)

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 .

Выражение (3.11) применимо как для равноотстоящих, так и для не равноотстоящих узлов. 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*).

Многочлен Лагранжа в явном виде содержит значения функций в узлах интерполяции, поэтому он удобен, когда значения функций меняются, а узлы интерполяции неизменны. Число арифметических операции, необходимых для построения многочлена Лагранжа, пропорционально  и является наименьшим для всех форм записи. К недостаткам этой формы записи можно отнести то, что с изменением числа узлов приходится все вычисления проводить заново.

Кусочно-линейная и кусочно-квадратичная локальные интерполяции являются частными случаями интерполяции многочленом Лагранжа.