Решение нелинейных уранений

Постановка задачи

Дана функция $f(x)$ непрерывная на некотором промежутке [a;b].

Надо: найти корни уравнения

$f(x)=0$ (1) с заданной точностью ε.

Если $f(x)$ - многочлен, то это алгебраическое уравнение,

иначе трансцендентное.

Методы решения

- ***прямые методы*** решения,

(формулы для квадратных уравнений или Кардано для кубических уравнений)

- ***итерационные***, когда решение получается путем многократного применения какого-то алгоритма.

В этом случае решение задачи разбивается на *два этапа*:

I. отделение корней;

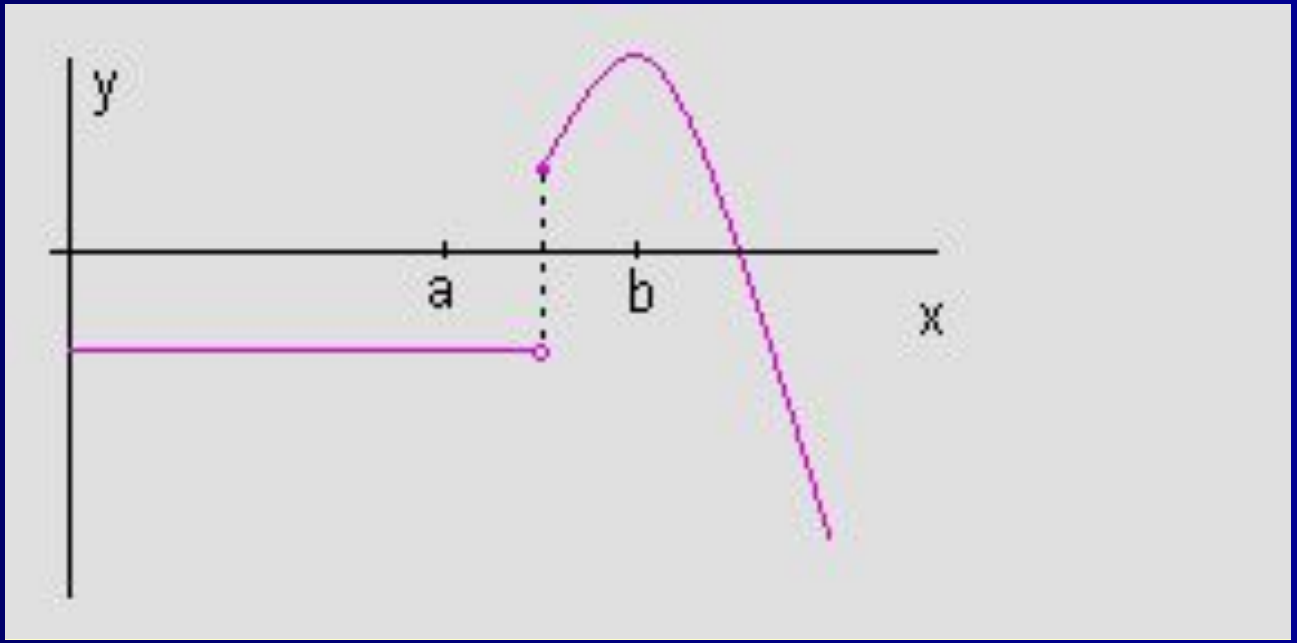
II. уточнение отделенного корня с заданной точностью.

Отделение корней

Необходимо выяснить, имеет ли уравнение (1) действительные корни.

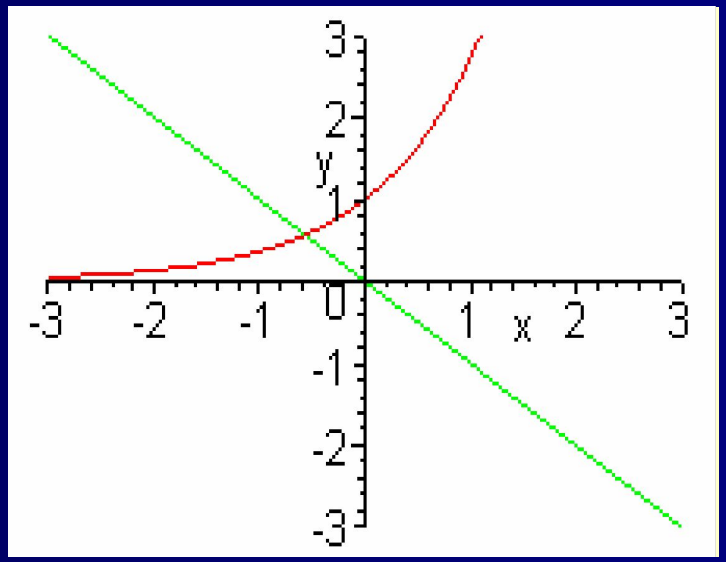
Будем исходить из теоремы о том, что если непрерывная функция на отрезке принимает значения разных знаков , то внутри отрезка существует хотя бы один корень.

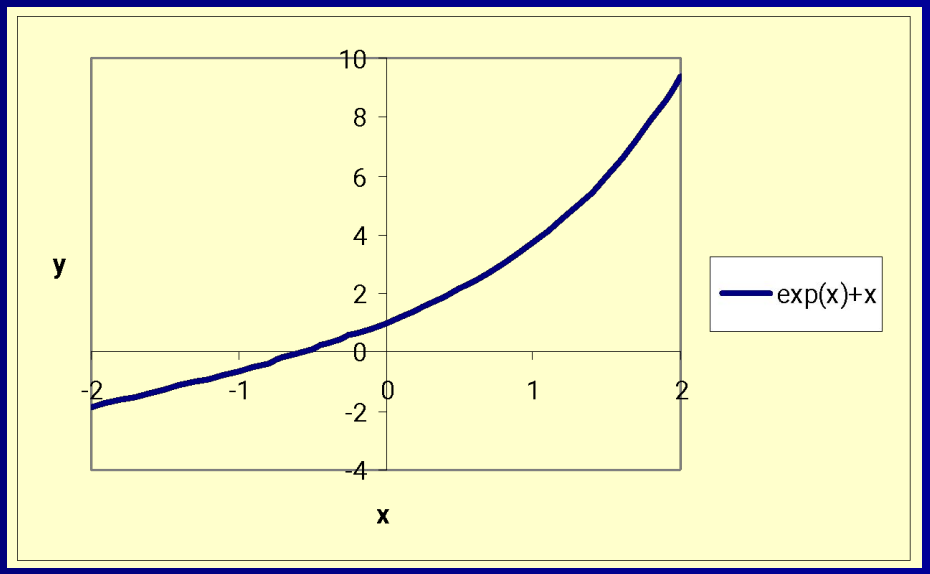
В данном случае условие непрерывности обязательно:



Решить вопрос о существовании корней уравнения можно, построив график функции:

Пример. Дано уравнение $e^x + x = 0$





Отделение корней

Отделить корни – это значит указать промежуток $[a, b]$, содержащий ***один и только*** один корень уравнения (1). Иначе говорят: указать отрезки изоляции корней.

Способы отделения корней

Аналитический. Исследуется функция и ее производная, находятся критические точки и интервалы изменения знака функции, учитывается поведение производной.

+

Дает гарантированный результат по отделению всех действительных корней уравнения

—

Требует нахождения производной, трудоемкий, рассчитан на «ручное» решение

Способы отделения корней

Графический. По графику функции находят отрезки изоляции корней.

\+

Дает наглядный и чаще всего достоверный результат

—

- Ограниченность интервала построения графика
- Программное построение графика может быть равнозначно решению исходной задачи

Способы отделения корней

Табличный. Значения вычисляются в ряде промежуточных точек ***f(x)***. Например, отрезок **[a;b]** разделим на ***n*** частей с шагом h. Фиксируются те интервалы $[x_i ; x_{i+1}]$, на которых функция меняет знак.

+

Легко программируемый алгоритм

—

Требует оптимального подбора шага для того, чтобы не пропустить корни, но и не слишком увеличивать время расчета

Список в Python

```
1  '''
2   Список - динамическая коллекция упорядоченных данных
3  '''
4  L = [123, '0b1111011', '0o173', '0x7b']
5
6  print(L[0], L[len(L)-1])
7
8  '''
9   Список содержит ссылки на объекты
10 '''
11 L1 = [1, 2, 3, 4, 5]
12
13 L2 = L1
14 L1[2] = 0
15
16 print(L2)
```

Результат: 123 0x7b

Результат: [1, 2, 0, 4, 5]

Генерация последовательностей

```
"""
range([start=0], stop, [step=1]) -
арифметическая прогрессия от start до stop с шагом step

"""
for i in range(7):
    print(i, end = ' ')
print()

for i in range(10, 20, 2):
    print(i, end = '; ')

print()
for i in range(10, 1, -3):
    print(i, end = '\t')

```

Результат:

0 1 2 3 4 5 6

10; 12; 14; 16; 18;

10	7	4

Библиотека NumPy: arange( )

NumPy предоставляет объект многомерного массива и набор подпрограмм для быстрых операций с

```
1  import numpy as np
2
3  # генерация вещественной последовательности
4  x = np.arange(0.3, 1.6, 0.3)
5
6  print(x)
7  print(type(x))
```

Результат:
[0.3 0.6 0.9 1.2 1.5]
<class 'numpy.ndarray'>

основы линейной алгебры, базовые статистические операции, случайное моделирование и многое другое.

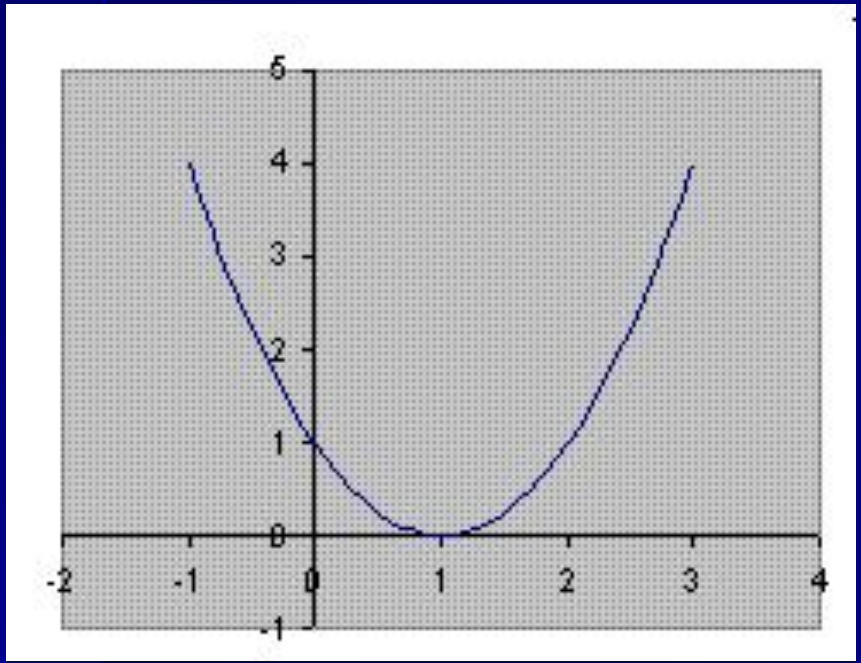
Реализация отделения корней

```
1 #отделение корней
2
3 def fun(x):
4     return x**3-5*x+1
5
6 x_min = -3
7 x_max = 3
8
9 h = 1
10 l = []
11
12 for x in range(x_min,x_max,h):
13     if fun(x)*fun(x+h)<0:
14         l.append([x,x+h])
15 if len(l)>0:
16     print('Отрезки изоляции корней',l)
17 else:
18     print('В диапазоне',[x_min,x_max],'корней нет')
```

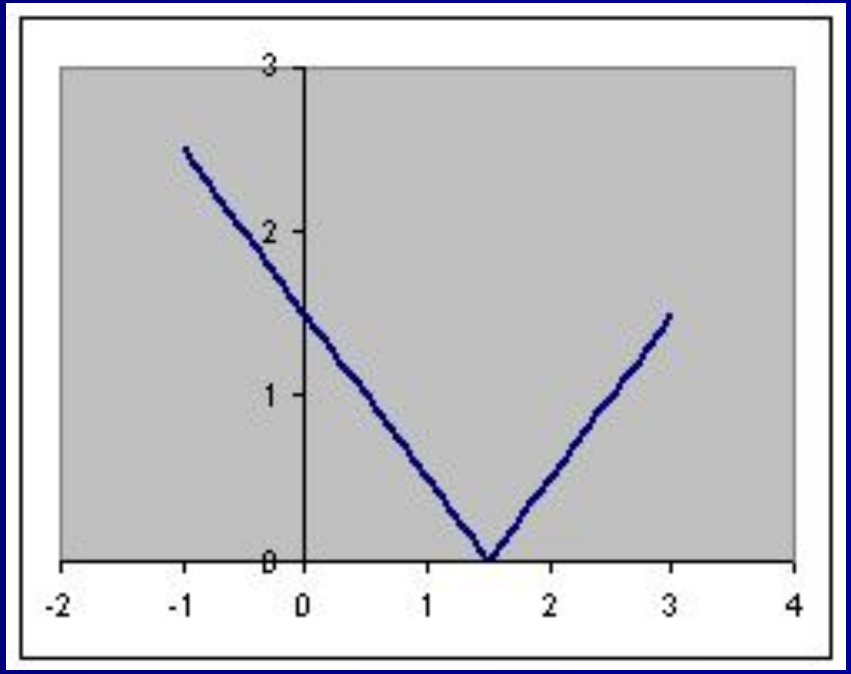
Результат:
Отрезки изоляции корней [[-3, -2], [0, 1], [2, 3]]

Исключения

f’(x)=0 в корне



f’(x) в корне не существует



Пример.

$$(x-1)(x-2)^2(x-3)^3 = 0$$

```
1  #отделение корней
2
3  def fun(x):
4      return (x - 1)*(x - 2)**2*(x - 3)**3
5
6  x_min = -3
7  x_max = 3
8
9  h = 1
10 l = []
11
12 for x in range(x_min,x_max,h):
13     if fun(x)*fun(x+h)<0:
14         l.append([x,x+h])
15 if len(l)>0:
16     print('Отрезки изоляции корней',l)
17 else:
18     print('В диапазоне',[x_min,x_max],'корней нет')
```

Результат:
В диапазоне [-3, 3] корней нет

Отделение корней

Результатом отделения корня является промежуток $[a; b]$ с единственным на нем корнем уравнения (1).

Предполагается в дальнейшем, что функция $f(x)$ на этом отрезке знак меняет, а производная $f''(x)$ знак сохраняет.

Итерационные методы

После того, как корни локализованы, задача сводится к уточнению корня уравнения $f(x)=0$ на промежутке $[a;b]$ с заданной точностью ε.

В итерационных методах для этого строится последовательность:

$$x_0, x_1, x_2, \ldots, x_i, \ldots ,$$

сходящаяся к точному решению x^*.

Типы сходимостей итерационных последовательностей

Скоростью сходимости итерационного метода называется скорость убывания величины $|x^*-x_i|$.

Если $|x^*-x_{i+1}| \le q\,|x^*-x_i|$, то имеет место ***линейная*** сходимость.

Если $|x^*-x_{i+1}| \le q\,|x^*-x_i|^{\alpha}$, $1<\alpha<2$, то имеет место ***сверхлинейная*** сходимость.

Если $|x^*-x_{i+1}| \le q\,|x^*-x_i|^{2}$, то имеет место ***квадратичная*** сходимость.

Достижение точности

Применяя итерационный метод, получаем последовательность $x_1, x_2, x_3, \ldots x_i, \ldots$, сходящуюся к точному решению уравнения.

Возникает вопрос, какой из элементов этой последовательности принять за приближенное значение с точностью ε и прекратить итерационный процесс уточнения корня?

Оценка точности приближения

может характеризоваться следующими величинами:

- Невязка $f(x_i)$
- Ошибка $x^* - x_i$
- Поправка $x_{i+1} - x_i$

Любая из этих величин может учитываться в условии окончания итерационного процесса приближения к корню.

Невязка $f(x_i)$

Условие достижения заданной точности:

$$|f(x_i)| < \varepsilon$$

- Процесс масштабирования может привести к потере точности: достаточно умножить обе части уравнения на число, соразмерное с ε, и процесс уточнения завершиться далеко от корня.

Ошибка $x^* - x_i$

Условие достижения заданной точности:

$$| x^* - x_i | < \varepsilon$$

— Требует знания корня x^* .

Поправка $x_{i+1} - x_i$

Условие достижения заданной точности:

$$|x_{i+1} - x_i| < \varepsilon$$

Требует доказательства того, что

$$|x_{i+1} - x_i| < |x^* - x_i|.$$