1. **Введение**

Методы решения уравнений делятся на прямые и итерационные. **Прямые методы** – это методы, позволяющие вычислить решение по формуле. Например, нахождение корней квадратного или кубического уравнения. **Итерационные методы** − это методы, в которых задается некоторое начальное приближение и строится сходящаяся последовательность приближений к точному решению, причем каждое последующее приближение вычисляется c использованием предыдущих.

Численное решение нелинейного уравнения заключается в вычислении с заданной точностью значения всех или некоторых корней уравнения и распадается на несколько задач: во-первых, надо исследовать количество и характер корней, во-вторых, определить их приближенное расположение, т.е. значения начала и конца отрезка, на котором лежит только один корень, в-третьих, выбрать интересующие нас корни и вычислить их с требуемой точностью. Вторая задача называется отделением корней. Решив ее, по сути дела, находят приближенные значения корней с погрешностью, не превосходящей длины отрезка, содержащего корень. Отметим два простых приема отделения действительных корней уравнения - табличный и графический. Первый прием состоит в вычислении таблицы значений функции f(x) в заданных точках xi и использовании следующих теорем математического анализа:

1. Если функция непрерывна на отрезке и , то внутри отрезка существует по крайней мере один корень уравнения .
2. Если функция непрерывна на отрезке , и на интервале сохраняет знак, то внутри отрезка существует единственный корень уравнения.
3. Отделение корней заданного уравнения графически

Рассмотрим задачу нахождения корней нелинейного уравнения

|  |  |
| --- | --- |
|  | (2.1) |

Корнями уравнения (2.1) называются такие значения х, которые при подстановке обращают его в тождество. Только для простейших уравнений удается найти решение в виде формул, т.е. аналитическом виде. Чаще приходится решать уравнения приближенными методами, наибольшее распространение среди которых, в связи с появлением компьютеров, получили численные методы.

Алгоритм нахождения корней приближенными методами можно разбить на два этапа. На первом изучается расположение корней и проводится их разделение. Находится область , в которой существует корень уравнения или начальное приближение к корню .

Существование на найденном отрезке ,, по крайней мере, одного корня уравнения (5.1) следует из условия Больцано:

|  |  |
| --- | --- |
|  | (2.2) |

При этом подразумевается, что функция непрерывна на данном отрезке.

Для графического отделения корней надо построить график функции , по которому можно судить, в каких интервалах находятся его точки пересечения с осью Ox.

|  |  |
| --- | --- |
| 9 | 4 (1 + x0.5 ) ln(x) − 1 = 0 |

Запишем исходное уравнение в виде: ln(x) = 1/(4 (1 + x0.5 )), т. е.

φ = ln(x), ψ= 1/(4 \*(1 + x0.5 ))

Таким образом, корни данного уравнения могут быть найдены как абсциссы точек пересечения кривых y = ln(x), y = 1/(4 \*(1 + x0.5 )).

Теперь построим графики функций и определим интервал изоляции корня.

Рис. 2.1 – Построение графика заданных функций

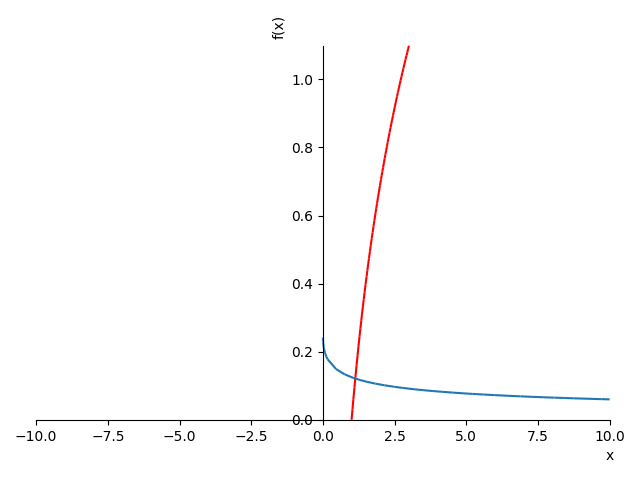


Рис. 2.2 – Скриншот программы (построение графика заданных функций)

1. Решить уравнение методом хорд

Пусть дано уравнение − дважды непрерывно дифференцируемая функция.

Пусть выполняется условие (2.2) и проведено отделение корней, т.е. на данном интервале (*a, b*) находится один корень уравнения. При этом, не ограничивая общности, можно считать, что  *.*

Пусть функция *f* выпукла на интервале (рис. 3.1).

  
Рис. 3.1

Заменим график функции хордой (прямой), проходящей через точки .

Уравнение прямой, проходящей через две заданные точки, можно записать в виде . В нашем случае получим .

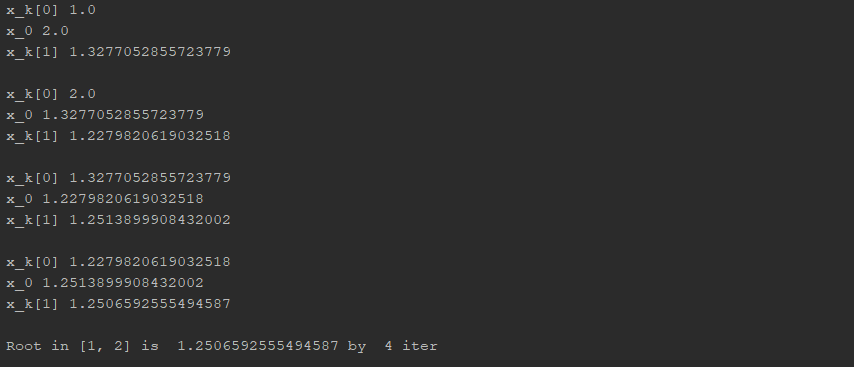
Найдем точку пересечения хорды с осью Oх.

Полагая , получаем из предыдущего уравнения:

Теперь возьмем интервал в качестве исходного и повторим вышеописанную процедуру (рис. 3.1). Получим

Продолжим процесс. Каждое последующее приближение вычисляется по рекуррентной формуле

Наша функция вогнута, поэтому



Заменить

Рис. 3.2 – Программная реализация метода хорд

1. Решить уравнение методом касательных

Пусть дано уравнение − дважды непрерывно дифференцируемая функция.

Если выполняется условие (2.2), то на данном интервале содержится корень уравнения. Предположим, что корень отделен, т.е. на данном интервале он только один. Не ограничивая общности, можно считать, что .



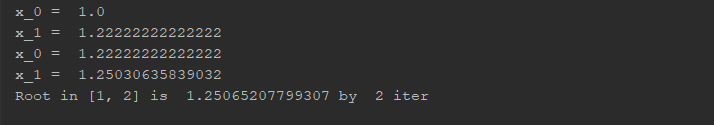
Рис. 4.1

Запишем уравнения касательной в точке :

Найдем точку ее пересечения с осью Ох. Получаем

Построив касательную в точке , находим точку ее пересечения с осью Ох:

Продолжая процесс, получим



заменить

Рис. 4.2 – Программная реализация метода касательных

1. Отделить корни заданной системы нелинейных уравнений графически

|  |  |
| --- | --- |
| 9 | {sin(x + y) + 1.2x = 1  {x2 + y2 = 1 |

Выразим x и y:

Построим графики этих функций:

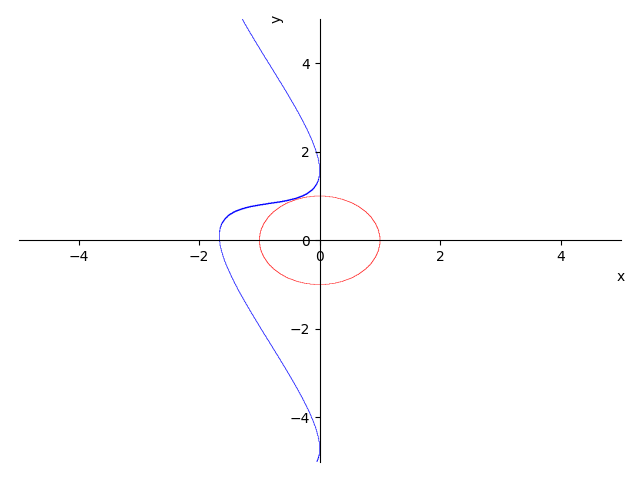


Рис. 5.1 – Построение графиков

1. Построить рабочие формулы метода простых итераций, метода Ньютона, реализующие процесс поиска корня системы нелинейных уравнений на найденном отрезке

Пусть есть система n уравнений с n неизвестными

Запишем ее в векторном виде: , где

,   
  
Метод простых итераций

Система преобразуется у виду:

, где

Запишем итерацию:

Метод Ньютона

Рассмотрим снова систему:

Пусть – начальное приближение. Предположим, что в окрестности начального приближения матрица Якоби

не вырождена, т.е. .

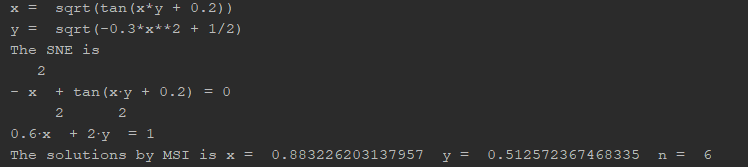
.

Отсюда

Получим:

1. Решить систему методом простых итераций

|  |  |
| --- | --- |
| 9 | {sin(x + y) + 1.2x = 1  {x2 + y2 = 1 |
|  |  |

Начальное приближение ,   


заменить

Рис. 7.1 – Программная реализация метода простых итераций для системы

Сверим результаты при помощи встроенной функции:



1. Решить систему методом Ньютона

Начальное приближение ,



заменить

1. **Решить систему модифицированным методом Ньютона**

Модифицируем метод Ньютона, изменив итерационную формулу. Для это будем вычислим матрицу Якоби лишь при начальном приближении



заменить