Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №2
Задание №1 часть I по курсу «Численные методы»
Вариант №12

Студент: С. М. Власова
Преподаватель: И. Э. Иванов
Группа: М8О-306Б
Дата:
Оценка:
Подпись:

Москва, 2020

Задание №2.1

Реализовать метод Ньютона решения нелинейных уравнений в виде программы, задавая в качестве входных данных точность вычислений. С использованием разработанного программного обеспечения найти положительный корень нелинейного уравнения (начальное приближение определить графически). Проанализировать зависимость погрешности вычислений от количества итераций.
Вариант: 12

$3^x - 5 \cdot x^2 + 1 = 0.$

1 Описание метода решения

Численное решение нелинейных (алгебраических или трансцендентных) уравнений вида

$$f(x) = 0$$

заключается в нахождении значений x, удовлетворяющих с заданной точностью данному уравнению и состоит из следующих основных этапов

1. *Отделение* (изоляция, локализация) корней уравнения.
2. *Уточнение* с помощью некоторого вычислительного алгоритма конкретного выделенного корня с заданной точностью.

Целью первого этапа является нахождение отрезков из области определения функции $f(x)$, внутри которых содержится только один корень решаемого уравнения. Иногда ограничиваются рассмотрением лишь какой-нибудь части области определения, вызывающей по тем или иным соображениям интерес. Для реализации данного этапа используются *графические* и *аналитические* способы.

При аналитическом способе отделения корней полезна следующая теорема.

Теорема 2. 1 *Непрерывная строго монотонная функция $f(x)$ имеет и притом единственный нуль на отрезке $[a, b]$ тогда и только тогда, когда на его концах она принимает значения разных знаков.*

Достаточным признаком монотонности функции $f(x)$ на отрезке $[a, b]$ является сохранение знака производной функции.
Графический способ отделения корней целесообразно использовать в том случае, когда имеется возможность построения графика функции $y = f(x)$. Наличие графика исходной функции дает непосредственное представление о количестве и расположении нулей функции, что позволяет определить промежутки, внутри которых содержится только один корень. Если построение графика функции $y = f(x)$ вызывает затруднение, часто оказывается удобным преобразовать уравнение к эквивалентному виду $f_1(x) = f_2(x)$ и построить графики функций $y = f_1(x)$ и $y = f_2(x)$. Абсциссы точек пересечения этих графиков будут соответствовать значениям корней решаемого уравнения.

Так или иначе, при завершении первого этапа, должны быть определены промежутки, на каждом из которых содержится только один корень уравнения. Для уточнения корня с требуемой точностью обычно применяется какой-либо итерационный метод, заключающийся в построении числовой последовательности $x^{(k)}$ $(k = 0, 1, 2, \ldots)$, сходящейся к искомому корню $x^{(\star)}$ исходного уравнения.

Метод Ньютона (метод касательных). При нахождении корня уравнения методом Ньютона, итерационный процесс определяется формулой

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}, \qquad k = 0, 1, 2, \ldots$$

Для начала вычислений требуется задание начального приближения $x^{(0)}$.
Условия сходимости метода определяются следующей теоремой.

Теорема 2. 2 *Пусть на отрезке $[a, b]$ функция $f(x)$ имеет первую и вторую производные постоянного знака и пусть $f(a) \cdot f(b) < 0$.*
Тогда, если точка $x^{(0)}$ выбрана на $[a, b]$ так, что

$$f(x^{(0)}) \cdot f''(x^{(0)}) > 0,$$

то начиная с нее последовательность $x^{(k)}$ $(k = 0, 1, 2, \ldots)$, определяемая методом Ньютона, монотонно сходится к корню $x^{(\star)} \in (a, b)$ уравнения $f(x) = 0$.

В качестве условия окончания итераций в практических вычислениях часто используется правило $|x^{(k+1)} - x^{(k)}| < \varepsilon \Rightarrow x^{(\star)} \approx x^{(k+1)}$.

2 Моя задача

Решить уравнение

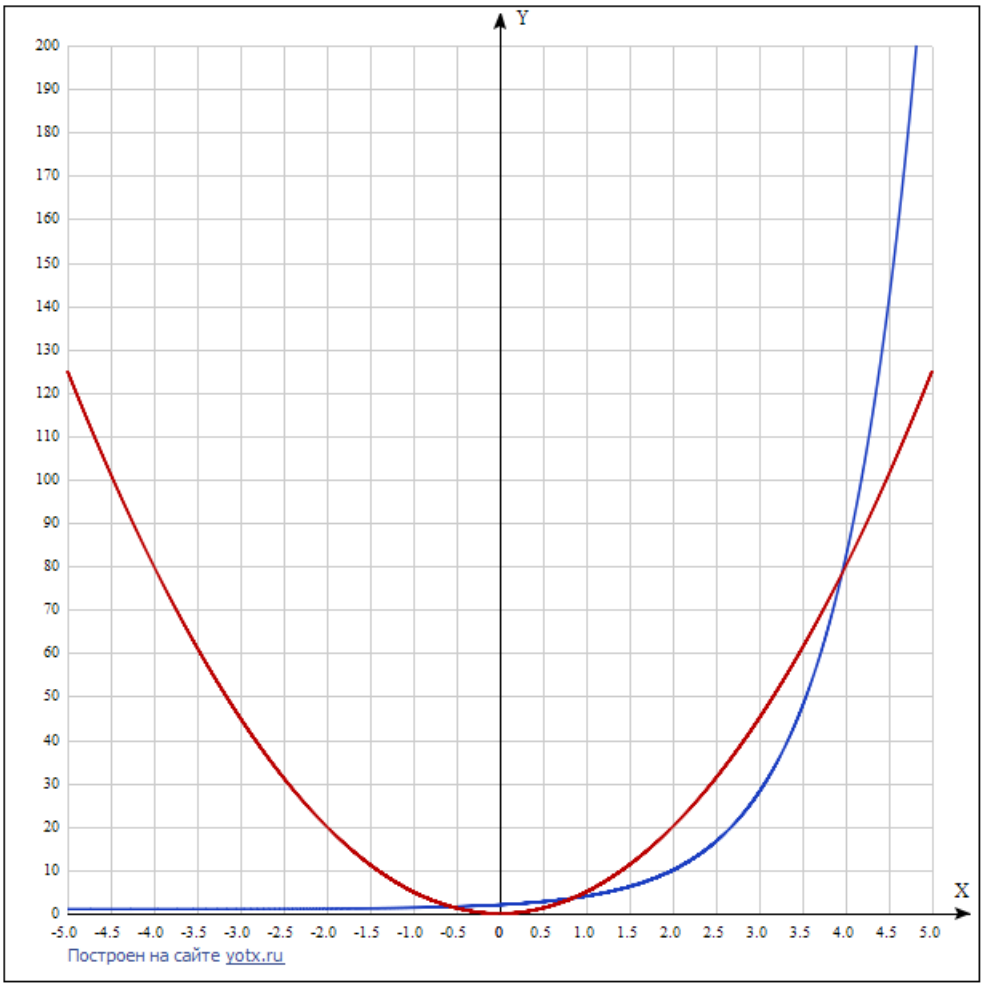
$$f(x) = 3^x - 5 \cdot x^2 + 1 = 0$$

с заданной точностью.

Решение:
Этап I. Локализация: для локализации корней применим графический способ. Преобразуем исходное уравнение к эквивалентному виду:

$$3^x + 1 = 5 \cdot x^2$$

Теперь построим графики функций $f_1(x) = 3^x + 1$ и $f_2(x) = 5 \cdot x^2$.



Графики функций пересекаются в трех точках, соответственно, уравнение имеет три

корня — два положительных и один отрицательный. Т.к. нас интересуют исключительно положительные корни, локализуем их.

$$0.5 < x_1^{(\star)} < 1$$

$$3.5 < x_2^{(\star)} < 4.5$$

Этап II. Уточнение: для уточнения корней будем использовать метод Ньютона. Для этого необходимо определить поведение первой и второй производных функции $f(x)$ на интервале уточнения корня и правильно выбрать начальное приближение $x^{(0)}$ для обоих корней.

Для $f(x) = 3^x - 5 \cdot x^2 + 1 = 0$ имеем

$$f'(x) = 3^x - 10 \cdot x$$

$$f''(x) = 3^x - 10$$

1. На интервале уточнения корня $x_1^{(\star)}$ первая и вторая производные принимают *отрицательные* значения.
   В качестве начального приближения я взяла правую границу интервала $x_1^{(0)} = 1$, т.к. значение $f(1) \cdot f''(1) > 0$.

2. На интервале уточнения корня $x_2^{(\star)}$ первая и вторая производные принимают *положительные* значения.
   В качестве начального приближения я взяла правую границу интервала $x_2^{(0)} = 4.5$, т.к. значение $f(4.5) \cdot f''(4.5) > 0$.

Следовательно, условия сходимости метода Ньютона выполняются.

3 Протокол

Входные данные через командную строку я задаю значение точности ε.
Выходные данные я вывожу на экран.
Скриншот консоли:
$\varepsilon = 0.1$.

```
(base) vlasochka@vlasochka-VPCSB11FX:~/Документы/ЧМ/Lab2$ g++ Newton2.1.cpp -o newton2.1
(base) vlasochka@vlasochka-VPCSB11FX:~/Документы/ЧМ/Lab2$ ./newton2.1
Введите точность вычислений: 0.1
1-й положительный корень нелинейного уравнения, найденный методом Ньютона: 0.838793
Число итераций: 3
2-й положительный корень нелинейного уравнения, найденный методом Ньютона: 3.95927
Число итераций: 3
(base) vlasochka@vlasochka-VPCSB11FX:~/Документы/ЧМ/Lab2$
```

$\varepsilon = 0.01$.

```
(base) vlasochka@vlasochka-VPCSB11FX:~/Документы/ЧМ/Lab2$ ./newton2.1
Введите точность вычислений: 0.01
1-й положительный корень нелинейного уравнения, найденный методом Ньютона: 0.837978
Число итераций: 4
2-й положительный корень нелинейного уравнения, найденный методом Ньютона: 3.95927
Число итераций: 3
(base) vlasochka@vlasochka-VPCSB11FX:~/Документы/ЧМ/Lab2$
```

$\varepsilon = 0.0001$.

```
(base) vlasochka@vlasochka-VPCSB11FX:~/Документы/ЧМ/Lab2$ ./newton2.1
Введите точность вычислений: 0.0001
1-й положительный корень нелинейного уравнения, найденный методом Ньютона: 0.837943
Число итераций: 5
2-й положительный корень нелинейного уравнения, найденный методом Ньютона: 3.95758
Число итераций: 6
(base) vlasochka@vlasochka-VPCSB11FX:~/Документы/ЧМ/Lab2$
```

Можно заметить, что сходимость метода Ньютона очень хорошая даже при малом значении оценки погрешности. При $\varepsilon = 0.0001$ положительные корни исходного уравнения следующие

$$x_1^{(\star)} \approx 0.837943$$

$$x_2^{(\star)} \approx 3.95758$$

Алгоритм находит их за 5-6 итераций.

4 Исходный код

Листинг 1: Метод Ньютона

```
#include<iostream>
#include<map>
#include<math.h>
#include<vector>

double F(double x) \\function
{
    double val = pow(3,x) - 5*pow(x,2) + 1;
    return val;
}

double FDer(double x) \\first derivation
{
    double val = pow(3,x) - 10*x;
    return val;
}

double FDDer(double x) \\second derivation
{
    double val = pow(3,x) - 10;
    return val;
}

bool CheckAB(std::pair<double, double>& p, double eps) \\check methods convergence
    conditions
{
    int a, b;
    bool ind = false;
    int fa = F(p.first);
    int fb = F(p.second);
    while(ind == false)
    {
        while(fa*fb > 0)
        {
            if(fa > 0)
            {
                p.first -= eps;
                p.second -= eps;
            }
```

```
39              else
40              {
41                  p.first += eps;
42                  p.second += eps;
43              }
44              fa = F(p.first);
45              fb = F(p.second);
46          }
47          if(FDer(p.first)*FDer(p.second) < 0 && FDDer(p.first)*FDDer(p.second) < 0)
48          {
49              p.first += eps;
50              p.second -= eps;
51              ind = false;
52          }
53          else ind = true;
54      }
55      return true;
56  }
57
58  int main()
59  {
60      double eps;
61      int it_count;
62      int i = 0;
63      std::vector<double> X(2);
64      std::vector<std::pair<double, double>> ab(2);
65      ab[0] = std::make_pair(0.5, 1.5);
66      ab[1] = std::make_pair(3.5, 4.5);
67
68      std::cout << "  : ";
69      std::cin >> eps;
70
71      for(int i = 0; i < 2; i++)
72      {
73          CheckAB(ab[i], eps);
74          X[1] = ab[i].second;
75          it_count = 0;
76          while(F(X[1])*FDDer(X[1]) <= 0)
77              X[1] -= eps;
78          do
79          {
80              X[0] = X[1];
81              X[1] = X[0] - F(X[0])/FDer(X[0]);
82              it_count++;
```

```
83          }
84          while(fabs(X[1]-X[0])>= eps);
85
86          std::cout << i + 1 << "-st(nd) positive solution of a nonlinear equation found by
                Newton's method: " << X[1] << std::endl;
87          std::cout << "The number of iterations: " << it_count << std::endl;
88      }
89      return 0;
90  }
```

5 Выводы

Выполнив первую часть первого задания второй лабораторной работы, я познакомилась с методом Ньютона, который решает задачу уточнения корней нелинейных уравнений на заданном интервале с заданной точностью. Этот метод является итерационным и базируется на последовательном приближении, согласно итерационной формуле. Чтобы метод Ньютона сходился, необходимо, чтобы выполнялся ряд условий. Одно из этих условий — существование двух производных постоянного знака на заданном интервале. Если все условия выполняются, этот метод эффективно и быстро находит значения корней с заданной точностью.