Санкт-Петербургский Государственный Университет

Вычислительный практикум

Приближённое решение нелинейных уравнений

Борис Мельников
b.melnikov17@gmail.com

под руководством Алцыбеева Глеба Олеговича
Преподаватель & Ассистент

3 марта 2023 г.

1 Введение

В повседневной жизни студенту математического факультета часто приходится искать решения уравнений, для которых задача отыскания точных корней является довольно трудозатратной. В этом случае целесообразно применение методов приближенных вычислений, где найденными решениями будут числа, отличающиеся от истинных корней уравнения на фиксированную величину.

Целью данной работы ставилось знакомство с основными итерационными методами вычисления приближенных решений нелинейного уравнения. В ходе работы были сделаны попытки реализовать удобный для использования интерфейс между пользователем и компьютером через встроенное в Windows 11 решение PowerShell.

2 Маленькие задачи и возникшие нюансы

Теперь постараемся резюмировать полученные в ходе решения поставленной здачи результаты.

2.1 Основные положения

1. Как говорилось раннее, задача заключается в нахождении приближенного решения нелинейного уравнения. Однако для упрощения вычислений методы использовались для нахождения корней только нечетной кратности.

2. Применены и реализованы программно 4 метода: метод бисекции, метод Ньютона, модифицированный метод Ньютона, метод секущих.

3. Изначально известны функция $f(x)$, которая и задает уравнение $f(x) = 0$, и ее производная $f'(x)$. Пользователь самостоятельно выбирает отрезок, на котором будут найдены решения уравнения, допустимую погрешность и число, на которое будет разбит выбранный отрезок. Входные параметры вводятся с клавиатуры в окне решения PowerShell.

4. Программа реализована на языке C++ в приложении Microsoft Visual Studio.

2.2 Ход работы

Поскольку у каждого конкретного уравнения может иметься более одного корня или даже бесконечное множество корней ($sin\frac{1}{x} = 0$), ограничим промежуток, на котором эти корни будут найдены. Тогда исходная задача сводится к выделению из исходного отрезка конечного числа подотрезков, на каждом из которых расположен только 1 корень. Сделать это можно так: пользователь самостоятельно выбирает некое число N, на которое будет разбит отрезок с заданными концами $[A, B]$; далее происходит фильтрация разбияния по критерию наличия корня на конкретном "кусочке функция $f(x)$ непрерывна, поэтом если она меняет знак на замкнутом множестве, то на этом множестве она достигает 0-значения (принцип Больцано-Коши). Далее, если пользователь по каким-то причинам неудовлетворен полученными отрезками малой длины, он может продолжить "дробление"уже новых отрезков и т.д. В конечном счете получается множество отрезков, содержащих по одному корню. Далее остается лишь уточнить корни, если за исходное решение принять левые концы всех отрезков (за исключением метода бисекции, где изначальный корень выбирается в зависимости от близости к правому или левому концу). Алгоритмы реализованных методов подробно описаны в [1], поэтому лишь скажем, что во всех случаях остановка вычислений происходит, когда абсолютная величина разности между нынешним значением приближения и предыдущим меньше заданной величины ошибки ε, причем для методов Ньютона (обычного и модифицированного) и метода секущих такой подход гарантирует, что $|x - x^*| < \varepsilon$, где x^* - истинное решение.

Рассматривая метод Ньютона (или его модифицированную версию), стоит сказать, что формально возможна ситуация, когда производная в некоторой точке равна 0, и существует риск деления на 0. Но понятно, что в таком случае деление не произойдет, так как такая точка является искомой, и алгоритм завершит работу.

3 Краткое описание интерфейса

В ходе исполнения программы пользователю прелагается самостоятельно ввести границы A, B отрезка, на котором будут найдены решения, и допустимую ошибку ε. После ввода эти данные изменению не подлежат, при неправильном порядке ввода левого и правого концов программа все же завершит работу, но число корней будет равным 0, что неверно. Далее пользователю предлагается ввести число N, на которое сам отрезок и будет разбит. Реализована защита, при которой значения меньшие или равные 1 не будут обработаны, и пользователю будет задан вопрос, хочет ли он вновь ввести данное значение. После первого разбиения в терминале будут показаны отрезки, содержащие корни, и пользователю предстоит решить, требуется ли дополнительное разбиение. В случае, если первичное значение N введено некорректно и пользователь отказывается его менять, программа прекращает работу с кодом ошибки 404. Если же первичное и последующие значения для разбиения корректны и пользователь удовлетворен полученными отрезками, происходит уточнение корней на данных отрезках с последующим выводом сопуствующей информации о ходе исполнения алгоритмов всех методов для всех корней.

4 Тестируемая задача

В качестве самой функции, относительно которой будет решено уравнение, выбрана $f(x) = 1.2 \cdot x^4 + 2 \cdot x^3 - 13 \cdot x^2 - 14.2 \cdot x - 24.1$ и, соответственно, ее производная $f'(x) = 4.8 \cdot x^3 + 6 \cdot x^2 - 26 \cdot x - 14.2$. За ε принята величина 10^{-6}, N же равно 10000. Концами являются $A = -5$, $B = 5$.

```
Output:
@@ Numerical methods for solving nonlinear equations

@@ Parameters:
    f(x) = 1.2x^4 + 2x^3 - 13x^2 - 14.2x - 24.1
    A = -5
    B = 5
    e = 0.000001

@@ Root separation procedure:
    Hint: Choose N greater than 1
    N = 10000

/!\ Number of intervals: 2
    [-3.968, -3.967];
    [3.287, 3.288];

    OK, let's split the intervals again?
    # ans: no

@@ Root Refinement:
/!\  #1 root:
  Bisector Method:
    Initial Root: -3.967
    Number of iterations: 9
    Final Root: -3.96719
```

```
    |x_m - x_{m - 1}|: 9.76562e-07
    Discrepancy ( |f(x)| ): 3.7478e-05

  Newton's Method:
    Initial Root: -3.968
    Number of iterations: 2
    Final Root: -3.96719
    |x_m - x_{m - 1}|: 2.28093e-08
    Discrepancy ( |f(x)| ): 4.9738e-14

  Modified Newton's Method:
    Initial Root: -3.968
    Number of iterations: 2
    Final Root: -3.96719
    |x_m - x_{m - 1}|: 2.21729e-07
    Discrepancy ( |f(x)| ): 2.76243e-08

  The Secant Method:
    Initial Root: -3.968
    Number of iterations: 2
    Final Root: -3.96719
    |x_m - x_{m - 1}|: 9.96579e-08
    Discrepancy ( |f(x)| ): 1.15911e-05

/!\  #2 root:
  Bisector Method:
    Initial Root: 3.288
    Number of iterations: 9
    Final Root: 3.28791
    |x_m - x_{m - 1}|: 9.76562e-07
    Discrepancy ( |f(x)| ): 0.000110016

  Newton's Method:
    Initial Root: 3.287
    Number of iterations: 2
    Final Root: 3.28791
    |x_m - x_{m - 1}|: 5.45369e-09
    Discrepancy ( |f(x)| ): 7.10543e-15

  Modified Newton's Method:
    Initial Root: 3.287
    Number of iterations: 2
    Final Root: 3.28791
    |x_m - x_{m - 1}|: 1.11328e-07
    Discrepancy ( |f(x)| ): 1.70603e-08

  The Secant Method:
    Initial Root: 3.287
    Number of iterations: 2
    Final Root: 3.28791
    |x_m - x_{m - 1}|: 5.28411e-08
    Discrepancy ( |f(x)| ): 7.17542e-06
```

Список литературы

[1] Лебедева А.В., Пакулина А.Н. Практикум по методам вычислений. Часть 1. СПб., СПбГУ. 2021. 156 - стр.