



Санкт-Петербургский Государственный Университет
Прикладная математика и информатика

Отчёт по вычислительному практикуму 1
Приближённое решение нелинейных уравнений

Выполнил:
Яковлев Денис Михайлович
Группа 21.Б06-мм
st095998@student.spbu.ru

Под руководством Алцыбеева Глеба Олеговича
Преподавателя и ассистента по дисциплине "Вычислительный практикум"

12 Марта 2023 г.

Содержание

1	Преамбула	1
2	Постановка задачи	1
3	Ход работы	2
3.1	Примечание	2
3.2	Пример: Входные данные	2
3.3	Вывод	4
4	Приложение	6
4.1	Метод бисекции	6
4.2	Метод Ньютона	7
4.3	Модифицированный метод Ньютона	8
4.4	Метод секущих	8

1 Преамбула

Обучаясь на программе "Прикладная математика и информатика" каждый студент получает фундаментальные и прикладные знания в области математических дисциплин и информатики, а также формирует компетенции в результате освоения основной образовательной программы.

Дисциплина "Вычислительный практикум" ("Computational Workshop") нацелена на формирование у студента следующих компетенций:

- **ПКП-4:** *Способен критически анализировать разрабатываемые решения, оценивать их эффективность и целесообразность*
- **ПКП-6:** *Способен обрабатывать, анализировать данные и делать выводы, используя соответствующий математический аппарат и современные прикладные программные средства*
- **ПКП-7:** *Способен преподавать математику и информатику в средней школе, специальных учебных заведениях на основе полученного фундаментального образования и научного мировоззрения*

2 Постановка задачи

Пусть дано алгебраическое или трансцендентное уравнение вида

$$f(x) = 0 \tag{1}$$

причем, известно, что все интересующие вычислителя корни находятся на отрезке $[A, B]$, на котором функция $f(x)$ определена и гладкая. Требуется найти все корни уравнения (1) на $[A, B]$ нечетной кратности (здесь $A, B, f(x)$ – параметры задачи). Решение задачи разбить на два этапа:

1. Процедура отделения корней уравнения (1) на отрезке $[A, B]$;
2. Уточнение корней уравнения (1) на отрезках перемены знака вида $[a_i, b_i]$;
 - а. Методом половинного деления (методом бисекции);
 - б. Методом Ньютона (методом касательных);
 - в. Модифицированным методом Ньютона;
 - г. Методом секущих

с заданной точностью $\epsilon > 0$ (ϵ – параметр задачи).

3 Ход работы

3.1 Примечание

- 1) Требования к оформлению задачи: вывести на печать название темы, исходные параметры задачи: A , B , вид функции $f(x)$, ϵ .
- 2) Отделение корней произвести способом табулирования $[A, B]$ с шагом $h > 0$ (где $h = \frac{(B-A)}{N}$, $N \geq 2$ — параметр задачи). При реализации выбирать достаточно малые значения h . Результатом решения задачи отделения корней является последовательный
- 3) При уточнении корней на каждом из отрезков $[a_i, b_i]$ указанными методами, выводить на печать (для каждого метода)
 - название метода (для порядка));
 - начальное(ые) приближение(я) к корню;
 - количество шагов m (в каждом методе своё) для достижения точности ϵ , та-кой что $|x_m - x_{m-1}| < \epsilon$;
 - приближенное решение x_m уравнения (1), найденное каждым из упомянутых методов с точностью ϵ ;
 - $|x_m - x_{m-1}|$ (в методе бисекции выводить длину последнего отрезка);
 - абсолютную величину невязки для пригл. решения $x_m : |f(x_m) - 0|$.

3.2 Пример: Входные данные

В качестве функции выбрана $f(x) = \sqrt{x} - 2\cos(\frac{\pi x}{2})$, а её производная $f'(x) = \frac{1}{2\sqrt{x}} + \pi \sin(\frac{\pi x}{2})$.
Концы отрезка: $A = 0, B = 4.5, \epsilon = 10^{-8}, N = 10^4$

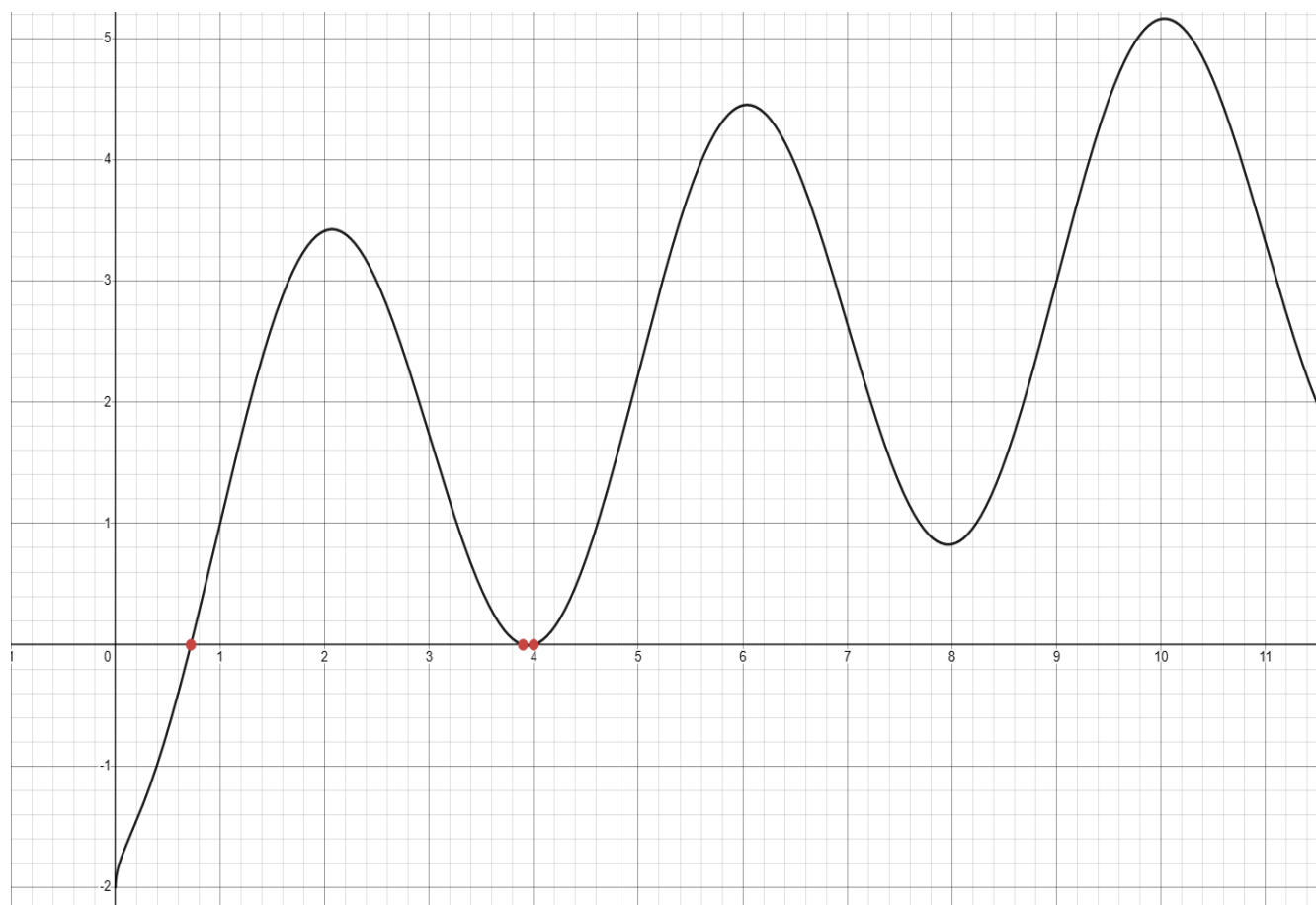


Рис. 1: График функции $f(x) = \sqrt{x} - 2\cos(\frac{\pi x}{2})$. Корни выделены красным цветом

Ссылка на Git-репозиторий: <https://github.com/DeMiYak/NumericalMethods>

3.3 Вывод

```
1      Тема: Приближённые решения нелинейных уравнений
2 f(x) = sqrt(x) - 2*cos(pi*x/2)
3 A = 0
4 B = 4.5
5 eps = 1e-8
6 N = 10000Метод
7
8
9
10 бисекции:Между
11
12 0.72045 и 0.7209:Начальное
13 приближение: 0.720675Число
14 итераций: 15Приближённое
15 решение: 0.720882847595Длина
16 последнего отрезка: 6.8664550712e-09Невязка
17 |f(X)-0|: 1.67105285165e-08Между
18
19 3.89745 и 3.8979:Начальное
20 приближение: 3.897675Число
21 итераций: 15Приближённое
22 решение: 3.89780396576Длина
23 последнего отрезка: 6.86645496017e-09Невязка
24 |f(X)-0|: 2.72432121307e-09Между
25
26 3.9996 и 4.00005:Начальное
27 приближение: 3.999825Число
28 итераций: 15Приближённое
29 решение: 3.9999999847Длина
30 последнего отрезка: 6.86645496017e-09Невязка
31 |f(X)-0|: 3.81626286128e-10Число
32
33 корней: 3Метод
34
35
36 Ньютона:Между
37
38 0.72045 и 0.7209:Начальное
39 приближение: 0.720675Число
40 итераций: 3Приближённое
41 решение: 0.720882842728
42 |x{m}-x{m+1}|: 1.11022302463e-16Невязка
43 |f(X)-0|: 2.22044604925e-16Между
44
45 3.89745 и 3.8979:Начальное
```

46 приближение: 3.897675Число
 47 итераций: 3Приближённое
 48 решение: 3.8978039767
 49 $|x\{m\}-x\{m+1\}|$: 2.52686760405e-13Невязка
 50 $|f(X)-0|$: 2.22044604925e-16Между
 51
 52 3.9996 и 4.00005:Начальное
 53 приближение: 3.999825Число
 54 итераций: 3Приближённое
 55 решение: 4
 56 $|x\{m\}-x\{m+1\}|$: 8.91731133379e-13Невязка
 57 $|f(X)-0|$: 2.22044604925e-16Число
 58
 59 корней: 3Модифицированный
 60
 61
 62 метод Ньютона:Между
 63
 64 0.72045 и 0.7209:Начальное
 65 приближение: 0.720675Число
 66 итераций: 3Приближённое
 67 решение: 0.720882842728
 68 $|x\{m\} - x\{m-1\}|$: 1.08446585045e-12Невязка
 69 $|f(X)-0|$: 3.33066907388e-16Между
 70
 71 3.89745 и 3.8979:Начальное
 72 приближение: 3.897675Число
 73 итераций: 3Приближённое
 74 решение: 3.8978039767
 75 $|x\{m\} - x\{m-1\}|$: 4.02158306656e-10Невязка
 76 $|f(X)-0|$: 2.5113244817e-13Между
 77
 78 3.9996 и 4.00005:Начальное
 79 приближение: 3.999825Число
 80 итераций: 3Приближённое
 81 решение: 4
 82 $|x\{m\} - x\{m-1\}|$: 1.04250741373e-09Невязка
 83 $|f(X)-0|$: 8.94395668638e-13Число
 84
 85 корней: 3Метод
 86
 87
 88 текущих:Между
 89
 90 0.72045 и 0.7209:Начальное
 91 приближение: 0.720882840903Число
 92 итераций: 1Приближённое

```

93 решение: 0.720882842728
94 |x{m}-x{m-1}|: 1.8250588818e-09Невязка
95 |f(X)-0|: 2.63122856836e-14Между
96
97 3.89745 и 3.8979:Начальное
98 приближение: 3.89780430627Число
99 итераций: 2Приближённое
100 решение: 3.8978039767
101 |x{m}-x{m-1}|: 3.07913250452e-10Невязка
102 |f(X)-0|: 2.22044604925e-16Между
103
104 3.9996 и 4.00005:Начальное
105 приближение: 3.9999980318Число
106 итераций: 2Приближённое
107 решение: 4
108 |x{m}-x{m-1}|: 9.6463725896e-11Невязка
109 |f(X)-0|: 0Число
110
111 корней: 3

```

4 Приложение

4.1 Метод бисекции

```

1 cout << '\n' << '\n' << "Метод бисекции: " << endl;
2 while(x < B)
3 {
4     double a = x, b = xNext;
5     double c, delta, fappr=(a+b)/2;
6     size_t m = 0;
7     if(f(a)*f(b)<=0)
8     {
9         counter++;
10        while(b - a >= 2*eps){
11            m++;
12            c = (a+b)/2;
13            if(f(b)*f(c)<=0)
14            {
15                a = c;
16            } else b = c;
17        }
18        delta = (b-a)/2;
19
20        cout << '\n' << "Между " << x << " и " << xNext << ":\n"<< "Начальное
приближение: " << fappr << endl;

```



```

21         cout << "Число итераций: " << m << "\Приближённое решение: " << c <<
"\Длина последнего отрезка: " << delta;
22         cout << "\Невязка |f(X)-0|: " << fabs(f(c)) << endl;
23
24     }
25
26     x = xNext;
27     xNext += h;
28 }
29 cout << '\n' << "Число корней: " << counter << endl;

```

4.2 Метод Ньютона

```

1     cout << '\n' << '\n' << "Метод Ньютона: " << endl;
2     while(x < B)
3     {
4         double a = x, b = xNext;
5         double c = (a+b)/2, delta, fappr=(a+b)/2;
6         size_t m = 0;
7         if(f(a)*f(b) <= 0){
8             counter++;
9             while(fabs(c - a) >= eps){
10                 m++;
11                 a = c;
12                 c = fi(c);
13                 if(c > xNext || c < x){
14                     cout << "Между " << x << " и " << xNext << ": Ошибка: корень не
может быть вычислен попробуйте( взять N побольше)" << endl;
15                     m = 0;
16                     break;
17                 }
18             }
19         }
20         if(m){
21             delta = fabs(c-a);
22
23             cout << '\n' << "Между " << x << " и " << xNext << ":\n" << "Начальное
приближение: " << fappr << endl;
24             cout << "Число итераций: " << m << "\Приближённое решение: " << c <<
"\n|x{m}-x{m+1}|: " << delta;
25             cout << "\Невязка |f(X)-0|: " << fabs(f(c)) << endl;
26         }
27         x = xNext;
28         xNext += h;
29     }
30     cout << '\n' << "Число корней: " << counter << endl;

```

4.3 Модифицированный метод Ньютона

```
1  cout << '\n' << '\n' << "Модифицированный метод Ньютона: " << endl;
2  while(x < B)
3  {
4      double a = x, b = xNext;
5      double c = (a+b)/2, delta, fappr=(a+b)/2;
6      double cf = df(c);
7      size_t m = 0;
8      if(f(a)*f(b) <= 0){
9          counter++;
10         while(fabs(c - a) >= eps){
11             m++;
12             a = c;
13             c = c - f(c)/cf;
14             if(c > xNext || c < x){
15                 cout << "Между " << x << " и " << xNext << ": Ошибка: корень не
может быть вычислен попробуйте( взять N побольше)" << endl;
16                 m = 0;
17                 break;
18             }
19         }
20     }
21     if(m){
22         delta = fabs(c-a);
23
24         cout << '\n' << "Между " << x << " и " << xNext << ":\n"<< "Начальное
приближение: " << fappr << endl;
25         cout << "Число итераций: " << m << "\Приближённое решение: " << c <<
"\n|x{m} - x{m-1}|: " << delta;
26         cout << "\Невязкан |f(X)-0|: " << fabs(f(c)) << endl;
27     }
28     x = xNext;
29     xNext += h;
30 }
31 cout << '\n' << "Число корней: " << counter << endl;
```

4.4 Метод секущих

```
1  cout << '\n' << '\n' << "Метод секущих: " << endl;
2  while(x < B){
3      double a = x, b = xNext;
```

```

4      double c = secf(b, a), delta, fappr=secf(b, a);
5      size_t m = 0;
6      if(f(a)*f(b)<=0){
7          counter++;
8          while(fabs(c - b) >= eps){
9              m++;
10             a = b;
11             b = c;
12             c = secf(c, a);
13             if(c < x || c > xNext){
14                 cout << "Между " << x << " и " << xNext << ": Ошибка: корень не
может быть вычислен попробуйте( взять N побольше)" << endl;
15                 m = 0;
16                 break;
17             }
18         }
19     }
20     if(m){
21         delta = fabs(c - b);
22         cout << '\n' << "Между " << x << " и " << xNext << ":\n" << "Начальное
приближение: " << fappr << endl;
23         cout << "Число итераций: " << m << "\Приближённое решение: " << c <<
"\n|x{m}-x{m-1}|: " << delta;
24         cout << "\Невязка |f(X)-0|: " << fabs(f(c)) << endl;
25     }
26     x = xNext;
27     xNext += h;
28 }
29 cout << '\n' << "Число корней: " << counter << endl;

```
