Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчет по лабораторной работе №1 дисциплины
«Вычислительная математика»

Выполнил студент группы ИВТ-22___________/Крючков И. С/
Проверил___________________________________/Исупов К. С./

Киров 2021

Задание:
1. Построить график функции f(x) и отделить один из корней уравнения: f(x).
2. Сузить интервал изоляции корня, если необходимо, проверив условие M <= 2m.
3. Уточнить корень с погрешностью e <= 0.00001 двумя численными методами: комбинированным методом и методом итераций.
4. Проверить полученное значение корня, используя систему Mathcad.

   Вариант 10
   Уравнение: x^3 – sin(x) = 0
   Интервал: [0.1, 1.0]

Теоретические сведения:
Теоретические сведения об уточнении корней комбинированным методом. Основная суть метода — сузить интервал изоляции с двух сторон. В зависимости от неподвижного конца интервала изоляции используются две пары формул, и на каждом шаге вычисляются приближенное значение корня по недостатку x_{k+1} и по избытку $\tilde{x}_{k+1}$.

Если неподвижна точка a:
- значение по недостатку вычисляется методом касательных:
  $$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, x_0 = a$$
- Значение по избытку вычисляется методом хорд:
  $$\tilde{x}_{k+1} = \tilde{x}_k - f(\tilde{x}_k)\, \frac{\tilde{x}_k - x_k}{f(\tilde{x}) - f(x_k)}, \tilde{x}_0 = b$$

Если неподвижна точка b:
- значение по недостатку вычисляется методом хорд:
  $$x_{k+1} = x_k - \frac{\tilde{x}_k - x_k}{f(\tilde{x}) - f(x_k)}, x_0 = a$$
- Значение по избытку вычисляется методом касательных:
  $$\tilde{x}_{k+1} = \tilde{x}_k - \frac{f(\tilde{x}_k)}{f'(\tilde{x}_k)}, \tilde{x}_0 = b$$

На каждом новом шаге расчета используют суженный с двух сторон интервал $[x_k, \tilde{x}_k]$.

Теоретические сведения об уточнении корней методом простых итераций.
Вещественное число ξ называется неподвижной точкой функции φ, если φ(ξ) = ξ.
Алгоритм:
1. Преобразуем уравнение к каноническому виду: f(x) = 0 → x = φ(x).
2. Выберем начальное приближение x_0 — любую точку из интервала [a, b].
3. Итерационный процесс осуществляем в соответствии с рекуррентным соотношением: $x_n+1 = φ(x_n)$, n = 1, 2, 3, . . .

Итерационный процесс сходится при условии, что первая производная итерационной функции φ(x) по модулю меньше единицы: |φ 0 (x)| < 1.

Если условие |φ'(x)| < 1 выполняется, то на отрезке [a, b], на котором локализован корень, в качестве начального приближения можно взять любую точку $x_0 \in [a, b]$. Скорость сходимости зависит от абсолютной величины производной: чем меньше |φ' (x)|, тем быстрее сходится процесс, что непосредственно следует из формулы (18): εn+1 = εnφ 0 (ξ).

Тип сходимости и критерий останова:

- Двусторонняя сходимость:
  Если φ'(x) < 0, то сходимость двусторонняя.
  Критерий окончания итерационного процесса $|x_n+1 - x_n| \le \varepsilon$ является объективным.
- Монотонная сходимость:
  Если φ'(x) > 0, то сходимость односторонняя.
  Критерий окончания итерационного процесса:
  $\frac{1}{1-q}|x_{n+1} - x_n| \le$ ε, где q = max| φ'(x)| на интервале [a, b]

Наибольшая скорость сходимости итерационного процесса:

$$Xn+1 = \varphi(x_n),\ n = 1, 2, \ldots$$

достигается при

$$\varphi'(x) = 1 + k \cdot f'(x) = 0$$

Этого можно добиться, если выбрать параметр k в уравнении φ(x) = x + k · f(x) зависящим от x в виде

$$k = -\frac{1}{f'(x)}$$

При этом итерационная формула переходит в формулу метода касательных Ньютона:

$$x_{n+1} = \varphi(x_n) = x - \frac{f(x)}{f'(x)}, n = 1,2, \ldots$$

Практическая часть

Исходная функция: $f(x) = x^3 - \sin(x)$

Первая производная: $f'(x) = 3x^2 - \cos{(x)}$

Вторая производная: $f''(x) = 6x + \sin{(x)}$

Канонический вид: $\varphi(x) = x - \frac{x^3 - \sin(x)}{k}$

$\varphi'(x) = 1 - \frac{1}{k} * 3x^2 - \cos(x)$

$|k| \ge \frac{N}{2}$

$N = \max|f'(x)|, \qquad [a, b]$

$N = 2.46 => |k| \ge 1.23$

$k = 7$

Ответ комбинированного метода: 0.928626 ± 0,00001

Ответ метода итераций: 0.928626 ± 0,00001

Проверка в системе wolframalpha:

Input interpretation

solve	$x^3 - \sin(x) = 0$

Solution over the reals

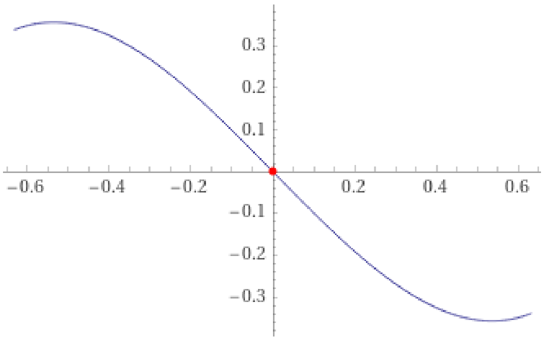
Exact forms | More digits

$x \approx -0.928626$

$x = 0$

$x \approx 0.928626$

Root plot



Download Page

POWERED BY THE **WOLFRAM LANGUAGE**

Экранные формы:

```
f(x) = x**3 - np.sin(x)
[0.1, 1.0]
f(a) = -0.09883341664682815
f(b) = 0.1585290151921035
f`(a) = -0.9650041652780258
f`(b) = 2.4596976941318602
f``(a) = 0.6998334166468283
f``(b) = 6.841470984807897
По недостатку - метод хорд
По избытку - метод касательных
M <= 2m: Неверно
Уточнение интервала изоляции: [0.920,0.930]
+---+-----------+-----------+----------+------------+------------+
| n |    xa     |   f(xa)   |    xb    |   f(xb)    |  xb - xa   |
+---+-----------+-----------+----------+------------+------------+
| 0 | 0.930000  |  0.002737 | 0.920000 | -0.016914  | -0.010000  |
| 1 | 0.928607  | -0.000038 | 0.928748 |  0.000242  |  0.000141  |
| 2 | 0.928626  | -0.000000 | 0.928626 |  0.000000  |  0.000000  |
+---+-----------+-----------+----------+------------+------------+
Решение: 0.928626
```

Рисунок 1 Комбинированный метод

```
f(x) = x^3 - sin(x)
phi(x) = x - (x^3 - sin(x))/7
phi(a) = 0.11411905952097545
phi(b) = 0.9773529978296995
phi`(a) = 1.1378577378968608
phi`(b) = 0.64861461512402
Уточнение интервала изоляции: [0.920,0.930]
Тип сходимости: Односторонняя
Критерий останова: 0.0000138099
phi`(an) = 0.723802879520494
phi`(bn) = 0.7147334260410427
Тип сходимости: Односторонняя
Критерий останова: 0.0000138099
+----+--------------+--------------+
| n  |      xn      |   phi(xn)    |
+----+--------------+--------------+
| 1  | 0.9224162314 | 0.9241624708 |
| 2  | 0.9241624708 | 0.9254212124 |
| 3  | 0.9254212124 | 0.9263268375 |
| 4  | 0.9263268375 | 0.9269775172 |
| 5  | 0.9269775172 | 0.9274445622 |
| 6  | 0.9274445622 | 0.9277795607 |
| 7  | 0.9277795607 | 0.9280197237 |
| 8  | 0.9280197237 | 0.9281918355 |
| 9  | 0.9281918355 | 0.9283151466 |
| 10 | 0.9283151466 | 0.9284034773 |
| 11 | 0.9284034773 | 0.9284667422 |
| 12 | 0.9284667422 | 0.9285120499 |
| 13 | 0.9285120499 | 0.9285444952 |
| 14 | 0.9285444952 | 0.9285677284 |
| 15 | 0.9285677284 | 0.9285843646 |
| 16 | 0.9285843646 | 0.9285962765 |
| 17 | 0.9285962765 | 0.9286048057 |
+----+--------------+--------------+
Решение: 0.928605
```

Рисунок 2 Метод простых итераций

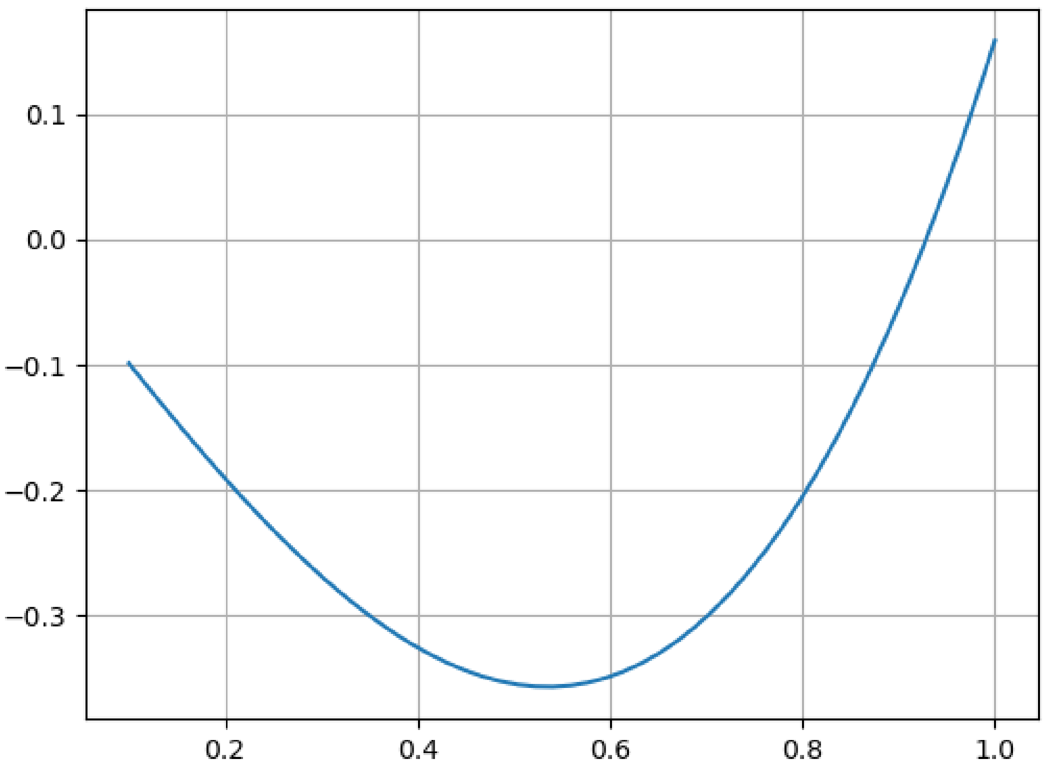


Рисунок 3 График функции

Листинг кода:

```
import numpy as np
import matplotlib.pyplot as plt
from prettytable import PrettyTable


fig, ax = plt.subplots()
ax.grid()

# интервал
a, b = 0.1, 1.0
# погрешность
eps = 0.00005

# функция
pf = "x^3 - sin(x)"
f = lambda x:  x**3 - np.sin(x)

# первая производная
df = lambda x: 3 * x**2 - np.cos(x)
#вторая производная
ddf = lambda x: 6 * x + np.sin(x)

k = 7
#phi(x)
pphi = "x - (x^3 - sin(x))/"+str(k)
phi = lambda x: x - ((x**3 - np.sin(x))/k)
#первая производная phi(x)
dphi = lambda x: 1-(1/k)*df(x)
pdphi = "1 - 1/"+str(k)+" * 3x^2 - cos(x)"

res = a
x = np.linspace(a, b)

def combined_method(left, right):
    def hords(xa, xb):
        return xb - f(xb) * (xb - xa) / (f(xb) - f(xa))

    def casat(xa):
        return xa - f(xa) / df(xa)

    fixed = f(left) * ddf(left) > 0
    if fixed:
        xa, xb = left, right
    else:
        xa, xb = right, left

    i = 0
    yield i, xa, f(xa), xb, f(xb), xb - xa

    while abs(xb - xa) > eps and i < 50:

        i += 1
```

```
        #Если неподвижна А
        if fixed:
            xb = hords(xa, xb)
            xa = casat(xa)
        else:
            xa = hords(xb, xa)
            xb = casat(xb)


        yield i, xa, f(xa), xb, f(xb), xb - xa


def iter_method(left, right, er):

    i = 0
    xn = left
    xn1 = right
    delta = xn-xn1
    while abs(delta) > er and i < 50:
        i += 1
        xn1 = phi(xn)
        delta = xn-xn1
        xn = xn1

        yield i, xn, phi(xn)



def checkform(left):
    if f(left) * ddf(left) > 0:
        r1 = 'По недостатку - метод касательных'
        r2 = 'По избытку - метод хорд'
    else:
        r1 = 'По недостатку - метод хорд'
        r2 = 'По избытку - метод касательных'
    return r1, r2

def isolate(left, right, acc):
    while f(left + acc) * f(right - acc) < 0:
        left += acc
        right -= acc

    while f(left) * f(left + acc) > 0:
        left += acc

    while f(right) * f(right - acc) > 0:
        right -= acc

    return left, right


def combined():
    print(f"f(x) = {pf}")
    print(f"[{a}, {b}]")
```

```
    print(f"f(a) = {f(a)}")
    print(f"f(b) = {f(b)}")
    print(f"f`(a) = {df(a)}")
    print(f"f`(b) = {df(b)}")
    print(f"f``(a) = {ddf(a)}")
    print(f"f``(b) = {ddf(b)}")
    r1, r2 = checkform(a)
    print(r1)
    print(r2)

    mm = abs(ddf(b)) <= abs(2*df(a))
    print(f"M <= 2m: {'верно' if mm else 'Неверно'}")

    an, bn = isolate(a,b, 0.01)
    print("Уточнение интервала изоляции: [{:.3f},{:.3f}]".format(an,
bn))

    tablex = PrettyTable()
    for t in combined_method(an,bn):
        #
print('{}\t{:.15f}\t{:.15f}\t{:.15f}\t{:.15f}\t{:.15f}'.format(*t))
        tablex.field_names = ["n", "xa", "f(xa)", "xb","f(xb)", "xb -
xa"]
        tablex.add_row(t)
        tablex.float_format = '.6'
        res = t[1]

    print(tablex)
    print(f'Решение: {res:.6f}')


def iteration():
    print(f"f(x) = {pf}")
    print(f"f(a) = {f(a)}")
    print(f"f(b) = {f(b)}")
    print(f"f`(a) = {df(a)}")
    print(f"f`(b) = {df(b)}")
    print(f"phi(x) = {pphi}")
    print(f"phi(a) = {phi(a)}")
    print(f"phi(b) = {phi(b)}")
    print(f"phi`(a) = {dphi(a)}")
    print(f"phi`(b) = {dphi(b)}")

    an, bn = isolate(a,b, 0.01)
    print("Уточнение интервала изоляции: [{:.3f},{:.3f}]".format(an,
bn))

    if dphi(an) < 0 and dphi(bn) < 0:
        ts = "Двусторонняя"
        e = eps
    else:
        ts = "Односторонняя"
        e = abs(1 - max(abs(dphi(an)), abs(dphi(bn)))) * eps

    print(f"Тип сходимости: {ts}")
```

```
    print(f"Критерий останова: {e:.10f}")

    print(f"phi`(an) = {dphi(an)}")
    print(f"phi`(bn) = {dphi(bn)}")

    tablex = PrettyTable()
    for t in iter_method(an,bn,e):
        # print('{}\t{:.15f}\t{:.15f}'.format(*t))
        tablex.field_names = ["n", "xn", "phi(xn)"]
        tablex.add_row(t)
        tablex.float_format = '.10'
        res = t[2]

    print(tablex)
    print(f'Решение: {res:.6f}')


# combined()
iteration()

plt.plot(x, f(x))
plt.show()
```

Вывод: в ходе выполнения лабораторной работы был закреплен лекционный материал по теме «Численные методы решения нелинейных уравнений». Закреплены на практике численные методы решение нелинейных уравнений: комбинационный метод и итерационный метод, а так же изучены особенности их применения.