ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

Московский Государственный Технический Университет

имени Н.Э. Баумана



Факультет «Фундаментальные науки»

Кафедра «Техническая физика»

**Лабораторная работа №1**

по курсу «Вычислительная физика»:

Выполнили: Коберник Т. Н., Плетенёв Б. А.

Группа: ФН4-71Б

Преподаватель: Ивлиев П.А.

Москва, 2022 г.

СОДЕРЖАНИЕ

[1. Теоретическая часть](#_Toc462452329) 3

[1) Алгоритм метода Ньютона 3](#_Toc462452330)

[2) Алгоритм модифицированного метода Ньютона](#_Toc462452330) 4

[3) Алгоритм метода касательных 6](#_Toc462452330)

[4) Алгоритм упрощенного метода Ньютона 7](#_Toc462452331)

[2. Постановка задачи 9](#_Toc462452332)

[3. Программа 9](#_Toc462452333)

[4. Результаты вычислений](#_Toc462452334) 11

[5. Выводы](#_Toc462452335) 12

**1. Теоретическая часть**

**1) Алгоритм метода Ньютона**

Метод Ньютона - итерационный [численный метод](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D1%8B) нахождения корня заданной [функции](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) посредством последовательных приближений.

Рассмотрим уравнение где .

Пускай существует некоторое решение этого уравнения x\*, тогда будем искать последовательные приближения к этому корню так что

Будем считать, что

Тогда функцию можно разложить в многомерный ряд Тейлора:

Где

* матрица Якоби первых производных в точке .

Пренебрегая остаточным членом, решим полученную систему линейных алгебраических уравнений относительно Получим

Это векторное уравнение представляет собой и*терационную процедуру Ньютона в многомерном случае*. Для ее запуска необходимо задать начальную точку .

Стандартный алгоритм метода Ньютона:

1. Задается некоторое начальное приближение
2. Вычисляется вектор-функция f() и матрица ее первых частных производных по x .
3. Решается система (1) и определяется
4. Получается новое приближение .
5. Процесс циклически повторяется до тех пор, пока не будет достигнута требуемая точность.

**2) Алгоритм модифицированного метода Ньютона**

Метод Ньютона является эффективным при наличие достаточно хорошего начального приближения. Если же выбрано грубо, то результат вычислений может расходиться. Чтобы понизить зависимость результата от начального приближения, был создан модифицированный метод Ньютона.

Рассмотрим случай с n = 2. Тогда имеем систему из двух уравнений:

Выберем начальное приближение ().

При разложении функций двух переменных в ряд Тейлора получим:

Последовательно решая эту систему, находим

Где А произвольный .

Распишем разложение функции в произвольной точке в компактном виде:

Модифицируем уравнение следующим образом:

Далее ищется значение , при котором аргумент функции принимает минимальное значение, оно и будет являться искомым корнем:

(4)

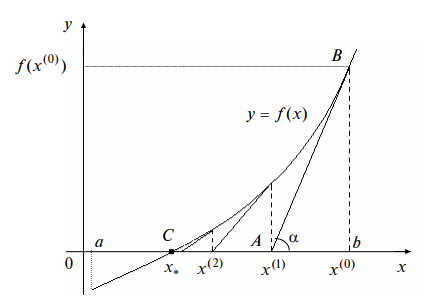
Алгоритм модифицированного метода Ньютона:

1. Задается некоторое начальное приближение
2. Вычисляется вектор-функция f() и матрица ее первых частных производных по x .
3. Решается система (2) и определяется
4. Получаем произвольное приближение и строим функцию от скалярного аргумента s (3).
5. Решаем задачу минимизации (4).
6. Получаем искомый корень функции

**3) Алгоритм метода касательных**

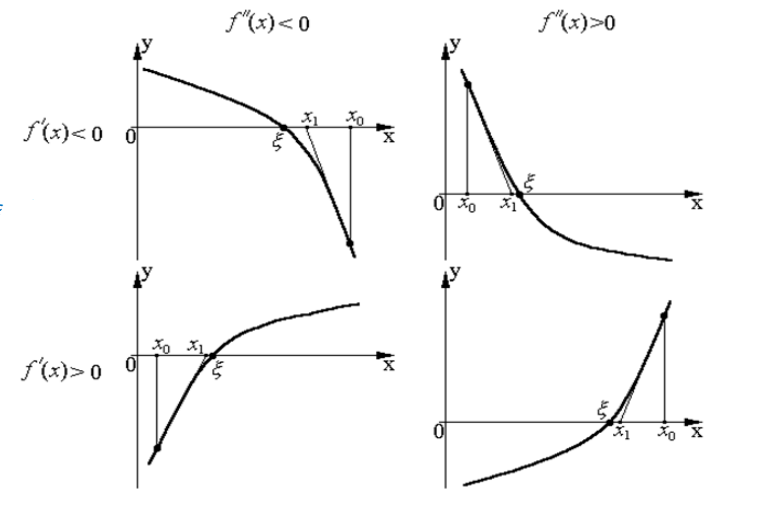
Метод касательных является частным случаем стандартного метода Ньютона при условии, что n = 1. В этом случае при разложении функции в ряд Тейлора, как было показано выше, мы получаем уравнение касательной к графику функции f(x) в некоторой точке , а затем приравниваем это уравнение к 0. Тем самым, мы фактически находим точку пересечения касательной с осью абсцисс, что и будет являться последующим приближением к нулю функции - .

Полученная формула (5) является формулой инерционного процесса.



*Рис 1. Геометрическая интерпретация метода касательных*

Как и в стандартном методе Ньютона, в методе касательных задается некоторое начальное приближение . При чем при выборе этой точки определяется некоторый отрезок [a, b] такой, что f(c) = 0, Это выполняется, если Далее в качестве выбирается тот конец отрезка , где выполняется условие (см. рис. 2).



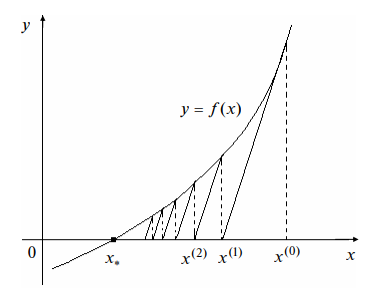
*Рис. 2. Иллюстрация условия выбора начального приближения при методе касательных.*

Таким образом, можем сформулировать алгоритм нахождения нулей функции с помощью метода касательных:

1. Задается начальное приближение (в соответствии с указанными выше условиями).
2. Вычисляется функция f() и ее первая производная .
3. Решается уравнение (5) и находится следующее приближение
4. Процесс циклически повторяется до тех пор, пока не будет достигнута требуемая точность
5. **Алгоритм упрощенного метода Ньютона**

В основе упрощенного метода Ньютона лежит уравнение:

Отличие от метода касательных заключается в том, что производная функции подсчитывается только в точке начального приближения , а на последующих итерациях не уточняется. Процесс последовательных приближений отражен на рис. 3. Первая итерация совпадает с первой итерацией метода касательных. На последующих итерациях соответствующие отрезки параллельны касательной, проведенной в начальной точке.



*Рис. 3. Иллюстрация упрощенного метода Ньютона*

Алгоритм нахождения нулей функции с помощью метода касательных:

1. Задается начальное приближение (аналогично методу касательных).
2. Вычисляется функция f() и ее первая производная .
3. Решается уравнение (6) и находится следующее приближение
4. Процесс циклически повторяется до тех пор, пока не будет достигнута требуемая точность (при этом производная функции рассчитывается единожды в точке ).

**2. Постановка задачи**

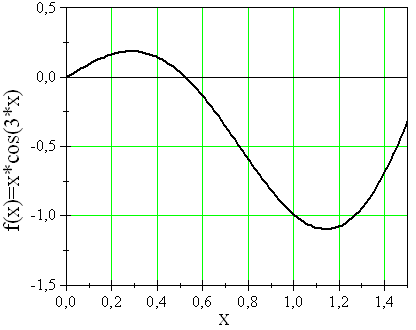
Найти решение уравнения **f(x)=0,** используя стандартный (метод касательных) и упрощенный методы Ньютона. Получить и сравнить результаты для нескольких вариантов итераций (*k* = 3; 5; 10; 20).

Для анализа результатов построить графики производных.

Вид функции:

Корень ищется на отрезке [0.3, 0.8]

График функции:



*Рис.4. График функции*

**3. Программа**

1. **import** math
2. **import** numpy as np
3. **import** matplotlib.pyplot as plt
5. **def** method\_of\_edges(x0, k):
6. xn **=** x0 **-** f(x0)**/**f1(x0)
7. **for** i **in** range(k):
8. x0 **=** xn
9. xn **=** x0 **-** f(x0)**/**f1(x0)
10. **return** x0
11. **def** easy\_method\_of\_edges(x0, k):
12. x0\_const **=** x0
13. xn **=** x0 **-** f(x0)**/**f1(x0\_const)
14. **for** i **in** range(k):
15. x0 **=** xn
16. xn **=** x0 **-** f(x0)**/**f1(x0\_const)
17. **return** x0
19. a **=** 0.3
20. b **=** 0.8
21. eps **=** 0.0001
22. print('Введите количество итераций')
23. k **=** int(input())
25. f **=** **lambda** x: x **\*** math.cos(3 **\*** x) # y = f(x)
26. f1 **=** **lambda** x: (f(x **+** eps) **-** f(x **-** eps)) **/** (2 **\*** eps) # y' = f'(x)
27. f2 **=** **lambda** x: (f1(x **+** eps) **-** f1(x **-** eps)) **/** (2 **\*** eps) # y'' = f''(x)
29. **if** f(a) **/** f2(a) > 0:
30. x0 **=** a
31. **else**:
32. x0 **=** b
33. y**=lambda** x: x **\*** np.cos(3 **\*** x)
34. y1**=lambda** x: np.cos(3 **\*** x) **-** 3 **\*** x **\*** np.sin(3 **\*** x)
35. y2**=lambda** x: **-**6 **\*** np.sin(3 **\*** x) **-** 9 **\*** x **\*** np.cos(3 **\*** x)
36. fig **=** plt.subplots()
37. x **=** np.linspace(0.2, 0.9,10000)
38. plt.plot(x, y(x), label**=**r'$f(x)=x\*cos(3x)$')
39. plt.plot(x,y1(x), label**=**r'$df/dx$')
40. plt.plot(x,y2(x), label**=**r'$d^2f/dx^2$')
41. plt.xlabel(r'$x$', fontsize**=**14)
42. plt.ylabel(r'$f(x)$', fontsize**=**14)
43. plt.grid(True)
44. plt.legend(loc**=**'best', fontsize**=**12)
45. plt.show()
47. x0 **=** easy\_method\_of\_edges(x0, k)
48. print('По упрощенному методу Ньютона:\nx0 =', x0)
49. print('f(x0) =', f(x0))
50. x0 **=** method\_of\_edges(x0, k)
51. print('По методу касательных:\nx0 =', x0)
52. print('f(x0) =', f(x0))

**4. Результаты вычислений**

В результате работы приведённой выше программы были получены следующие результаты.

При количестве итераций k=3

По методу касательных:

= 0.523601280928209,

По упрощенному методу Ньютона:

= 0.5261638493813741,

При количестве итераций k=5

По методу касательных:

= 0.5235987755982989,

По упрощенному методу Ньютона:

= 0.5238811974884282,

При количестве итераций k=10

По методу касательных:

= 0.5235987755982989,

По упрощенному методу Ньютона:

= 0.5235999473744786,

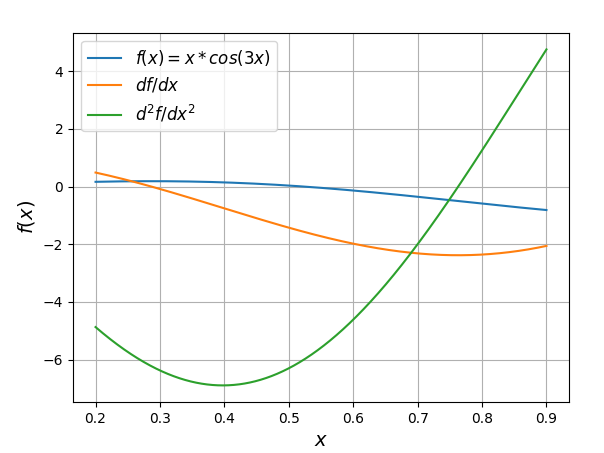
При количестве итераций k=20

По методу касательных:

= 0.5235987755982989,

По упрощенному методу Ньютона:

= 0.5235987756185352,



*Рис. 5. Графики заданной функции f(x) и ее производных на отрезке [0.3, 0.8]*

**5. Выводы**

По результатам вычислений видно, что метод касательных даёт более точные значения, нежели упрощённый метод Ньютона. При малом количестве итераций разница в точности достигает 14 порядков. При последовательном увеличении итераций эта разница уменьшается вплоть до 6 порядков (при k=20). Не трудно заметить, что это уменьшение связано с увеличением точности значения, получаемого исключительно упрощённым методом Ньютона. Метод касательных, в свою очередь, практически не чувствителен к увеличению количества итераций. Результаты, полученные с помощью этого метода при k=5, 10 и 20, различались на число .

Таким образом, можно сделать вывод, что при необходимости нахождения достаточно точного значения корня функции (до 17 порядка), лучше использовать метод касательных. Причем вычисления проводить следует при малом количестве итераций. Это не приведёт к потере точности, но облегчит нагрузку на вычислительную машину. Если же требуется не столь высокая точность вычислений, то можно использовать упрощённый метод Ньютона. Однако необходимо контролировать количество итераций. Поскольку при малом k погрешность вычислений достаточно высока.