



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ДОН-
СКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)**

Факультет «Информатика и вычислительная техника»
Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Зав. кафедрой «ПОВТиАС»

(подпись) В.В. Долгов
(И.О.Ф.)
«__» _____ 2021 г.

ОТЧЕТ

по технологической (проектно-технологической) практике

на базе ФГБОУ ВО «Донской государственный технический университет»

кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Обучающийся _____ А.И. Костюченко
(подпись, дата)
Обозначение отчета УП.260000.000 Группа ВМО22

Направление 02.03.03 Математическое обеспечение и администрирование информационных систем

Профиль Математическое обеспечение и администрирование информационных систем

Руководитель практики:

от кафедры доцент _____ Т.А. Медведева
подпись, дата

Оценка _____
дата _____ подпись _____

Ростов-на-Дону
2021



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВА-
НИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)**

Факультет _____ ИиВТ _____
(наименование факультета)
Кафедра _____ ПОВТиАС _____
(наименование кафедры)

ЗАДАНИЕ

по технологической (проектно-технологической) практике

на базе ФГБОУ ВО «Донской государственный технический университет» _____

кафедра «Программное обеспечение вычислительной техники и автоматизированных систем» _____

в период с «_8_» _____ февраля _____ 2021 г. по «_5_» _____ июня _____ 2021 г.

Обучающийся _____ Костюченко Артём Иванович

Обозначение отчета _____ УП.260000.000 _____ Группа ВМО22

Срок представления отчета на кафедру «_____» _____ 2021 г.

Содержание индивидуального задания:

Метод Фибоначчи для минимизации функции одной переменной

Руководитель практики от кафедры доцент _____ Т.А. Медведева
подпись, дата

Задание принял к исполнению _____ А.И. Костюченко
подпись, дата



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)**

Факультет «Информатика и вычислительная техника»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Зав. кафедрой «ПОВТ и АС»

В.В. Долгов

(подпись)

«___» _____ 2021 г.

Рабочий график (план) проведения практики

№	Мероприятие	Срок выполнения
1	Прохождение вводного и первичного инструктажа по охране труда на рабочем месте и инструктажа по пожарной безопасности на объекте	с 08.02.2021 по 11.02.2021
2	Получение индивидуального задания и постановка задачи	с 12.02.2021 по 17.03.2021
3	Реализация алгоритмов численных методов решения нелинейных уравнений	с 18.02.2021 по 17.03.2021
4	Реализация алгоритмов численных методов решения нелинейных уравнений	с 18.03.2021 по 07.04.2021
5	Выбор среды разработки программы и ее реализация	с 07.04.2021 по 28.04.2021
6	Тестирование программного средства и составление отчета о проделанной работе	с 29.04.2021 по 25.05.2021
7	Защита итогового отчета по практике	с 26.05.2021 по 05.06.2021

Руководитель практики от кафедры доцент

Т.А. Медведева

Ростов-на-Дону

2021

ДНЕВНИК ПРОХОЖДЕНИЯ ПРАКТИКИ

Дата	Место работы	Выполняемые работы	Оценка руководителя
с 08.02.2021 по 10.02.2021	ДГТУ	Знакомство с предприятием, прохождение вводного инструктажа	
с 10.02.2021 по 11.02. 2021	ДГТУ	Ознакомление с территорией предприятия, прохождение первичного инструктажа по ТБ, ПБ	
с 12.02.2021 по 18.02.2021	ДГТУ	Получение индивидуального задания на практику	
с 18.02.2021 по 17.03.2021	ДГТУ	Изучение теории методов оптимизации	
с 18.03.2021 по 07.04.2021	ДГТУ	Алгоритмическое конструирование	
с 07.04.2021 по 28.04.2021	ДГТУ	Программное конструирование и отладка программы	
с 29.04.2021 по 25.05.2021	ДГТУ	Тестирование программного средства и проверка полученных результатов, написание отчета о проделанной работе	
с 26.05. 2021 по 05.06.2021	ДГТУ	Защита итогового отчета по практике	

Руководитель практики от кафедры доцент _____

подпись, дата

Т.А. Медведева

ОТЗЫВ - ХАРАКТЕРИСТИКА

Обучающийся Костюченко Артём Иванович

2 курс группы ВМО22

кафедры «Программное обеспечение вычислительной техники и автоматизированных систем»

Вид практики: технологическая (проектно-технологическая)

Наименование места практики: кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Обучающийся выполнил задания программы практики, разработал приложение с визуальным интерфейсом для минимизации функции одной переменной методом Фибоначчи, а также демонстрация графика введённой функции.

Дополнительно ознакомился/изучил платформу QtDesigner для создания интерфейса, способы парсинга строчного типа (string) для возможности работать с любой введённой пользователем функцией, а также более детально ознакомился с языком программирования Python, фреймворком PyQt5 и средой разработки Visual Studio Code.

Заслуживает оценки _____

Руководитель практики от кафедры
доцент Медведева Т.А.

« ____ » _____ 2021 г.

Содержание

Введение	7
1. Теоретический разбор	8
1.1 Экстремум функции	8
1.2 Методы одномерной оптимизации	9
1.3 Числа Фибоначчи	10
1.4 Минимизация метода Фибоначчи.....	12
1.5 Выводы по главе	13
1.6 Постановка задачи	13
2 Алгоритмическое конструирование	14
2.1 Алгоритм метода Фибоначчи	14
2.2 Схема работы программы.....	15
2.3 Выводы по главе	16
3 Программное конструирование	17
3.1 Выбор языка программирования(и среда разработки)	17
3.2 Выбор среды разработки.....	18
3.3 Описание основных подпрограмм	19
3.4 Выводы по главе	20
4 Тестирование программного средства	21
4.1 Описание программного средства	21
4.2 Результаты работы программного средства	23
4.3 Выводы по главе	29
Заключение.....	30
Перечень использованных информационных ресурсов	31
Приложение А.....	32
Приложение Б	37

					УП.260000.000		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.		Костюченко А.И.			Метод Фибоначчи для минимизации функции одной переменной	Лист.	Лист
Провер.		Медведева Т.А.					Листов
							6
							41
					ДГТУ Кафедра «ПОВТуАС»		

Введение

В настоящее время большое значение имеет решение прикладных задач, которые способствуют развитию различных отраслей науки. Значительная часть практических задач связана с методами оптимизации.

В данной работе рассматривается одномерный метод оптимизации, который относится к численным методам поиска безусловных экстремумов.

При рассмотрении практических задач функция $f(x)$ может быть не задана в аналитическом виде или неизвестно, является ли она дифференцируемой. В таких случаях используются численные методы нахождения безусловных экстремумов, к которым относится метод Фибоначчи. Поэтому тема, рассматриваемая в данной работе, является актуальной при решении теоретических и прикладных задач.

Целью данной работы является рассмотрение и реализация алгоритма для нахождения безусловных экстремумов заданных функций методом Фибоначчи, создание рабочей программы на языке Python. В процессе реализации поставленной задачи необходимо изучить теоретический материал по методам оптимизации, в том числе сравнение приближенных методов нахождения точек экстремума, а также выявить преимущества и недостатки исследуемого метода.

1. Теоретический разбор

В данном разделе рассматриваются основная теория метода Фибоначчи для минимизации функции одной переменной . Приведены расчеты формулы и алгоритмы вычисления интервала.

1.1 Экстремум функции

Определение. Точка x_0 из области определения функции f называется *точкой минимума* этой функции, если найдется такая δ -окрестность $[x_0 - \delta; x_0 + \delta]$ точки x_0 , что для всех $x \neq x_0$ из этой окрестности выполняется неравенство:

$$f(x) > f(x_0 \dots)$$

Определение. Точка x_0 из области определения функции f называется *точкой максимума* этой функции, если найдется такая δ -окрестность $[x_0 - \delta; x_0 + \delta]$ точки x_0 , что для всех $x \neq x_0$ из этой окрестности выполняется неравенство:

$$f(x) < f(x_0)$$

Точки минимума и максимума называются *точками экстремума*, а значения функции в этих точках называются *экстремумами функции*.

Необходимое условие экстремума дифференцируемой функции.

Теорема 1. Если функция $f(x)$ дифференцируема в точке c и имеет в этой точке локальный экстремум, то $f'(c) = 0$.

Первое достаточное условие экстремума дифференцируемой функции.

Теорема 2. Пусть точка c является точкой возможного экстремума функции $f(x)$, и пусть функция $f(x)$ дифференцируема всюду в некоторой окрестности точки c . Тогда, если в пределах указанной окрестности производная $f'(x)$ положительна (отрицательна) слева от точки c и отрицательна (положительна) справа от точки c , то функция $f(x)$ имеет в точке c локальный максимум (минимум). Если же $f'(x)$ имеет один и тот же знак слева и справа от

					УП.260000.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

точки c , то экстремума в точке c нет.

Второе достаточное условие экстремума.

Иногда вызывает затруднение исследование знака первой производной $f'(x)$ слева и справа от точки возможного экстремума. На этот случай указывается другое достаточное условие наличия экстремума в данной точке c возможного экстремума, не требующее исследования знака $f'(x)$ в окрестности c , но предполагающее существование в точке c отличной от нуля конечной второй производной $f^{(2)}(x)$.

Теорема 3. Пусть функция $f(x)$ имеет конечную вторую производную в данной точке c возможного экстремума. Тогда функция $f(x)$ имеет в точке c максимум, если $f^{(2)}(c) < 0$, и минимум, если $f^{(2)}(c) > 0$.

Замечание. Теорема 3 имеет более узкую сферу действия, чем теорема 2.

Так, теорема 3 не решает вопроса об экстремуме для случая, когда вторая производная $f^{(2)}(x)$ не существует в точке c , а также для случая, когда $f^{(2)}(x) = 0$. В последнем случае для решения вопроса о наличии экстремума нужно изучить поведение в точке производных высших порядков.

1.2 Методы одномерной оптимизации. Преимущества и недостатки метода

Методы оптимизации имеют большое практическое применение, которое заключается в оптимизации некоторой функции $f(x)$. Такую функцию называют целевой.

Существуют различные методы оптимизации, такие как оптимальный пассивный поиск, метод деления пополам, метод Фибоначчи и золотого сечения, которые основаны на сравнении значений функции $f(x)$ вычисляемых в точках x_1, x_2, \dots, x_n . Это методы так называемого прямого поиска, в котором точки x являются пробными.

					УП.260000.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		9

Метод деления отрезка пополам требует на каждой итерации вычисления двух новых значений функции, поскольку найденные на предыдущем шаге значения не используются. Метод Фибоначчи в этом случае имеет преимущество, т.к. на каждой итерации за исключением первой, требуется одно значение функции. Также преимуществом метода Фибоначчи над остальными является гарантируемое сокращение интервала на заданном отрезке.

Поскольку основной задачей метода Фибоначчи и других методов прямого поиска является сокращение заданного интервала, эффективность метода можно оценивать, посмотрев во сколько раз уменьшилась первоначальная длина интервала, при заданном количестве вычислений. Также большое значение имеет оценка погрешности.

Эффективность методов, также характеризуется числом итераций. при которой необходимо достичь определенной точности ε .

1.3 Числа Фибоначчи

Метод Фибоначчи очень тесно связан с числами, названными в честь великого математика Фибоначчи, настоящее имя которого Леонардо Пизанский. Стоит уделить некоторое внимание биографии ученого, с чьим именем непосредственно связан описываемый метод.

Леонардо Пизанский (около 1170 – около 1250) являлся первым крупным математиком средневековой Европы. Фибоначчи – это прозвище, которое переводится с итальянского как “Хороший сын родился”. Отец его был торговец, поэтому часто путешествовал по разным странам. Чаще всего его отец бывал в Алжире, где юный Леонардо обучался математике у арабских учителей. Он изучал труды исламских математиков, по переводам ознакомился с трудами античных и индийских математиков. Фибоначчи написал ряд математических трактатов на основе усвоенных им знаний. Самый известный труд ученого называется “Книга абака”. В этой книге были изложены все

					УП.260000.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		10

арифметические и алгебраические сведения того времени с большой ясностью и глубиной. В отдельной главе, посвященной арифметической и геометрической прогрессии, ряда квадрата, впервые описана последовательность чисел Фибоначчи.

Помимо числовой последовательности Леонардо Пизанский сделал ряд открытий:

- сформулировал правило для нахождения суммы членов произвольной арифметической прогрессии;
- рассмотрел возвратную последовательность, в которой каждое число, начиная с третьего, равно сумме двух предшествующих ему чисел;
- описал способ приведения дробей к общему знаменателю с помощью нахождения наименьшего общего кратного знаменателя (более рациональный чем использовали арабские математики).
- ввел термин «частное» для обозначения результата деления;
- описал способ приведения дробей к общему знаменателю с помощью нахождения наименьшего общего кратного знаменателя (более рациональный чем использовали арабские математики).

Кроме этого, он разработал свои методы решения различных задач. Леонардо Пизанский внес большой вклад в математику средневековой Европы.

Числа Фибоначчи - элементы числовой последовательности

0,1,1,2,3,5,8,13,21,34,55,... или 1,1,2,3,5,8,13,21,34,55,...

По определению, первые две цифры в последовательности Фибоначчи 0 и 1 (или альтернативно, 1 и 1), и каждое последующее число равно сумме двух предыдущих чисел. Название по имени средневекового математика Леонардо Пизанского (известного как Фибоначчи). Иногда число 0 не рассматривается как член последовательности.

Более формально, последовательность чисел Фибоначчи $\{F_n\}$ задается линейным рекуррентным соотношением:

					УП.260000.000	Лист 11
Изм.	Лист	№ докум.	Подпись	Дата		

$$F_n = F_{n-1} + F_{n-2}, n \geq 2$$

В первой формуле: $F_0 = 0 ; F_1 = 1$

Во второй формуле: $F_0 = 1 ; F_1 = 1$

1.4 Минимизация метода Фибоначчи

Требуется найти безусловный минимум функции $f(x)$ одной переменной, то есть такую точку $x^* \in R$, что $f(x^*) = \min_{x \in R} f(x)$

Для построения эффективного метода одномерной минимизации, работающего по принципу последовательного сокращения интервала неопределенности, следует знать правило выбора на каждом шаге двух внутренних точек. Желательно, чтобы одна из них всегда использовалась в качестве внутренней и для следующего интервала. Тогда количество вычислений функции сократится вдвое и одна итерация потребует расчета только одного нового значения функции. В методе Фибоначчи реализована стратегия, обеспечивающая максимальное гарантированное сокращение интервала неопределенности при заданном количестве вычислений функции и претендующая на оптимальность. Эта стратегия опирается на числа Фибоначчи.

Стратегия поиска. Метод относится к последовательным стратегиям. Задается начальный интервал неопределенности и количество N вычислений функции. Алгоритм уменьшения интервала опирается на анализ значений функции в двух точках (см. рисунок 1). Точки вычисления функции находятся с использованием последовательности из $N + 1$ чисел Фибоначчи. На первой итерации требуются два вычисления функции, а на каждой последующей только по одному. Условия окончания процесса поиска стандартные: поиск заканчивается, когда длина текущего интервала неопределенности оказывается меньше установленной величины.

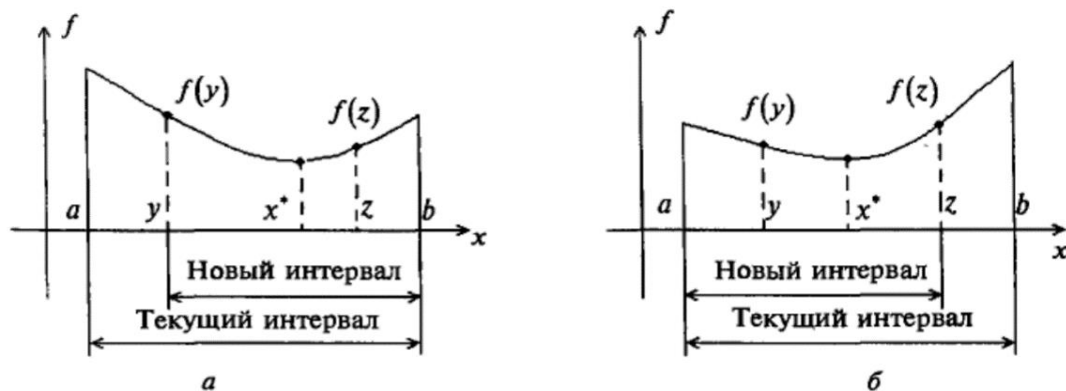


Рисунок 1 – Алгоритм уменьшения интервала

1.5 Выводы по главе

В данной главе рассмотрен теоретический материал, связанный с понятием экстремума и условиями его существования, приведен краткий обзор методов оптимизации функции одной переменной, подробно описан используемый метод Фибоначчи.

1.6 Постановка задачи

Изучить теоретический материал по методам оптимизации функции одной переменной. Разработать программное средство для нахождения экстремумов функции методом Фибоначчи. Тестировать программу с помощью ряда разнотипных функций. Используя встроенные функции системы MathCAD, проверить правильность полученных результатов и визуализировать графики тестируемых функций.

2 Алгоритмическое конструирование

В данном разделе рассматриваются основные алгоритмы работы программного средства: алгоритм минимизации функции одной переменной методом Фибоначчи . Приведено их описание и схема работы.

2.1 Алгоритм метода Фибоначчи

В данном подразделе пошагово описывается основной алгоритм минимизации методом Фибоначчи .

Шаг 1. Задаётся точность e и число вычислений функции n . Присваиваем $j = 1$.

Шаг 2. Вычисляем $x_1 = a + (\frac{F_{n-2}}{F_n} * (b - a) - \frac{(-1)^n}{F_n} * e)$ где F_n, F_{n-2} – числа Фибоначчи

Шаг 3. Вычисляем $x_2 = a + (\frac{F_{n-1}}{F_n} * (b - a) + \frac{(-1)^n}{F_n} * e)$ где F_n, F_{n-1} – числа Фибоначчи

Шаг 4. Вычисляем $f(x_1)$ и $f(x_2)$ где $f(x)$ – заданная функция.

Шаг 5. Если $f(x_1) < f(x_2)$ то :

$$b = x_2;$$

$$x_2 = a + (\frac{F_{n-1}}{F_{n-j+1}} * (b - a) + \frac{(-1)^{n-j+1}}{F_{n-j+1}} * e);$$

$X = x_2$, где X искомая точка

Переходим к шагу 6

Иначе:

$$a = x_1;$$

$$x_1 = a + (\frac{F_{n-1}}{F_{n-j+1}} * (b - a) - \frac{(-1)^{n-j+1}}{F_{n-j+1}} * e);$$

$X = x_2$, где X искомая точка

Переходим к шагу 6.

Шаг 6. Увеличиваем $j = j + 1$ и проверяем условия 6.1 и 6.2 .

6.1. Если $j < n - 1$ переходим к шагу 4

6.2. Если $j \geq n - 1$ то работа программы окончена и искомая точка найдена

2.2 Схема работы программы

В данном подразделе алгоритм минимизации функции одной переменной визуализирована в виде блок-схемы на рисунке 2.

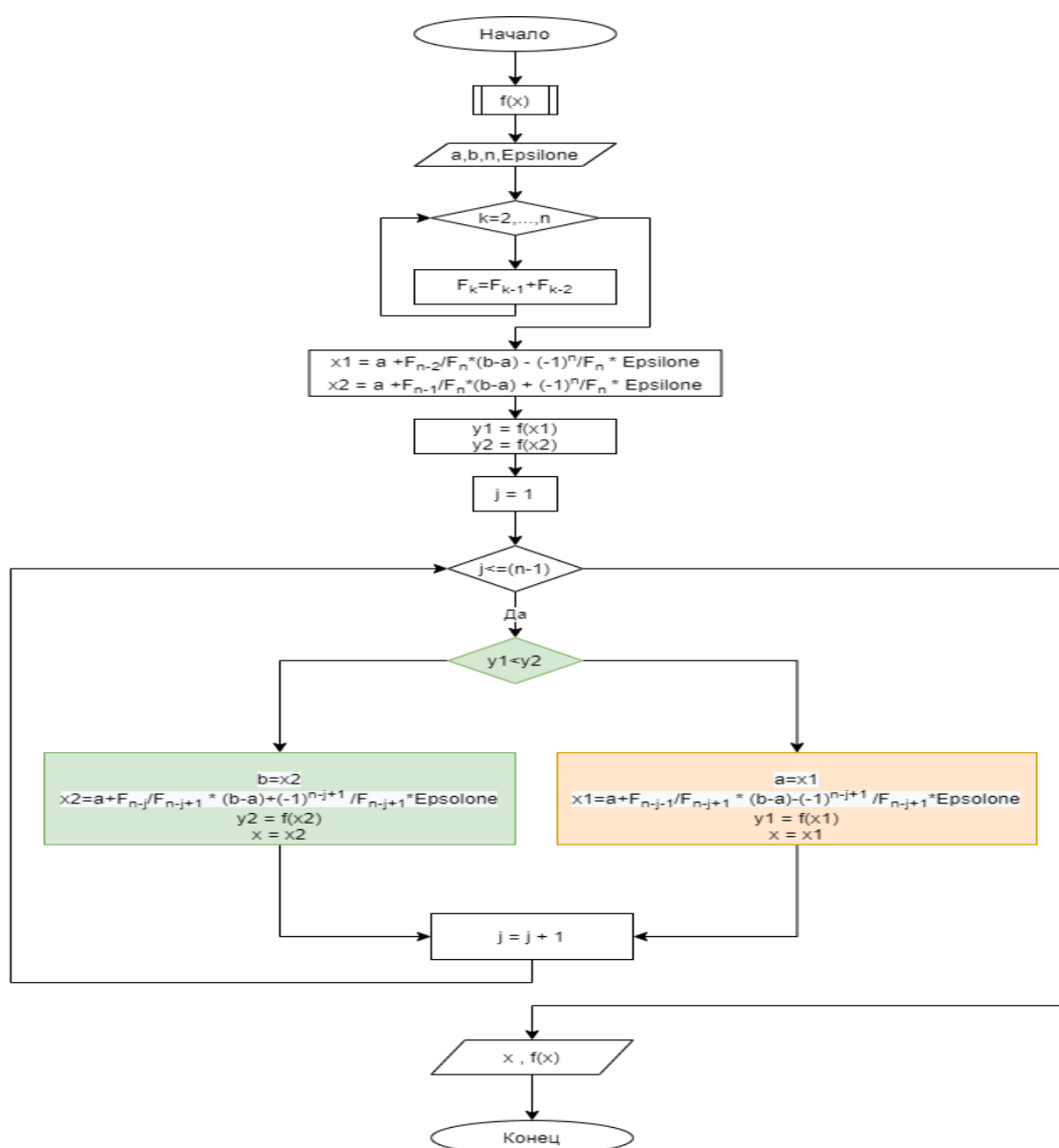


Рисунок 2 – Алгоритм работы программы

2.3 Выводы по главе

В данной главе пошагово и в виде схемы представлен алгоритм позволяющий решить поставленную задачу, и реализовать метод Фибоначчи для минимизации одной переменной за определенное количество шагов с заданной точностью.

					УП.260000.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		16

3 Программное конструирование

В данном разделе будут обоснованы выбор языка программирования, используемый для реализации программы, а также представлены основания выбора среды программирования.

3.1 Выбор языка программирования

Программный код был написан и оформлен на языке программирования Python, графики и расчёты для сравнения были построены с помощью системы MathCAD.

Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным — всё является объектами

Python является мультипарадигмальным языком программирования, поддерживающим императивное, процедурное, структурное, объектно-ориентированное программирование, метапрограммирование и функциональное программирование. Задачи обобщённого программирования решаются за счёт динамической типизации. Аспектно-ориентированное программирование частично поддерживается через декораторы, более полноценная поддержка обеспечивается дополнительными фреймворками. Такие методики как контрактное и логическое программирование можно реализовать с помощью библиотек или расширений. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений с глобальной блокировкой интерпретатора (GIL), высокоуровневые структуры данных.

					УП.260000.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		17

Python стал одним из самых популярных языков, он используется в анализе данных, машинном обучении, DevOps и веб-разработке, а также в других сферах, включая разработку игр. За счёт читабельности, простого синтаксиса и отсутствия необходимости в компиляции язык хорошо подходит для обучения программированию, позволяя концентрироваться на изучении алгоритмов, концептов и парадигм. Применяется язык многими крупными компаниями, такими как Google или Facebook. По состоянию на апрель 2021 года Python занимает третье место в рейтинге TIOBE популярности языков программирования с показателем 11,03%. «Языком года» по версии TIOBE Python объявлялся в 2007, 2010, 2018 и 2020 году [5].

Язык обладает чётким и последовательным синтаксисом, продуманной модульностью и масштабируемостью, благодаря чему исходный код написанных на Python программ легко читаем. При передаче аргументов в функции Python использует вызов по соиспользованию

Популярная компьютерная система MathCAD (Mathematical Computer Aided Design – Математическая система автоматизированного проектирования) является наиболее универсальной математически ориентированной системой, обладающей как возможностями численных и аналитических (символьных) вычислений, так и средствами оформления документов на высоком профессиональном уровне. Отличительной чертой MathCAD является объединение в одном рабочем документе математического описания алгоритма решения задач, заданного в виде привычных математических формул и символов с комментариями и результатами вычислений в виде чисел, таблиц и различных графиков. Библиотеки и программные пакеты расширения системы обеспечивают ее применение для автоматизации решения математических задач в различных областях науки, техники и образования.

3.2 Выбор среды разработки

					УП.260000.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		18

Visual Studio Code — это редактор исходного кода. Он поддерживает ряд языков программирования, подсветку синтаксиса, IntelliSense, рефакторинг, отладку, навигацию по коду, поддержку Git и другие возможности. Многие возможности Visual Studio Code недоступны через графический интерфейс, зачастую они используются через палитру команд или JSON-файлы (например, пользовательские настройки). Палитра команд представляет собой подобие командной строки, которая вызывается сочетанием клавиш.

Visual Studio также позволяет заменять кодовую страницу при сохранении документа, символы перевода строки и язык программирования текущего документа.

С 2018 года появилось расширение Python для Visual Studio Code с открытым исходным кодом. Оно предоставляет разработчикам широкие возможности для редактирования, отладки и тестирования кода.

На март 2019 года посредством встроенного в продукт пользовательского интерфейса можно загрузить и установить несколько тысяч расширений только в категории «programming languages» (языки программирования).

Также расширения позволяют получить более удобный доступ к программам, таким как Docker, Git и другие. В расширениях можно найти линтеры кода, темы для редактора и поддержку синтаксиса отдельных языков

3.3 Описание основных модулей программы

Программный код написан и оформлен на языке высокого уровня – Python в среде Visual Studio Code.

Для визуализации графиков выбрана библиотека Matplotlib.

Для создания массивов и хранения в них используемых данных выбрана библиотека Numpy.

Для создания пользовательского интерфейса выбрана библиотека PyQt5.

Для работы с директорией программы выбрана программа OS

					УП.260000.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		19

Для вычисления функции используется функция из основного файла программного средства `create_mod()` генерирующая файл-модуль «agrigator_func» в котором хранится функция `matfunc()` полученная через функцию основного файла `intoequa()` которая хранит в себе формулу введенную пользователем и переведенную в формат пригодный для работы с ней в среде Python . Также файл-модуль хранит в себе значение функции введенной пользователем ранее , которое в случае аварийного прекращения работы программы вернет исследуемую функцию .

В функцию `create_mod()` передается параметр «text» типа `string` , предварительно прошедший через функцию-агрегатор `intoequa()` , после чего функция создает файл-модуль , из которого позднее основной файл программного средства подгружает функцию `matfunc()` которая возвращает значение функции в точке.

Для корректной работы основной программного средства , при первом ее запуске проверяется есть ли файл-модуль , если есть то из него подкачивается предыдущая функция , если файла нет то он создается с функцией по умолчанию $(x - 2) * \cos(x)$.

3.4 Выводы по главе

В данной главе был обоснован выбор языка программирования, а также среда программирования для создаваемого программного средства. Кроме того, были описаны основные подпрограммы программного средства. В таблице было приведено описание каждой из подпрограмм с указанием типа входных параметров и возвращаемого значения.

4 Тестирование программного средства

В данном разделе будет описано программное средство и протестировано 3 контрольных примера.

4.1 Описание программного средства

Для нахождения минимума и максимума функции, в качестве входных данных, вводятся значения в следующем порядке:

(Классы и методы которые в коде)

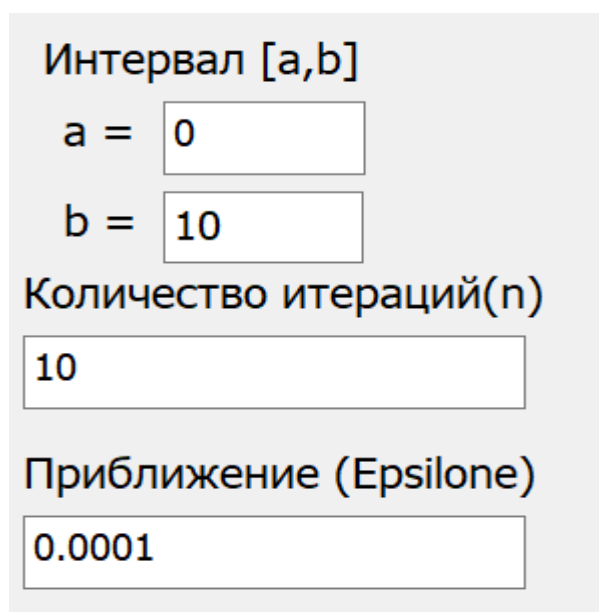
$F(x)$ – уравнение;

a, b – интервал поиска;

n – количество вычислений целевой функции;

Epsilone – точность поиска;

Все значения имеют тип данных float.



Интервал $[a,b]$

$a =$

$b =$

Количество итераций(n)

Приближение (Epsilone)

Рисунок 3 – пример ввода/вывода данных

В результате выполнения программы на экран выводятся точки минимума и максимума, а также соответствующие им значения целевой функции (см. рисунок 4).

```
Xmin = 3.2615
F(Xmin) = -1.2524
Xmax = 6.6927
F(Xmax) = 4.3047
```

Рисунок 4 – Пример выполнения программы

При произведении вычислений задействованы такие классы и функции как :

- Ui_MainWindow() – Класс который отвечает за графический интерфейс
- intoequa() – функция-агрегатор , переводит функцию написанную пользователем в вид пригодный для математических вычислений в среде Python с помощью библиотеки Math
- create_graph() – функция генерирующая график на основе полученной функции и массива точек полученных с помощью встроенной функции Numpy , linspace()
- create_mod() – функция генерирующая файл-модуль который содержит в себе формулу созданную через intoequa()

На рисунке 2 изображен пользовательский интерфейс, в котором 5 полей:

1. Ввод функции одной переменной
2. Выбор начальной точки интервала
3. Выбор конечной точки интервала
4. Выбор количества итераций метода

5. Выбор точности расчетов

The screenshot shows a user interface for a mathematical software application. On the left, there are several input fields and buttons. At the top, a text box for $f(x) =$ is followed by a 'ДОБАВИТЬ' (Add) button. Below this, there are input fields for 'Интервал [a,b]' with sub-fields for 'a =' and 'b ='. Further down are fields for 'Количество итераций(n)' and 'Приближение (Epsilon)'. A 'ВЫЧИСЛИТЬ' (Calculate) button is positioned below these. At the bottom left, there are labels for 'Xmin =', 'F(Xmin) =', 'Xmax =', and 'F(Xmax) ='. On the right side of the interface is a graph of a function on a grid. The x-axis is labeled from 1.0 to 5.0 in increments of 0.5. The function curve starts at approximately (1.0, -0.5), rises to a peak of about 0.5 at x=2.5, and then falls to approximately -1.5 at x=5.0.

Рисунок 5 – Пользовательский интерфейс

После заполнения поля функции « $f(x)$ » необходимо нажать на кнопку «ДОБАВИТЬ», чтобы программа создала математическую функцию с которой будет работать. После добавления математической функции и заполнения оставшихся полей, необходимо нажать на «ВЫЧИСЛИТЬ» чтобы программа вычислила экстремумы функции и их значения на заданном промежутке.

4.2 Результаты работы программного средства

Для проверки работы программного модуля решение приведено тремя способами: нахождение экстремумов (Python 3.8), графическое построение и расчеты при помощи встроенных функций в среде MathCAD.

1. В результате выполнения программы (Python 3.8) найдены точки минимума и максимума, а также значения целевой функции в этих точках. Функция: $y(x) = (x - 2) * \cos(x)$, на промежутке $[1, 8]$. Решение приведено на рисунке 6.

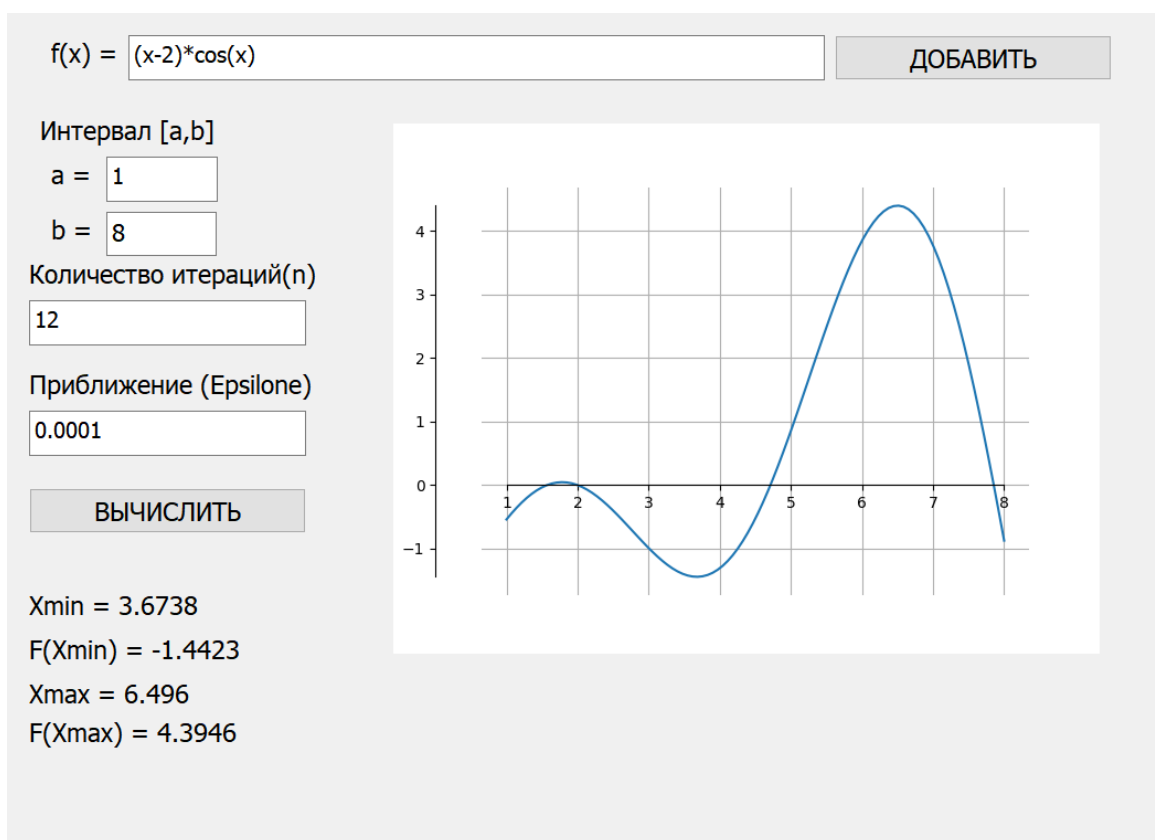


Рисунок 6 – Пример нахождения экстремумов функции

Рассмотрим график функции построенный в среде MathCAD на рисунке 6, и решение с помощью стандартных функций MathCAD на рисунке 7.

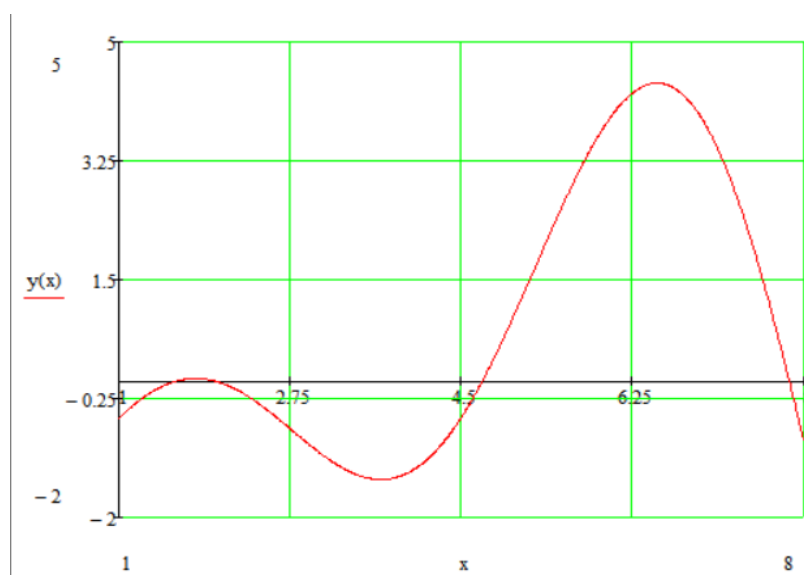


Рисунок 7 – Построение графика функции в среде MathCAD

$$y(x) := (x - 2) \cdot \cos(x)$$

$$x := 3$$

$$x_{\min} := \text{root}\left(\frac{d}{dx}y(x), x\right) = 3.6788$$

$$x := 7$$

$$x_{\max} := \text{root}\left(\frac{d}{dx}y(x), x\right) = 6.5018$$

$$y(x_{\min}) = -1.4423$$

$$y(x_{\max}) = 4.3947$$

Рисунок 8 – Решение в среде mathCAD

2. В качестве второго примера рассматривается функция $y(x) = (1 - \sin(x)) \cdot \cos(x)$, на промежутке $[2, 8]$. Пример работы программы Python приведен на рисунке 9.

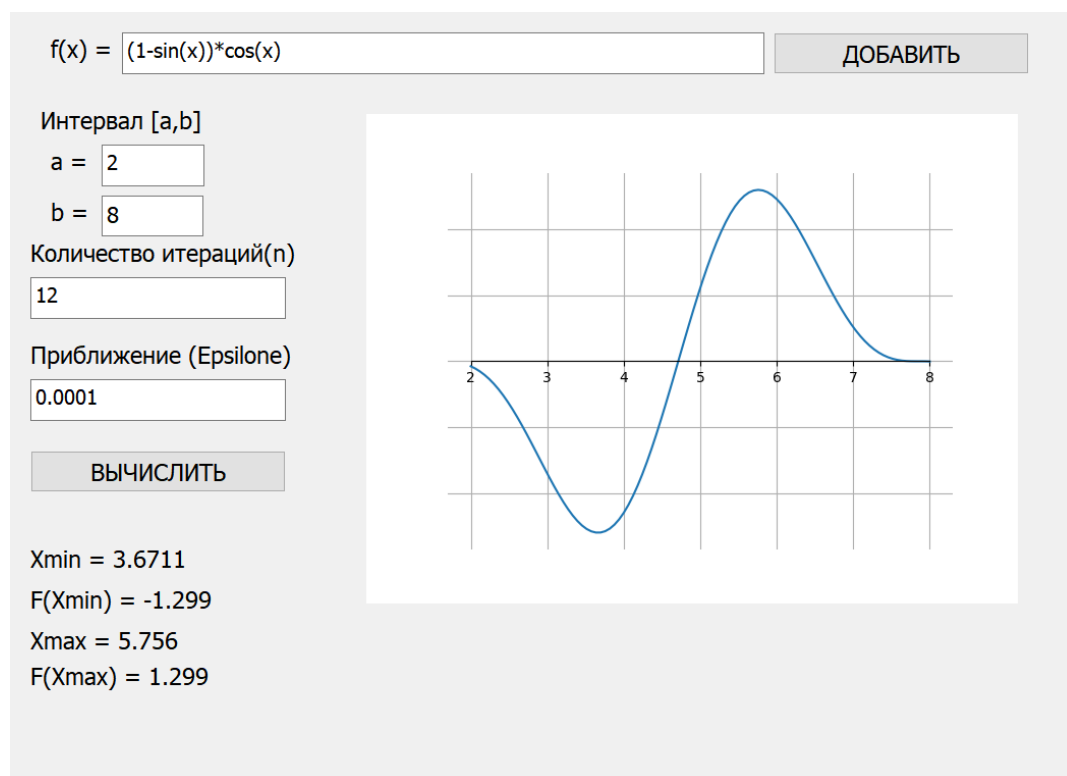


Рисунок 9 – Полученные результаты программы

Построение графика и решение указаны на рисунках 10 и 11

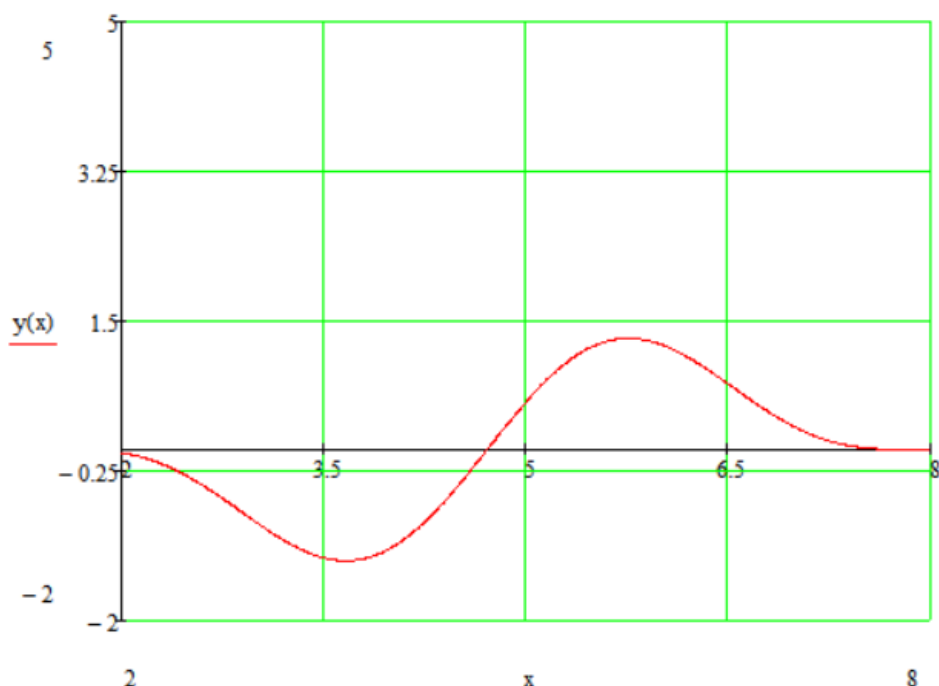


Рисунок 10 – Визуализация графика в среде mathCAD

$$y(x) := (1 - \sin(x)) \cdot \cos(x)$$

$$x := 4$$

$$x_{\min} := \text{root}\left(\frac{d}{dx}y(x), x\right) = 3.6652$$

$$x := 6$$

$$x_{\max} := \text{root}\left(\frac{d}{dx}y(x), x\right) = 5.7596$$

$$y(x_{\min}) = -1.299$$

$$y(x_{\max}) = 1.299$$

Рисунок 11 – Вычисления точек минимума и максимума в среде mathCAD

3. В качестве второго примера рассматривается функция $y(x) = (\sin(x)^2 - (-6) * \cos(x)) * (x^2 - 15)$, на промежутке $[1, 6]$. Пример работы

программы Python приведен на рисунке 12.

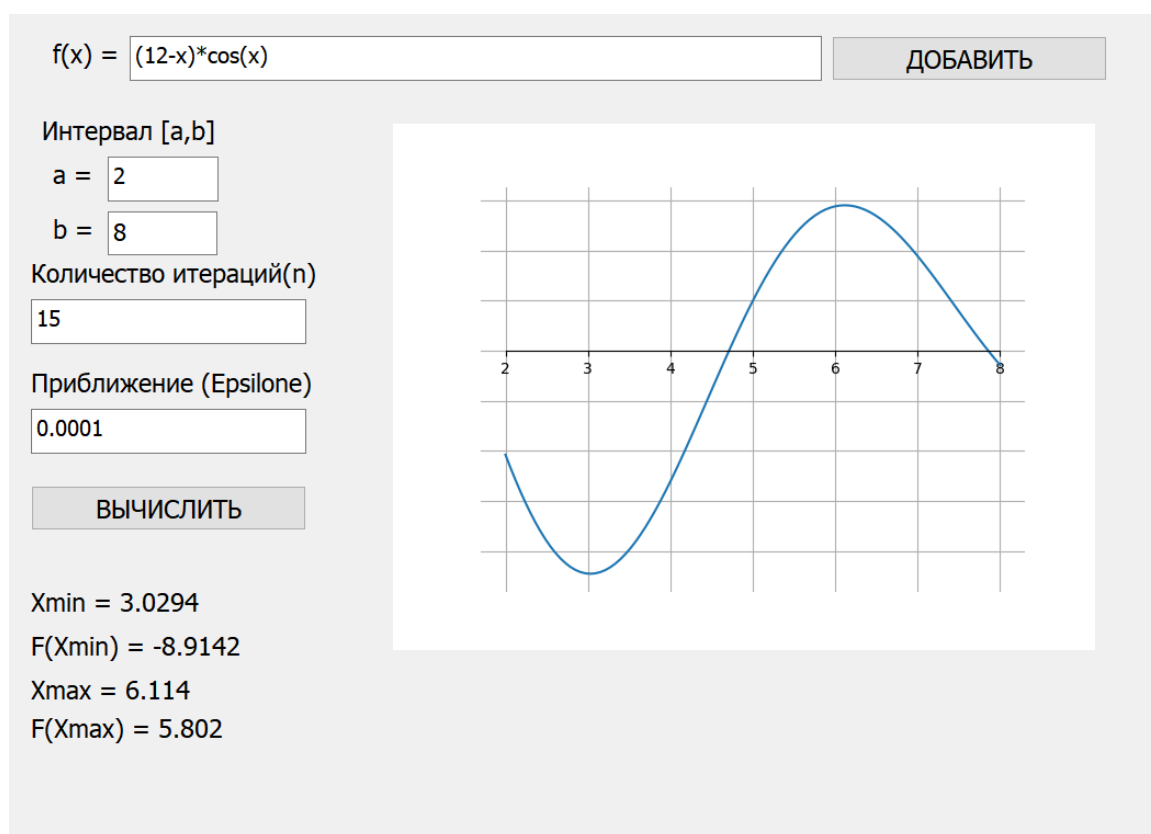


Рисунок 12 – Результаты программного средства

Построение графика и решение в среде MathCAD указаны на рисунках 13 и 14.

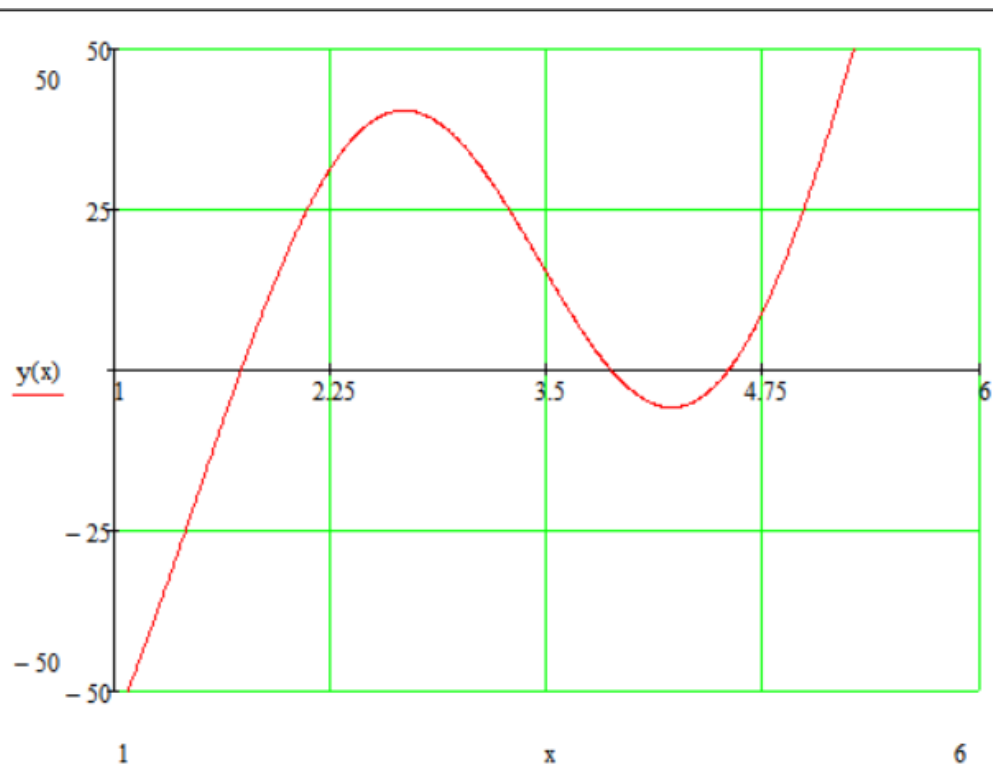


Рисунок 13 – Визуализация графика в среде mathCAD

$$y(x) := [\sin(x)^2 - (-6)\cos(x)] \cdot (x^2 - 15)$$

$$x := 1$$

$$x_{\min} := \text{root}\left(\frac{d}{dx}y(x), x\right) = 0$$

$$x := 5$$

$$x_{\max} := \text{root}\left(\frac{d}{dx}y(x), x\right) = 4.2219$$

$$y(x_{\min}) = -90$$

$$y(x_{\max}) = -5.7851$$

Рисунок 14 – Решение в среде mathCAD

4.3 Выводы по главе

На основании рассмотренных примеров можно сделать вывод о том, что результаты, полученные в процессе работы программного средства с заданной точностью совпадают с результатами расчетов с помощью встроенных функций среды MathCAD и найденные экстремумы исследуемых функций проиллюстрированы графиками

					УП.260000.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		29

Заключение

В данной работе рассмотрен один из ключевых методов прямого поиска метод Фибоначчи, а также разобран алгоритм метода. В результате выполнения задания учебной практики реализовано программное средство, разработанное на языке Python, которое позволяет решать задачу нахождения экстремумов функции за заданное количество шагов.

Данное программное средство может применяться в учебных целях: в предметах связанных с методами оптимизации, в рамках ВУЗа в предметах, связанных с вычислительной математикой, как средство демонстрации работы изучаемых методов .

В ходе выполнения работы был построен аналитически обоснованный алгоритм решения поставленной задачи и разработано соответствующее программное средство, конечный исходный код которого можно просмотреть в Приложении Б. Таким образом была решена поставленная задача и выполнена проверка корректной работоспособности программного средства.

					УП.260000.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		30

Перечень использованных информационных ресурсов

1. Пантелеев А.В. Методы оптимизации в примерах и задачах: Учеб. Пособие/А.В. Пантелеев, Т.А. Летова. – 2-е изд., исправл. – М.: Высш. шк., 2005. – 544 с.: ил.
2. Соболев Б.В. Методы оптимизации: практикум / Б.В. Соболев, Б.Ч.Месхи, Г.И.Каныгин. – Ростов н/Д : Феникс, 2009. – 380, [4] с. (Высшее образование).
3. Моисеев Н.Н., Иванилов Ю.П., Столярова Е.М. Методы оптимизации - М.: Наука, 1978
4. Н. Н. Воробьев. Числа Фибоначчи. — Наука, 1978
- 5.URL: <http://all4study.ru/proektirovanie/metodfibonachci.html>
6. URL:http://optimizaciyasapr.narod.ru/bez_odnomer/fibonachhi4.html
7. URL: <https://ru.wikipedia.org/wiki/Mathcad>
8. URL: <https://ru.wikipedia.org/wiki/C%2B%2B>
9. URL: <https://ru.wikipedia.org/wiki/Python>

Приложение А Техническое задание

СОГЛАСОВАНО

Доцент каф. «ПОВТиАС»

_____/Медведева Т.А

« ____ » _____ 2020г.

А.1 Введение

А.1.1 Наименование программного средства

Наименование программного средства – «Метод Фибоначчи для функции одной переменной».

А.1.2 Область применения

Программное средство может применяться в учебных целях: в предметах связанных с методами оптимизации, в рамках ВУЗа в предметах, связанных с вычислительной математикой, как средство демонстрации работы изучаемых методов ,

А.2 Основание для разработки

Разработка ведется на основании документа «Учебный план для студентов ВУЗа» направление 02.03.03 «Математическое обеспечение и администрирование информационных систем» кафедры «Программное обеспечение вычислительной техники и автоматизированных систем» факультета «Информатика и вычислительная техника» Донского Государственного Технического Университета.

					УП.260000.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		32

А.3 Назначение разработки

А.3.1 Функциональное назначение

Функциональное назначение программного средства заключается в минимизации функции одной переменной методом Фибоначчи.

А.3.2 Эксплуатационное назначение

Эксплуатационное назначение состоит в использовании программного средства на персональном компьютере (ПК) с операционной системой Windows.

А.4 Требования к программе

А.4.1 Требования к функциональным характеристикам

Программное средство должно осуществлять следующие функции:

- Ввод произвольного нелинейного уравнения и начальных данных пользователем.
- Вывод точек максимума и минимума функции и значения функции в этих точках.
- Вывод графика функции.
- Обработка ошибок ввода.

А.4.2 Требования к надежности

Для надежной работы программного средства необходимо, чтобы выполнялись следующие условия:

- бесперебойное питание технического средства, на котором находится продукт;

					УП.260000.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		33

- регулярная проверка программного средства на наличие вирусов;
- отсутствие шума и спокойная обстановка.

А.4.2.1 Входные данные

Входными данными являются: функция, интервал , требуемая точность, количество итераций.

А.4.2.2 Выходные данные

Выходными данными являются: точки минимума минимума и максимума , значения функции в точках минимума и максимума , визуализация графика

А.4.3 Условия эксплуатации

Для стабильного функционирования и оптимальной работы программного продукта необходимо соблюдение всех требований и правил эксплуатации вычислительной техники. Каких-либо требований к пользователю данного приложения нет.

А.4.4 Требование к составу и параметрам технических средств

В состав технических средств должен входить ПК с операционной системой windows 7 и выше, включающая в себя:

- 64-разрядный процессор;
- оперативная память объемом не менее 1 Гбайт.

Дополнительные требования и ограничения к составу и параметрам технических средств не вводятся.

А.4.5 Требование к информационной и программной совместимости

					УП.260000.000	Лист
						34
Изм.	Лист	№ докум.	Подпись	Дата		

Для функционирования программного средства необходима операционная система Windows 7 и выше.

Пользовательский интерфейс должен быть интуитивно понятным и содержать подсказки.

Исходные коды программы должны быть реализованы на языке Python. В качестве интегрированной среды разработки программы должна быть использована среда Visual Studio Code.

А.4.6 Требования к упаковке и маркировке

К упаковке и маркировке специальных требований не предъявляются.

А.4.7 Требования к транспортировке и хранению

Условия транспортирования, места хранения, условия складирования и сроки хранения в различных условиях должны соответствовать требованиям, предъявляемым к носителям информации, на которых будет содержаться данное программное изделие. Программное средство может храниться на любых цифровых носителях информации (жесткий диск, компакт – диск, флэш накопитель и т. п.).

А.5 Требование к программной документации

Программная документация состоит из следующего:

- титульный лист;
- лист задания на учебную ознакомительную практику;
- пояснительная записка к учебной ознакомительной практике;
- техническое задание по ГОСТ 19.201-78 ЕСПД;
- исходный код программного средства по ГОСТ 19.401-79 ЕСПД.

А.6 Стадии и этапы разработки

Реализация программного средства состояла из следующих этапов:

					УП.260000.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		35

- исходный код программного средства по ГОСТ 19.401-79 ЕСПД.
- постановка задачи (с 11.02.2021 по 12.02.2021);
- изучение предметной области (с 13.02.2021 по 17.03.2021);
- алгоритмическое конструирование (с 18.03.2021 по 07.04.2021);
- программная реализация (с 07.04.2021 по 28.04.2021);
- тестирование приложения (с 29.04.2021 по 25.05.2021);
- разработка отчета (с 26.05.2021 по 05.06.2021).

А.7 Порядок и контроль приемки

Порядок и контроль приемки определяются заведующим кафедрой «ПОВТ и АС», подразумевающие собой демонстрацию показателя владения средствами для разработки программных средств в различных направлениях.

Главным требованием к приемке является наличие корректного работающего программного средства и отчета, предоставленного в печатном виде.

Разработчик технического задания:

Костюченко Артём Иванович

«_____» _____ 2021 г.

					УП.260000.000	Лист
Изм.	Лист	№ докум.	Подпись	Дата		36

Приложение Б

Исходный код программного средства

Полный код программного средства находится на съемном носителе, прилагаемом к данной работе.

```
# -*- coding: utf-8 -*-
```

```
from agrigator_func import input_form
import os
from sys import argv, executable
```

```
import math,matplotlib.pyplot as plt,numpy as np
from PyQt5 import QtCore, QtGui, QtWidgets
```

```
if os.path.isfile('agrigator_func.py'):
    from agrigator_func import matfunc
else:
    with open("agrigator_func.py","w") as fl:
        print(f"import math
```

```
def matfunc(x):
    print("(x-2)*math.cos(x)")
    return((x-2)*math.cos(x))
```

```
def input_form():
    return("")
"",file=fl)
    from agrigator_func import matfunc
```

```
# корректировка файла для собственного модуля
def intoequa(text = str()):
```

```
    if text.find("arccos") != -1:
        text = text.replace("cos","math.acos")
    if text.find("arcsin") != -1:
        text = text.replace("sin","math.asin")
    if text.find("cos") != -1:
        text = text.replace("cos","math.cos")
    if text.find("sin") != -1:
        text = text.replace("sin","math.sin")
    if text.find("tg") != -1:
        text = text.replace("tg","math.tan")
    if text.find("arctg") != -1:
```

					УП.260000.000	Лист
						37
Изм.	Лист	№ докум.	Подпись	Дата		

```

    text = text.replace("arctg","math.atan")
if text.find("ctg") != -1:
    text = text.replace("ctg", "(-1)/math.sin")
if text.find("exp") != -1:
    text = text.replace("exp", "math.exp")
if text.find("^") != -1:
    text = text.replace("")

    return(text)
# для расчета функций
def f(x):
    return -matfunc(x)
def func(x):
    return matfunc(x)

#График функции
def create_graph(a,b):
    ox = np.linspace(a,b,100)
    oy = []
    for value in ox:
        oy.append(func(value))
    fig = plt.figure()
    plt.plot(ox,oy)
    plt.grid(True)
    ax = fig.add_subplot(111)
    ax.spines['left'].set_position('zero')
    ax.spines['right'].set_color('none')
    ax.spines['bottom'].set_position('zero')
    ax.spines['top'].set_color('none')
    ax.spines['left'].set_smart_bounds(True) # 3.8
    ax.spines['bottom'].set_smart_bounds(True) # 3.8
    ax.xaxis.set_ticks_position('bottom')
    ax.yaxis.set_ticks_position('left')
    fig.savefig('figure.png',dpi = 100)

# создание отдельного файла для вычислений
def create_mod(text):
    with open("agrigator_func.py","w") as fl:
        print(f"import math

def matfunc(x):
    print("{text}")
    return({text})

```

```

def input_form():
    text1 = "{text}"
    text1 = text1.replace("math.", "")
    return(text1)
'',file=fl)
print("Модуль записан")

# Интерфейс

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(1051, 755)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.label = QtWidgets.QLabel(self.centralwidget)
        self.label.setGeometry(QtCore.QRect(20, 520, 221, 31))
        font = QtGui.QFont()
        font.setPointSize(14)
        self.label.setFont(font)
        self.label.setObjectName("label")
        self.pushButton_2 = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton_2.setGeometry(QtCore.QRect(750, 20, 251, 41))
        font = QtGui.QFont()
        font.setPointSize(14)
        self.pushButton_2.setFont(font)
        self.pushButton_2.setObjectName("pushButton_2")
        self.label_2 = QtWidgets.QLabel(self.centralwidget)
        self.label_2.setGeometry(QtCore.QRect(350, 100, 640, 480))
        self.label_2.setMinimumSize(QtCore.QSize(640, 480))
        self.label_2.setMaximumSize(QtCore.QSize(640, 480))
        self.label_2.setText("")
        self.label_2.setPixmap(QtGui.QPixmap("figure.png"))
        self.label_3 = QtWidgets.QLabel(self.centralwidget)
        self.label_3.setGeometry(QtCore.QRect(20, 600, 251, 31))
        font = QtGui.QFont()
        font.setPointSize(14)
        self.label_3.setFont(font)
        self.label_3.setObjectName("label_3")
        self.label_4 = QtWidgets.QLabel(self.centralwidget)
        self.label_4.setGeometry(QtCore.QRect(20, 560, 221, 31))
        font = QtGui.QFont()

```

```

font.setPointSize(14)
self.label_4.setFont(font)
self.label_4.setObjectName("label_4")
self.label_5 = QtWidgets.QLabel(self.centralwidget)
self.label_5.setGeometry(QtCore.QRect(20, 630, 241, 41))
font = QtGui.QFont()
font.setPointSize(14)
self.label_5.setFont(font)
self.label_5.setObjectName("label_5")
self.pushButton = QtWidgets.QPushButton(self.centralwidget)
self.pushButton.setGeometry(QtCore.QRect(20, 430, 251, 41))
font = QtGui.QFont()
font.setPointSize(14)
self.pushButton.setFont(font)
self.pushButton.setObjectName("pushButton")
self.label_6 = QtWidgets.QLabel(self.centralwidget)
self.label_6.setGeometry(QtCore.QRect(40, 130, 51, 31))
font = QtGui.QFont()
font.setPointSize(14)
self.label_6.setFont(font)
self.label_6.setObjectName("label_6")
self.label_7 = QtWidgets.QLabel(self.centralwidget)
self.label_7.setGeometry(QtCore.QRect(40, 180, 51, 31))
font = QtGui.QFont()
font.setPointSize(14)
self.label_7.setFont(font)
self.label_7.setObjectName("label_7")
self.label_8 = QtWidgets.QLabel(self.centralwidget)
self.label_8.setGeometry(QtCore.QRect(30, 90, 171, 31))
font = QtGui.QFont()
font.setPointSize(14)
self.label_8.setFont(font)
self.label_8.setObjectName("label_8")
self.label_9 = QtWidgets.QLabel(self.centralwidget)
self.label_9.setGeometry(QtCore.QRect(20, 220, 261, 31))
font = QtGui.QFont()
font.setPointSize(14)
self.label_9.setFont(font)
self.label_9.setObjectName("label_9")
self.label_10 = QtWidgets.QLabel(self.centralwidget)
self.label_10.setGeometry(QtCore.QRect(20, 320, 271, 31))
font = QtGui.QFont()
font.setPointSize(14)
self.label_10.setFont(font)

```



```

self.label_10.setObjectName("label_10")
self.plainTextEdit = QtWidgets.QPlainTextEdit(self.centralwidget)
self.plainTextEdit.setGeometry(QtCore.QRect(90, 130, 101, 41))
font = QtGui.QFont()
font.setPointSize(12)
self.plainTextEdit.setFont(font)
self.plainTextEdit.setObjectName("plainTextEdit")
self.plainTextEdit_2 = QtWidgets.QPlainTextEdit(self.centralwidget)
self.plainTextEdit_2.setGeometry(QtCore.QRect(90, 180, 100, 40))
self.plainTextEdit_2.setMaximumSize(QtCore.QSize(100, 40))
font = QtGui.QFont()
font.setPointSize(13)
self.plainTextEdit_2.setFont(font)
self.plainTextEdit_2.setObjectName("plainTextEdit_2")
self.plainTextEdit_3 = QtWidgets.QPlainTextEdit(self.centralwidget)
self.plainTextEdit_3.setGeometry(QtCore.QRect(20, 260, 251, 41))
font = QtGui.QFont()
font.setPointSize(12)
self.plainTextEdit_3.setFont(font)
self.plainTextEdit_3.setObjectName("plainTextEdit_3")
self.plainTextEdit_4 = QtWidgets.QPlainTextEdit(self.centralwidget)
self.plainTextEdit_4.setGeometry(QtCore.QRect(20, 360, 251, 41))
font = QtGui.QFont()
font.setPointSize(12)
self.plainTextEdit_4.setFont(font)
self.plainTextEdit_4.setObjectName("plainTextEdit_4")
self.plainTextEdit_5 = QtWidgets.QPlainTextEdit(self.centralwidget)
self.plainTextEdit_5.setGeometry(QtCore.QRect(110, 20, 631, 41))
font = QtGui.QFont()
font.setPointSize(12)
self.plainTextEdit_5.setFont(font)
self.plainTextEdit_5.setObjectName("plainTextEdit_5")
self.label_11 = QtWidgets.QLabel(self.centralwidget)
self.label_11.setGeometry(QtCore.QRect(40, 20, 71, 31))
font = QtGui.QFont()
font.setPointSize(14)
self.label_11.setFont(font)
self.label_11.setObjectName("label_11")
MainWindow.setCentralWidget(self.centralwidget)
self.menubar = QtWidgets.QMenuBar(MainWindow)
self.menubar.setGeometry(QtCore.QRect(0, 0, 1051, 26))
self.menubar.setObjectName("menubar")
MainWindow.setMenuBar(self.menubar)
self.statusbar = QtWidgets.QStatusBar(MainWindow)

```

```

self.statusbar.setObjectName("statusbar")
MainWindow.setStatusBar(self.statusbar)

self.retranslateUi(MainWindow)
QtCore.QMetaObject.connectSlotsByName(MainWindow)

def retranslateUi(self, MainWindow):
    from agrigator_func import input_form
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("Main-
Window", "Метод Фибоначчи для функции одной переменной"))
    self.label.setText(_translate("MainWindow", "Xmin = "))
    self.label_3.setText(_translate("MainWindow", "Xmax = "))
    self.label_4.setText(_translate("MainWindow", "F(Xmin) = "))
    self.label_5.setText(_translate("MainWindow", "F(Xmax) = "))
    self.pushButton.setText(_translate("MainWindow", "ВЫЧИСЛИТЬ"))
    self.pushButton.clicked.connect(self.main_func)
    self.pushButton_2.clicked.connect(self.create_module)
    self.label_6.setText(_translate("MainWindow", "a = "))
    self.label_7.setText(_translate("MainWindow", "b = "))
    self.label_8.setText(_translate("MainWindow", "Интервал [a,b]"))
    self.label_9.setText(_translate("MainWindow", "Количество итераций(n)"))
    self.label_10.setText(_translate("MainWindow", "Приближение (Epsilone)"))
    self.plainTextEdit.setPlainText(_translate("MainWindow", ""))
    self.plainTextEdit_2.setPlainText(_translate("MainWindow", ""))
    self.plainTextEdit_3.setPlainText(_translate("MainWindow", ""))
    self.plainTextEdit_4.setPlainText(_translate("MainWindow", "0.0001"))
    self.plainTextEdit_5.setPlainText(_translate("MainWindow", f"{in-
put_form()}"))
    self.label_11.setText(_translate("MainWindow", "f(x) = "))
    self.pushButton_2.setText(_translate("MainWindow", "ДОБАВИТЬ"))

def create_module(self):
    from sys import argv, executable
    import os
    equa = self.plainTextEdit_5.toPlainText()
    equa = intoequa(equa)
    equa = create_mod(equa)
    os.execl(executable, os.path.abspath(__file__), *argv)

def main_func(self):

    equa = self.plainTextEdit_5.toPlainText()
    from agrigator_func import matfunc

```

```

# Интервал
a = int(self.plainTextEdit.toPlainText())
a1 = a
b = int(self.plainTextEdit_2.toPlainText())
b1 = b
# Количество вычислений
N = int(self.plainTextEdit_3.toPlainText())
N1 = N
Epsilone = float(self.plainTextEdit_4.toPlainText())

# Создание графика

create_graph(a,b)
self.label_2.setPixmap(QtGui.QPixmap("figure.png"))

# Входные данные
Fib = [1,1]

for i in range(2,N+1):
    Fib.append(Fib[i-1]+Fib[i-2])

if N % 2 == 0:
    sign = 1
else:
    sign = -1

x1 = a + ((Fib[N-2] * (b-a) - sign*Epsilone)/Fib[N])
x2 = a + ((Fib[N-1]*(b-a) + sign*Epsilone)/Fib[N])

f1 = f(x1)
f2 = f(x2)

j = 1

# Нахождение максимума
while (j <= (N-1)):

    if ((N - j + 1)% 2 == 0):
        sign = 1
    else:
        sign = -1

```

```

if (f1 <= f2):
    b = x2
    x2 = x1
    f2 = f1
    x1 = a + ((Fib[N - j - 1]*(b-a)-sign * Epsilone)/Fib[N-j+1])
    f1 = f(x1)
    x = x2
else:
    a = x1
    x1 = x2
    f1 = f2
    x2 = a + ((Fib[N - j] * (b-a) + sign * Epsilone)/Fib[N-j+1])
    f2 = f(x2)
    x = x1
j += 1

```

Перезапись(дублирование) входных данных

```

Fib1 = [1,1]
for i in range(2,N+1):
    Fib1.append(Fib1[i-1]+Fib1[i-2])

```

```

if N1 % 2 == 0:
    sign1 = 1
else:
    sign1 = -1

```

```

x3 = a1 + ((Fib1[N1-2]* (b1-a1) - sign1*Epsilone) / Fib1[N1])
x4 = a1 + ((Fib1[N1-1]* (b1-a1) + sign1*Epsilone) / Fib1[N1])

```

```

f3 = func(x3)
f4 = func(x4)
j1 = 1

```

```

# Нахождение максимума
while (j1 <= (N1-1)):
    if ((N1 - j1 + 1)% 2 == 0):
        sign1 = 1
    else:
        sign1 = -1

    if (f3 <= f4):

```

```

        b1 = x4
        x4 = x3
        f4 = f3
        x3 = a1 + ((Fib1[N1 - j1 - 1] * (b1-a1) - sign1 * Epsilone)/Fib1[N1-j1+1])
        f3 = func(x3)
        max = x4
    else:
        a1 = x3
        x3 = x4
        f3 = f4
        x4 = a1 + ((Fib1[N1 - j1] * (b1-a1) + sign1 * Epsilone)/Fib1[N1-j1+1])
        f4 = func(x4)
        max = x3
    j1 += 1

rou_num = len(str(Epsilone))-2

_translate = QtCore.QCoreApplication.translate
self.label.setText(_translate("Main-
Window", f"Xmin = {round(max,rou_num)}"))
self.label_3.setText(_translate("Main-
Window", f"Xmax = {round(x,rou_num)}"))
self.label_4.setText(_translate("MainWindow", f"F(Xmin) = {round(-
f(max),rou_num)}"))
self.label_5.setText(_translate("MainWindow", f"F(Xmax) = {round(-
f(x),rou_num)}"))

if __name__ == "__main__":

    import sys
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())

```