

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»



# ОТЧЕТ

по технической (проектно-технической) практике

на 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 |
|  | должность | подпись, дата | имя, отчество, фамилия |

Оценка

дата подпись преподавателя

Ростов-на-Дону 2021



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ 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 ФГБОУ ВО «Донской государственный технический университет»

кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

в период с «08» февраля 2021г. по «05» июня 2021 г.

Обучающийся Костюченко Артём Иванович Обозначение отчета УП.260000.000 Группа ВМО22

Срок представления отчета на кафедру «05» июня 2021 г. Содержание индивидуального задания

Метод Фибоначчи для минимизации функции одной переменной



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»

**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 18.02.2021 |
| 3 | Теоретическое изучение методов оптимизации для нахождения экстремумов функции одной переменной | с 18.02.2021 по 17.03.2021 |
| 4 | Изучение метода Фибоначчи и  областей его применения | 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 |

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 |  |

ОТЗЫВ – ХАРАКТЕРИСТИКА

Обучающийся Костюченко Артём Иванович 2 курса группы ВМО22 кафедра «ПОВТиАС» Вид практики учебная

Наименование места практики ФГБОУ ВО ДГТУ

Обучающийся выполнил задания программы практики.

К выполнению работы подошел ответственно. Все этапы прохождения учебной практики выполнены в срок.

Дополнительно ознакомился/изучил:

Среда разработки Python

Заслуживает оценки

Руководитель практики от кафедры

Т.А. Медведева

« » 20 г.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **СОДЕРЖАНИЕ**  ВВЕДЕНИЕ 7   1. Теоретический обзор 8    1. Необходимые и достаточные условия существования экстремума функции одной переменной 8    2. Методы одновременной минимизации . Преимущества и недостатки метода Фибоначчи 9    3. Леонардо Пизанский (Фибоначчи). Числа Фибоначчи. 10    4. Минимизация метода Фибоначчи 12    5. Выводы по главе 13    6. Постановка задачи 13 2. Алгоритмическое конструирование 14    1. Схема работы алгоритма метода Фибоначчи 14    2. Выводы по главе 15 3. Программное конструирование 16    1. Выбор языка программирования 16    2. Описание программного средства 18    3. Выводы по главе 18 4. Тестирование программного средства 19    1. Результаты работы программы 19    2. Выводы по главе 23   ЗАКЛЮЧЕНИЕ 24  СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 26  ПРИЛОЖЕНИЕ А Исходный код программного средства 27 | | | | | | | | | | |
|  |  |  |  |  | УП.260000.000 | | | | | |
|  |  |  |  |  |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| *Разраб.* | | *Костюченко А.И.* |  |  | *Метод Фибоначчи для минимизации функции одной переменной* | *Лит.* | | | *Лист* | *Листов* |
| *Руководит* | | *Медведева Т.А* |  |  |  |  |  | *6* | *29* |
|  | |  |  |  | *ДГТУ*  *кафедра ПОВТиАС* | | | | |
| *Проверил* | |  |  |  |
| *Утверд.* | |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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. | | | | | | |
|  |  |  |  |  | УП.260000.000 | *Лист* |
|  |  |  |  |  | *7* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1 Теоретический обзор.**  **1.1 Экстремум функции. Необходимые и достаточные условия существования экстремума.**  *Определение*. Точка из области определения функции называется *точкой минимума* этой функции, если найдется такая 𝛿-окрестность точки *,* что для всех из этой окрестности выполняется неравенство:  *Определение*. Точка из области определения функции называется *точкой максимума* этой функции, если найдется такая 𝛿-окрестность точки *,* что для всех из этой окрестности выполняется неравенство:  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 дифференцируема в точке и имеет в этой точке локальный экстремум, то .  *Первое достаточное условие экстремума дифференцируемой функции.*  *Теорема 2*. Пусть точка является точкой возможного экстремума  функции , и пусть функция дифференцируема всюду в некоторой  окрестности точки . Тогда, если в пределах указанной окрестности  производная положительна (отрицательна) слева от точки и  отрицательна (положительна) справа от точки , то функция имеет в  точке локальный максимум (минимум). Если же имеет один и тот же знак слева и справа от точки , то экстремума в точке нет.  *Второе* *достаточное условие экстремума.*  Иногда вызывает затруднение исследование знака первой производной слева и справа от точки возможного экстремума. На этот случай указывается другое достаточное условие наличия экстремума в данной точке возможного экстремума, не требующее исследования знака в окрестности , но предполагающее существование в точке отличной от нуля конечной второй производной . | | | | | | |
|  |  |  |  |  | УП.260000.000 | *Лист* |
|  |  |  |  |  | *8* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Теорема 3. Пусть функция имеет конечную вторую производную в данной точке возможного экстремума. Тогда функция имеет в точке максимум, если .  *Замечание*. Теорема 3 имеет более узкую сферу действия, чем теорема 2.  Так, теорема 3 не решает вопроса об экстремуме для случая, когда вторая производная не существует в точке , а также для случая, когда . 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 Фибоначчи**  Методы оптимизации имеют большое практическое применение, которое заключается в оптимизации некоторой функции . 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 вычисляемых в точках . Это методы так называемого прямого поиска, в котором точки являются пробными.  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. | | | | | | |
|  |  |  |  |  | УП.260000.000 | *Лист* |
|  |  |  |  |  | *9* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1.3 Леонардо Пизанский (Фибоначчи) . 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 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; | | | | | | |
|  |  |  |  |  | УП.260000.000 | *Лист* |
|  |  |  |  |  | *10* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| − 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 задается линейным рекуррентным соотношением:  В первой форму:  Во второй форме: | | | | | | |
|  |  |  |  |  | УП.260000.000 | *Лист* |
|  |  |  |  |  | *11* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1.4 Минимизация метода Фибоначчи**  Требуется найти безусловный минимум функции одной переменной, то есть такую точку , что  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 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 | | | | | | |
|  |  |  |  |  | УП.260000.000 | *Лист* |
|  |  |  |  |  | *12* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Для метода Фибоначчи характеристика относительного начального интервала неопределенности находится по формуле (3)  где N – количество вычислений функции  Замечания:   1. При заданном количестве N вычислений функции метод Фибоначчи обеспечивает минимальную величину конечного интервала неопределенности. 2. Нумерация интервалов неопределенности: 3. На − й итерации длина интервала неопределенности сокращается по правилу .   **1.5 Вывод 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 MATLAB, проверить правильность полученных результатов и визуализировать графики тестируемых функций. | | | | | | |
|  |  |  |  |  | УП.260000.000 | *Лист* |
|  |  |  |  |  | *13* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **2 Алгоритмическое конструирование**  **2.1 Схема работы**  Реализуемый алгоритм для метода Фибоначчи представлен в виде схемы на рисунке 2.  C:\Users\pepega\Downloads\practice (1).png  рисунок 2 | | | | | | |
|  |  |  |  |  | УП.260000.000 | *Лист* |
|  |  |  |  |  | *14* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **2.2 Вывод по главе**  Вышеописанный алгоритм позволяет решить поставленную задачу, и реализовать метод Фибоначчи для минимизации одной переменной за определенное количество шагов с заданной точностью. | | | | | | |
|  |  |  |  |  | УП.260000.000 | *Лист* |
|  |  |  |  |  | *15* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **3 Программное конструирование**   * 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* |
|  |  |  |  |  | *16* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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. | | | | | | |
|  |  |  |  |  | УП.260000.000 | *Лист* |
|  |  |  |  |  | *17* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **3.2 Описание программы**  Программный код написан и оформлен на языке высокого уровня - Python в Visual Studio Code.  Для нахождения минимума и максимума функции, в качестве входных данных, вводятся значения в следующем порядке:  a, b – интервал поиска;  n – количество вычислений целевой функции;  Epsilone – точность поиска;  Все значения имеют тип данных float.  (см. рисунок 3).    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).    рисунок 4 – Пример выполнения программы  **3.3 Вывод по главе**  В данной главе рассмотрена реализованная программа, которая позволяет решить поставленную задачу, найти минимумы и максимумы исследуемых функций. | | | | | | |
|  |  |  |  |  | УП.260000.000 | *Лист* |
|  |  |  |  |  | *18* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **4 Тестирование программы (16)**  **4.1 Результаты работы программного средства(14)**  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: , на промежутке [1,8]. Решение приведено на рисунке 5.    Рисунок 5 – 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 6 – Построение графика функции в среде MathCAD | | | | | | |
|  |  |  |  |  | УП.260000.000 | *Лист* |
|  |  |  |  |  | *19* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рисунок 7 – Решение в среде mathCAD  2. В качестве второго примера рассматривается функция , на промежутке [2, 8]. Пример работы программы Python приведен на рисунке 8.    Рисунок 8 – Полученные результаты программы  Построение графика и решение указаны на рисунках 9 и 10 | | | | | | |
|  |  |  |  |  | УП.260000.000 | *Лист* |
|  |  |  |  |  | *20* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рисунок 9 – Визуализация графика в среде mathCAD    Рисунок 10 – Решение в среде mathCAD | | | | | | |
|  |  |  |  |  | УП.260000.000 | *Лист* |
|  |  |  |  |  | *21* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3. В качестве второго примера рассматривается функция , на промежутке [2, 8]. Пример работы программы Python приведен на рисунке 11.    Рисунок 11 – Полученные результаты программы  Построение графика и решение указаны на рисунках 12 и 13    Рисунок 12 – Визуализация графика в среде mathCAD | | | | | | |
|  |  |  |  |  | УП.260000.000 | *Лист* |
|  |  |  |  |  | *22* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Рисунок 13 – Решение в среде mathCAD  **3.2 Вывод 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* |
|  |  |  |  |  | *23* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **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 С++, которое позволяет решать задачу нахождения экстремумов функции за заданное количество шагов. В результате цели, поставленные в начале работы, были достигнуты, а задачи выполнены. | | | | | | |
|  |  |  |  |  | УП.260000.000 | *Лист* |
|  |  |  |  |  | *24* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **СПИСОК ИСОЛЬЗОВАННЫХ ИСТОЧНИКОВ**  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. Соболь Б.В. Методы оптимизации: практикум / Б.В. Соболь,  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. Воробьёв. 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> | | | | | | |
|  |  |  |  |  | УП.260000.000 | *Лист* |
|  |  |  |  |  | *25* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ПРИЛОЖЕНИЕ А**  **исходный код программного средства**  # Python Version : 3.8  import math  # Математические функции  def f(x):      return (x-2)\*math.cos(x)  def func(x):      return -(x-2)\*math.cos(x)  print(“Исходные данные”)  # Интервал  a = int(input("Введите a : "))  a1 = a  b = int(input("Введите b : "))  b1 = b  # Количество вычислений  N = int(input("Введите количество вычислений целевой функции : "))  N1 = N  Epsilone =  0.0001 # float(input("Введите приближение :"))  print("Введите приближение Epsilone : ",Epsilone)  # Входные данные  Fib = [1,1]  for i in range(2,N):      Fib.append(Fib[i-1]+Fib[i-2])  # print(Fib)  if N % 2 == 0:      sign = 1  else:      sign = -1  x1 = a + (Fib[N-2] \* (b-a) - sign\*Epsilone) / Fib[N-1]  x2 = a + (Fib[N-2] \* (b-a) - sign\*Epsilone) / Fib[N-1]  f1 = f(x1)  f2 = f(x2)  j = 1  # Нахождение минимума  while (j <= (N-1)):      if ((N - j +1)% 2 == 0):          sign = 1 | | | | | | |
|  |  |  |  |  | УП.260000.000 | *Лист* |
|  |  |  |  |  | *26* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| 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]          f1 = f(x1)          x = x2      else:          a = x1          x1 = x2          f1 = f2          x1 = a + (Fib[N - j - 1] \* (b-a) - (sign \* Epsilone))/Fib[N-j]          f2 = f(x2)          x = x1      j += 1  # Перезапись входных данных  Fib1 = [1,1]  for i in range(2,N):      Fib1.append(Fib1[i-1]+Fib1[i-2])  # print(Fib1)  if N1 % 2 == 0:      sign1 = 1  else:      sign1 = -1  x3 = a1 + (Fib1[N1-2] \* (b1-a1) - sign1\*Epsilone) / Fib1[N1-1]  x4 = a1 + (Fib1[N1-1] \* (b1-a1) - sign1\*Epsilone) / Fib1[N1-1]  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 = a + (Fib1[N1 - j1 - 1] \* (b1-a1) - (sign1 \* Epsilone))/Fib1[N1-j1]          f3 = func(x3)          max = x4 | | | | | | |
|  |  |  |  |  | УП.260000.000 | *Лист* |
|  |  |  |  |  | *27* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |
| else:          a1 = x3          x3 = x4          f3 = f4          x4 = a1 + (Fib1[N1 - j1 - 1] \* (b1-a1) - (sign1 \* Epsilone))/Fib1[N1-j1]          f4 = func(x4)          max = x3      j1 += 1  print(f'Результат оптимизации:\n min = {x}\nmax = {max}\nf(min) = {f(x)}\nf(max) = {f(max)}') | | | | | | |
|  |  |  |  |  | УП.260000.000 | *Лист* |
|  |  |  |  |  | *28* |
| *Изм.* | *Лист* | *№ докум.* | *Подпись* | *Дата* |