

И.Г. Куклина

**МЕТОДЫ ОПТИМИЗАЦИИ ЛИНЕЙНОГО
И НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ**

Учебное пособие

Нижний Новгород
2024

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Нижегородский государственный архитектурно-строительный университет»

И.Г. Куклина

МЕТОДЫ ОПТИМИЗАЦИИ ЛИНЕЙНОГО
И НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Утверждено редакционно-издательским советом университета
в качестве учебного пособия

Нижний Новгород
ННГАСУ
2024

УДК 004.94
К 88
ББК 32.973

Рецензенты:

Т.И. Курагина – канд. техн. наук, доцент, заместитель директора по учебно-методической работе ИТС ФГБОУ ВО «Нижегородский государственный технический университет им. Р. Е. Алексеева»

И.В. Ногарева – заместитель директора по учебно-производственной работе ГБПОУ «Нижегородский техникум транспортного обслуживания и сервиса»

Куклина, И.Г. Методы оптимизации линейного и нелинейного программирования : учебное пособие / И.Г. Куклина ; Министерство образования и науки Российской Федерации, Нижегородский государственный архитектурно-строительный университет. – Нижний Новгород : ННГАСУ, 2024. – 101 с. – ISBN 978-5-528-00589-8. – Текст : непосредственный.

В учебном пособии представлено использование современного математического процессора MathCAD для решения задач оптимизации линейного и нелинейного программирования. Подробно изложен учебный материал по теоретическому и лабораторному курсу дисциплины «Методы оптимизации линейного и нелинейного программирования» для магистрантов, обучающихся по направлению подготовки 09.04.02 – «Информационные системы и технологии».

Библиогр.: 16 назв.

ББК 32.973

ISBN 978-5-528-00589-8

© И.Г. Куклина, 2024
© ННГАСУ, 2024

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	5
1. Решение задач математической статистики и теории вероятностей.....	10
1.1. Формула испытаний Бернулли.....	11
1.2. Решение задач теории вероятностей и математической статистики в практике работы информационных систем	13
1.3. Задания для самостоятельной работы	17
1.4. Определение числовых характеристик случайных величин.....	20
1.5. Практические задачи вероятности попадания случайной величины в заданный интервал	22
1.6. Задания для самостоятельной работы	25
2. Локальные максимумы и минимумы. Граничные максимумы и минимумы. Задачи линейного программирования.	28
2.1. Пример выполнения задания	29
2.2. Задания для самостоятельной работы	32
2.3. Матричный вид решения задач линейного программирования	33
2.4. Пример выполнения задания	35
2.5. Задания для самостоятельной работы	36
3. Идентификация стационарных точек функции одной переменной	39
3.1. Максимизация нелинейной унимодальной функции.....	40
3.2. Пример выполнения задания	40
3.3. Пример выполнения задачи в математическом процессоре	41
3.4. Задания для самостоятельной работы	44
4. Математическая модель целочисленной задачи линейного программирования	46
4.1. Выполнение оптимального целочисленного исчисления методом графического решения в процессоре MathCAD	48
4.2. Задания для самостоятельного решения задачи оптимизационного целочисленного исчисления графическими методами	53
5. Решение задач со многими переменными при помощи линейного программирования. Метод Жордана-Гаусса - полное исключение неизвестных	55
5.1. Пример решения задачи	56

5.2. Задания для самостоятельного решения задачи методом полного исключения неизвестных	59
6. Поиск точки оптимума внутри заданного интервала численными методами	61
6.1. Методы исключения	62
6.2. Решение задачи поиска оптимума методом деления интервала пополам.....	63
6.3. Решение задачи с использованием пользовательской функции.....	68
6.4. Задания для самостоятельной работы	70
7. Практические математические модели в задачах оптимизации	72
7.1. Задачи практического применения методов оптимизации в программировании.....	72
7.2. Задания для самостоятельной работы	78
8. Оптимизационные задачи в теории колебаний.....	81
8.1. Оптимизационные задачи динамики. Вертикальные колебания технических систем.....	81
8.2. Расчет дифференциальных уравнений колебательных процессов и получение частотных характеристик.....	88
8.3. Расчет контактных напряжений в контактных парах	94
Заключение.....	98
Библиографический список.....	99

ВВЕДЕНИЕ

Современные специалисты в сфере IT технологий должны обладать отличными знаниями высшей математики, уметь грамотно, быстро и уверенно справляться с задачами дифференциального и интегрального исчисления, решать задачи теории вероятностей и описания случайных процессов. Все указанные научные направления высшей математики всегда были приоритетными и фундаментальными знаниями любого инженера. Построение моделей алгоритмизации на основе математической логики при создании многих проектов современных информационных технологий – первостепенные умения исследователя-магистранта. Создание алгоритмизированных моделей невозможно без выстроенной математической логики. Настоящее учебное пособие предназначено специалистам в сфере информационных технологий для решения инженерных задач автоматизации производства и получения углубленных знаний в области проведения испытаний и анализа статистических данных исследовательских процессов.

Учебная дисциплина "Автоматизированная обработка научной информации" позволяет студентам получить базовые знания в сфере математического моделирования с применением современных математических платформ. Проектирование и создание автоматизированной структурированной системы сложной модели исследовательских задач уже требует углубленных знаний многокритериальных оптимизационных задач. Необходимый пакет сведений линейного и нелинейного программирования, которым должен владеть любой специалист информационных систем в наши дни, теоретически изложен в данном пособии. Практическое решение поставленных задач в курсе прохождения обучения по программе магистратуры по направлению подготовки 09.04.02 – «Информационные системы и технологии», направленности (профиля): «Технология разработки информационных систем» предложено при помощи платформы математических

расчетов MathCAD. Большие возможности и легкость применения математического процессора MathCAD, использование коллективной работы, ориентированность на создание интерактивных документов с великолепным визуальным сопровождением вычислений делает систему автоматизированной обработки для решения задач оптимизации линейного и нелинейного программирования компьютерной алгебры практически незаменимой при проведении научных изысканий. Решение задачи и отображение вычислений в виде общепринятых математических выражений с использованием принципа "What You See Is What You Get – что видишь, то и получаешь" (стандарт де-факто WYSIWYG для современных приложений) открывает безграничные возможности для исследователей, не знакомых с принципами программирования.

Имеющие основополагающие знания в области программирования магистранты направления "Информационные системы и технологии" в практических задачах применяют модели распределенных вычислений и программные модули с возможностью трассируемости инженерных математических решений. Прекрасно выполненный интерфейс и интуитивно понятное математическое описание даже сложных программных комплексов и алгоритмических решений полностью компенсирует отсутствие отладчика.

Следует отметить, что задачи математического программирования подразделяются на задачи линейного и нелинейного программирования. Если рассматриваемые функции являются линейными, то задача решается методами линейного программирования, если же хотя бы одна из исследуемых функций нелинейная, то задача решается по принципам нелинейного программирования. Использование современного математического процессора MathCAD отлично приспособлено для применения эффективных методов решения проблем нелинейного программирования. Решение задач оптимизации предлагает алгоритмы представления моделей линейного

программирования с несколькими целевыми функциями и одномерные и многомерные задачи нелинейного программирования. В учебном пособии обстоятельно изложен учебный материал по теоретическому, лабораторному и практическому курсу дисциплин: «Методы оптимизации линейного и нелинейного программирования» и «Применение методов оптимизации в научных исследованиях» для магистрантов, обучающихся по направлению подготовки 09.04.02 – «Информационные системы и технологии», направленностей (профилей): «Технология разработки информационных систем», «Компьютерная графика и Web-визуализация», «Искусственный интеллект в системах и сетях передачи данных».

Впервые о теории линейного программирования заговорили американские математики Джордж Данциг и Тьяллинг Купманс [1]. Исследования линейных моделей проводил в конце 30-х гг. XX в. ленинградский математик Л.В. Канторович, он же разработал способ решения условно-экстремальных задач близкий к симплекс-методу. Мировое признание роли оптимизационных задач принесла Нобелевская премия, присужденная Т. Купмансу и Л.В. Кантаровичу в 1975 году.

Специалисты и магистры современности должны стремительно демонстрировать наличие математического набора профессиональных знаний, реально представлять практические умения и навыки, позволяющие быстро и точно решать нестандартные задачи математического моделирования. Линейное программирование связано не просто с разновидностью математического аппарата, но и с широким классом объектов реальных процессов.

Стремительное накопление знаний в области автоматизированного физического и математического проектного комплекса, создаваемое банком автоматизированных математических платформ, приводит к необходимости формировать обработчики информационных ресурсов, осуществляющие эффективное визуализированное и понятное пользователю преоб-

разование видов и типов информации. Данными возможностями обладают современные пакеты математических автоматизированных редакторов, направленные на возможность быстро и правильно организовать процесс обработки научных исследований и экспериментов. Передовой технологией современности выступает MatCAD.

Встроенная среда программирования пакета автоматизированных математических решений MathCAD – методология программной инженерии создания ведущего программного обеспечения, предоставляющая возможность разработки уникальных программных продуктов. Грамотное применение редактора математических исследований в решении задач линейной и нелинейной оптимизации незаменимо в производственном процессе создания новых автоматизированных структур, так как позволяет сформировать научного работника, приспособленного к информационно-аналитической работе с применением сложного математического аппарата.

Теоретические сведения в пособии представлены в кратком изложении и предназначены для подготовки к выполнению практического задания. Наполненность теории позволяет воздержаться от необходимости обращения к справочным материалам по рассматриваемой тематике. Показаны основные правила перехода от прямых линейных оптимизационных задач к двойственным задачам. Рассматривается применение встроенных функций MathCAD для проведения вычислений при выполнении оптимизационных решений с унимодальными функциями. Многомерные задачи нелинейного программирования с поиском экстремума, аналитический метод поиска оптимума функции многих переменных по принципу применения критерия Сильвестра, метаэвристические методы оптимизации включены в пособие.

В настоящем учебно-методическом пособии рассмотрены основные теоретические обоснования математических методов и практических решений использования инструментальных средств для выполнения задач

математического программирования при помощи пакета математических расчетов MathCAD PRO. Однокритериальная и многоцелевая постановки позволяют решить задачи безусловной и условной линейной и нелинейной оптимизации. Применяются метаэвристические алгоритмы стохастического поиска глобальных решений и нечеткой оптимизации. Для лучшего понимания и усвоения материала предложены комплекты заданий для самостоятельной работы, рассматриваемые методы применяют современные компьютерные реализации.

Проведенные вычисления и примеры оформления документированного отчета представлены в качестве скриншотов в учебном пособии. Отчетностью по выполнению предложенных заданий (выполненных в виде вариантов) должны служить файлы магистрантов, прикрепленные к системе Moodle. Выполненные задания и оформленный отчетный материал по предмету, итоговым результатом которого является экзамен, в значительной степени способствуют формированию знаний будущего магистра направления 09.04.02 "Информационные системы и технологии". Материал, изложенный в пособии, может быть также использован аспирантами и специалистами соответствующих профилей в своей научно-исследовательской работе.

1. Решение задач математической статистики и теории вероятностей

Решение задач теории вероятностей и математической статистики - наука, начало которой положено в средних веках - как попытка подведения теории под статистику азартных игр (кости, рулетка). Эмпирические факты и свойства реальных событий подвели основу под основные понятия теории вероятностей. Изучая азартные игры и прогноз выигрыша, первые вероятностные закономерности открыли основатель математического анализа Блез Паскаль, итальянский математик Джероламо Кардано, один из основателей аналитической геометрии Пьер Ферма. Исследуя далее вопросы статистики, возникающие при бросании костей Христиан Гюйгенс ввел основные понятия: величину шанса, математическое ожидание, дискретные случаи. Голландский механик, математик и астроном, Христиан Гюйгенс ван Зёйлихем первым сформулировал теоремы сложения и умножения вероятностей в 1657 году, на двадцать лет раньше Блеза Паскаля и Пьера Ферма.

Положил начало вариационному исчислению швейцарский математик Якоб Бернулли. Член Парижской академии наук и Берлинской академии наук Бернулли доказал частный случай Закона больших чисел. Страстный любитель азартных игр Джероламо Кардано положил начало формулировке закона больших чисел. Он показал, что точность эмпирической статистики растет при увеличении количества испытаний. Якоб Бернулли вывел правила подсчета вероятности сложных событий и доказал, что в серии испытаний частота событий не является хаотичной, а стремится к предельному вероятностному теоретическому значению. Позже французский математик Семён Дени Пуассон обосновал более общую форму закона больших чисел.

1.1. Формула испытаний Бернулли

Допустим, производится один и тот же опыт n раз при неизменных условиях. Случайное событие A может произойти в каждом из этих опытов с вероятностью p и не произойти с вероятностью $q = 1 - p$. Вероятность, что событие A произойдет в этих n испытаниях, равно k раз $0 \leq k \leq n$ и находится по формуле Бернулли:

$$P_n(k) = C_n^k p^k (1 - p)^{n-k} \quad (1.1)$$

где C_n^k - число сочетаний равное

$$C_n^k = \frac{n!}{(n-k)! k!} \quad (1.2)$$

$n! = 1 \cdot 2 \cdot 3 \cdots n$ - факториал числа n .

Основная полезность формулы Бернулли в том, что она позволяет определить, когда из-за слишком большого количества элементарных исходов простейшие комбинационные способы расчета вариантов неприменимы.

Некоторые вероятностные расчеты, используемые в теории вероятностей, когда применяется формула Бернулли:

1. Вероятность того, что событие A произойдет не более, чем k раз.

$Prob(k)$ обозначение вероятности того, что событие A произойдет не более, чем k раз.

Применяя теорему сложения вероятностей несовместимых событий, получаем выражение:

$$Prob(k) = P_n(0) + P_n(1) + \cdots + P_n(k) = \sum_{i=0}^k P_n(i) \quad (1.3)$$

2. Вероятность того, что событие A произойдет не менее, чем k раз.

Данная вероятность $R_n(k)$ будет равна сумме вероятностей:

$$R_n(k) = P_n(k) + P_n(k + 1) + \dots + P_n(n) = \sum_{j=k}^n P_n(j) \quad (1.4)$$

Беря в расчет, что

$$\sum_{i=0}^n P_n(i) = P_n(0) + P_n(1) + \dots + P_n(k) + \dots + P_n(n) = 1 \quad (1.5)$$

получается

$$R_n(k) = 1 - Prob(k - 1). \quad (1.6)$$

3. Вероятность того, что событие A произойдет хотя бы один раз. Это означает, что событие A произойдет не менее одного раза, событие может произойти один, два и т.д. до n раз.

Данная вероятность события $R_n(k)$ будет равна:

$$R_n(1) = P_n(1) + P_n(2) + \dots + P_n(n) = \sum_{j=1}^n P_n(j) = 1 - P_n(0), \quad (1.7)$$

4. Вероятность того, что событие A произойдет не более одного раза.

Эта вероятность события равна:

$$Prob(1) = P_n(0) + P_n(1) = [1 + (n - 1)p](1 - p)^{n-1}. \quad (1.8)$$

Для работы с формулами Бернулли математический пакет MathCAD оснащен следующими функциями:

– $combin(n, k)$ - вычисляет число сочетаний C_n^k по формуле (1.2);

- $dbinom(k, n, p)$ - вычисляет по формуле Бернулли вероятность $P_n(l)$ по формуле (1.1);
- $pbinom(k, n, p)$ - вычисляет по формуле Бернулли вероятность $Prob(k)$ по формуле (1.3).

1.2. Решение задач теории вероятностей и математической статистики в практике работы информационных систем

Задача 1

С целью проверки качества изготовленных мониторов была сформирована контрольная группа – десять мониторов, из которых наугад выполнялась выборка случайных мониторов с возвратом взятого монитора в контрольную группу после проведения проверки качества изображения (данная схема имеет название выборки с возвратом). Величина бракованных мониторов во всей созданной партии составила 0,05. Необходимо определить, каковы вероятности найти в контрольной группе:

- *событие A* - бракованные мониторы;
- *событие B* - не более двух кондиционных мониторов;
- *событие C* - не менее двух бракованных мониторов.

Решение.

Условие задачи определяет: $n = 10$, $k = 2$, $p = 0,05$, $q = 1 - p = 0,95$. Вероятность *события A* равна вероятности того, что в контрольной группе будет обнаружен хотя бы один бракованный монитор, данная вероятность определится по формуле (1.7):

$$P(A) = R_{10}(1) = \sum_{j=1}^{10} P_n(j). \quad (1.9)$$

Вероятность *события B* будет найдена по формуле (1.3):

$$Prob(2) = \sum_{j=0}^2 P_n(j). \quad (1.10)$$

Вероятность *события C* будет найдена по формуле (1.4):

$$R_{10}(2) = \sum_{j=2}^{10} P_n(j) = 1 - Prob(1). \quad (1.11)$$

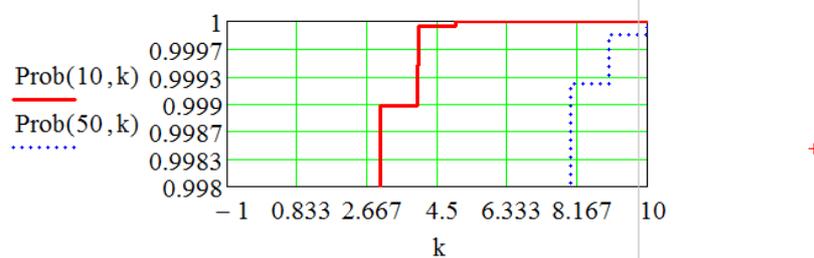
Расчет вероятностей событий *A, B, C* с применением функций пользователя $Prob(n, k)$ и $R(n, k)$ в математическом процессоре MathCAD показан на рисунке 1.

Приведенные функции MathCAD позволяют выполнить исследование влияния исходных данных на результаты испытаний по алгоритму Бернулли. Как пример из множества вариантов такого исследования следует привести результаты анализа влияния числа испытаний $n = 10$ и $n = 50$ на вероятность $P_n(k)$.

После проведенного решения примера (рисунок 1) можно сделать выводы, что:

- *событие A* будет выполнено - в партии окажется хотя бы один бракованный монитор;
- *событие B* выполняется с высокой долей вероятности;
- *событие C* является маловероятным.

$n := 10$	$p := 0.05$	
$C(n, m) := \text{combin}(n, m)$		Число сочетаний
$P(n, j) := \text{dbinom}(j, n, 0.05)$		Вероятность по формуле Бернулли
$\text{Prob}(n, k) := \text{pbinom}(k, n, 0.05)$		$R(n, k) := 1 - \text{Prob}(n, k - 1)$
Вероятность события А		$R(10, 0) = 1$
Вероятность события В		$\text{Prob}(10, 2) = 0.988$
Вероятность события С		$R(10, 2) = 0.086$



$n := 10$	$p := 0.05$	
$C(n, m) := \text{combin}(n, m)$		Число сочетаний
$P(n, j) := \text{dbinom}(j, n, 0.05)$		Вероятность по формуле Бернулли
$\text{Prob}(n, k) := \text{pbinom}(k, n, 0.05)$		$R(n, k) := 1 - \text{Prob}(n, k - 1)$
Вероятность события А		$R(10, 0) = 1$
Вероятность события В		$\text{Prob}(10, 2) = 0.988$
Вероятность события С		$R(10, 2) = 0.086$

Рисунок 1.1. Расчет вероятностей событий примера 1

График рисунка 1.1 при решении задачи демонстрирует смещение графика комбинаторики при увеличении числа исследуемых объектов.

Задача 2

Вероятность выхода из строя одного монитора из ста при одновременном запуске равна 0,1. Определить наименьшее число запусков мониторов N_{min} , которое надо произвести, чтобы с вероятностью 0,8 произошел выход из строя одного из испытуемых мониторов.

Решение.

Вероятность события A заключается в выходе из строя одного монитора при одновременном запуске испытуемых ста мониторов, и вероятность этого события равна $p = 0,1$. Вероятность того, что при запуске произойдет выход из строя хотя бы одного монитора определяется по формуле (1.12):

$$1 - P_n(0) = 1 - (1 - p)^n = 1 - q^n, \quad (1.12)$$

Причем, $q = 1 - p$. Число запусков комплекта мониторов N_{min} можно найти из неравенства (13):

$$1 - q^{N_{min}} \geq 0.8 \quad (1.13)$$

Решение предложенной задачи отображено на рисунке 1.2.

$n := 0$	Given
$1 - 0.9^n \geq 0.8$	Решаемое неравенство
$V := \text{Find}(n) = 15.276$	
$V_{min} := \text{ceil}(V) = 16$	

Рисунок 1.2. Решение задачи примера 2

Встроенная в MathCAD функция *ceil* производит округление полученного результата неравенства до ближайшего большего целого и выдает результат решения задачи в виде $N_{min} = 16$. Следовательно, необходимо

выполнить 16 общих включений мониторов для того, чтобы вышел из строя один монитор.

1.3. Задания для самостоятельной работы

Задача 1

Исследуется работа центрального обрабатывающего устройства (ЦП) при одной интенсивности работы персонального компьютера. Выборка центральных процессоров осуществлялась из контрольной группы случайным образом. С целью проверки качества изготовленных центральных процессоров были проведены тестовые испытания. Величина центральный процессоров, показавших неудовлетворительные характеристики, составила 0,009. Необходимо определить, каковы вероятности найти в контрольной группе:

- *событие A* - бракованные центральные процессоры;
- *событие B* - не более пяти кондиционных центральных процессоров;
- *событие C* - не менее трех бракованных процессоров.

Задача 2

Проблемы с аккумулятором мобильного компьютера (ноутбука) – наиболее частая неисправность. Без аккумулятора ноутбук не просто теряет свои мобильные свойства автономности, но и может произойти потеря данных при внезапном отключении мобильного компьютера. Рассматривается работа аккумулятора при одной интенсивности работы ноутбуков. Выборка ноутбуков осуществлялась из контрольной группы случайным образом. С целью проверки качества изготовленных аккумуляторов были проведены тестовые испытания. Количество аккумуляторов, показавших неудовлетворительные характеристики, составило 0,06. Необходимо определить, каковы вероятности найти в контрольной группе:

- *событие A* - бракованные аккумуляторы;
- *событие B* - не более двух кондиционных аккумуляторов;
- *событие C* - не менее одного бракованного аккумулятора.

Задача 3

Для выполнения качественной печати рассматривается работа цветных принтеров. Выборка цветных принтеров осуществляется из контрольной группы случайным образом. С целью проверки скорости печати принтеров и основных характеристик цветопередачи при выполнении печатающим устройством рабочих операций были выполнены тестовые испытания. Величина принтеров, показавших неудовлетворительные характеристики, составила 0,07. Необходимо определить, каковы вероятности найти в контрольной группе:

- *событие A* - бракованные принтеры;
- *событие B* - не менее трех кондиционных принтеров;
- *событие C* - не более двух бракованных принтеров.

Задача 4

Основная неисправность протяжного сканера – замятие бумаги. В структурах организации при работе с важными документами должны быть использованы надежные аппараты сканирования. Вероятность выхода из строя одного сканера из ста (замятие хотя бы одного листа бумаги) при одновременном запуске равна 0,6. Определить наименьшее число запусков ста сканеров N_{min} , которое необходимо произвести, чтобы с вероятностью 0,9 произошло замятие листа хотя бы в одном из сканеров.

Задача 5

Вероятность выхода из строя одного центрального процессора при перегреве из ста при одновременном перегреве всех испытуемых центральных процессоров равна 0,4. Определить наименьшее число возможных перегревов N_{min} , которое должно произойти, чтобы с вероятностью 0,9 произошел выход из строя одного из испытуемых центральных процессоров.

Задача 6

Мультимедийный проектор - достаточно сложное устройство, к тому же часто ремонт его связан с подъемом ремонтника к потолку. Неисправности проекторов чаще типовые, чтобы их выявить – проводятся испытания. Вероятность выхода из строя одного мультимедийного проектора из ста при одновременном запуске равна 0,4. Определить наименьшее число запусков мультимедийных проекторов N_{min} , которое надо произвести, чтобы с вероятностью 1 произошел выход из строя одного из испытуемых мультимедийных проекторов.

1.4. Определение числовых характеристик случайных величин

Случайной является величина, которая в результате испытаний примет одно значение априори – еще до опыта. Это значение неизвестно и зависит от определенной совокупности случайных величин.

Дискретная случайная величина может принимать множество значений x_1, x_2, \dots, x_n и вероятности появления этих значений p_1, p_2, \dots, p_n .

Непрерывная случайная величина случайным образом может принимать любое значение на всей вещественной оси $(-\infty, \infty)$ или на конечном отрезке $[a, b]$.

Основные законы, представляющие непрерывные случайные величины:

- функция распределения указывает, что случайная величина X менее заданного значения x ;

$$F(x) = P(X < x), \quad (1.14)$$

- плотность распределения

$$f(x) = \frac{d}{dx} F(x) \quad (1.15)$$

Принимается, что $F(x)$ – непрерывная функция, для которой справедливы следующие свойства:

$$0 \leq F(x) \leq 1 \quad (1.16)$$

$$F(-\infty) = \lim_{x \rightarrow -\infty} F(x) = 0 \quad (1.7)$$

$$F(\infty) = \lim_{x \rightarrow \infty} F(x) = 1 \quad (1.18)$$

$$F(b) \geq F(a), \text{ если } b > a \quad (1.19)$$

Свойство формулы 1.19 указывает на то, что $F(x)$ является функцией неубывающей.

$F(x)$ можно вычислить, зная плотность распределения $f(x)$:

$$F(x) = \int_{-\infty}^{\infty} p(z) dz \quad (1.20)$$

$f(x)$ является функцией неотрицательной:

$$f(x) \geq 0 \quad (1.21)$$

и должна удовлетворять условию:

$$\int_{-\infty}^{\infty} f(x) dx = 1 \quad (1.22)$$

Для непрерывных случайных величин с непрерывной функцией $F(x)$ справедливы неравенства:

$$P(\alpha < X < \beta) = P(\alpha \leq X < \beta) = P(\alpha < X \leq \beta) = P(\alpha \leq X \leq \beta) \quad (1.23)$$

Причем:

$$P(\alpha < X < \beta) - \text{вероятность попадания в открытый интервал} \\ (\alpha, \beta); \quad (1.24)$$

$$P(\alpha \leq X < \beta) - \text{вероятность попадания в полуоткрытый интервал} \\ [\alpha, \beta); \quad (1.25)$$

$$P(\alpha < X \leq \beta) - \text{вероятность попадания в полуоткрытый интервал} \\ (\alpha, \beta]; \quad (1.26)$$

$$P(\alpha \leq X \leq \beta) - \text{вероятность попадания в закрытый интервал} \\ [\alpha, \beta]; \quad (1.27)$$

Вычисление вероятности попадания в полуоткрытый интервал $[\alpha, \beta)$ возможно двумя методами:

$$P(\alpha \leq X < \beta) = F(\beta) - F(\alpha) \quad (1.28)$$

$$P(\alpha \leq X < \beta) = \int_{\alpha}^{\beta} f(x) dx \quad (1.29)$$

В соответствии с равенствами 1.23 соотношения 1.28 и 1.29 определяют вероятность попадания случайной величины в отрезки с другими типами границ.

1.5. Практические задачи вероятности попадания случайной величины в заданный интервал

Задача 1

Функция $F(x)$ задается выражением:

$$F(x) = \begin{cases} 0, & \text{если } x < 0; \\ \frac{x^2}{16}, & \text{если } 0 \leq x < 2; \\ x - \frac{7}{4}, & \text{если } 2 \leq x < \frac{11}{4}; \\ 1, & \text{если } x \geq \frac{11}{4}. \end{cases} \quad (1.30)$$

Необходимо определить плотность $f(x)$ и вероятность попадания случайной величины X в интервал $(1, 2.5)$.

Решение.

Программный модуль решения задачи в математическом пакете MathCAD продемонстрирован на рисунке 1.3. Построены график $F(x)$ и

кривая $f(x)$. Вероятность попадания решения в искомый отрезок $(1, 2.5)$ вычислена двумя способами:

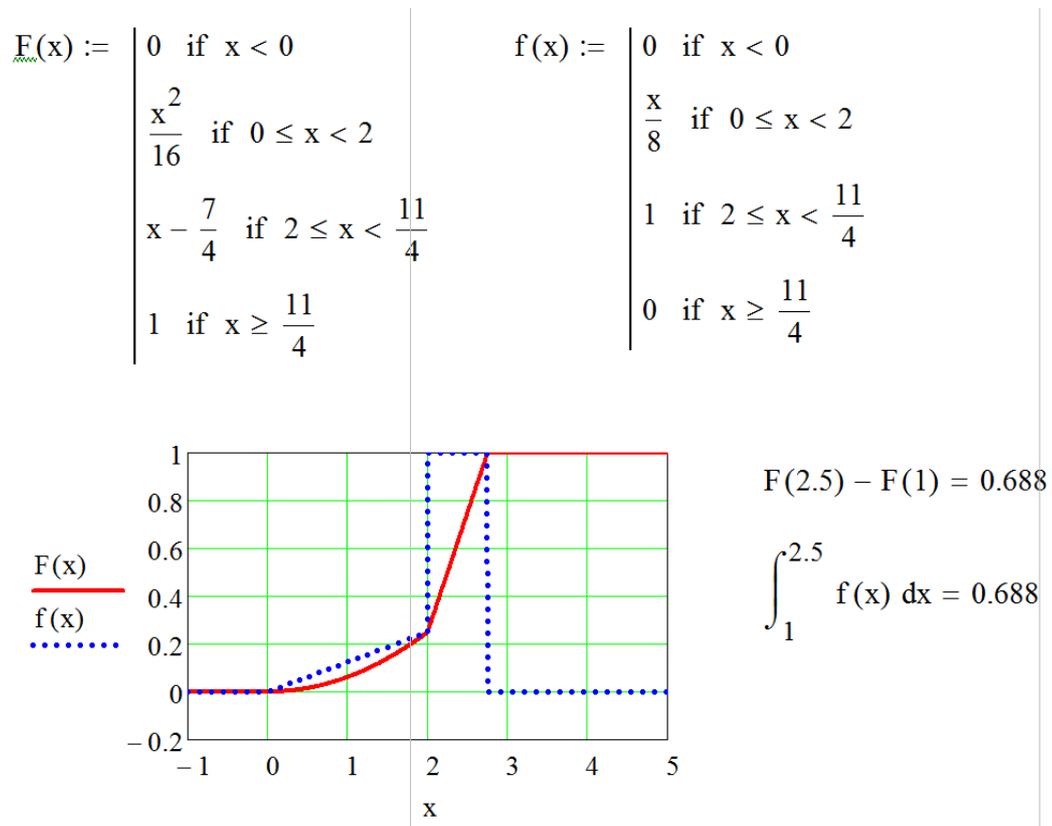


Рисунок 1.3. Решение задачи получения результата вычисления и попадания результата в заданный интервал

Задача 2

Системой уравнений определена плотность распределения случай-ной величины:

$$f(x) = \begin{cases} 0, & x < 2 \\ \frac{A}{x^2}, & x \geq 2 \end{cases} \quad (1.31)$$

Необходимо определить коэффициент A функцию распределения случайной величины $F(x)$, вероятность попадания случайной величины в интервал $(2, 3)$, вероятность того, что при четырех независимых испытаниях случайная величина ни разу не попадет в интервал $(2, 3)$.

Решение.

Программный модуль решения задачи в математическом пакете MathCAD продемонстрирован на рисунке 1.4. Незвестный множитель определен из 1.22:

$$\int_2^{\infty} \frac{A}{x^2} dx = A \int_2^{\infty} \frac{dx}{x^2} = 1 \quad (1.32)$$

$\int_2^{\infty} \frac{A}{x^2} dx \rightarrow \frac{A}{2}$	$\frac{A}{2} = 1 \quad A = 2$
$\int_2^{x \cdot x} \frac{2}{z^2} dz \rightarrow 1 - \frac{2}{x^2}$	$F(x) := \frac{x-2}{x}$
$p := F(3) - F(2) = 0.333$	
$q := 1 - p = 0.667$	
$q^4 = 0.198$	

Рисунок 1.4. Нахождение коэффициента A

Система уравнений:

$$f(x) = \begin{cases} 0, & x < 2 \\ \frac{A}{x^2}, & x \geq 2 \end{cases} \quad (1.33)$$

Принимает вид:

$$f(x) = \begin{cases} 0, & x < 2 \\ \frac{2}{x^2}, & x \geq 2 \end{cases} \quad (1.34)$$

Так же в программе MathCAD вычислена в символьном виде $F(x)$ на интервале $[2, \infty)$, следовательно:

$$F(x) = \begin{cases} 0, & x < 2 \\ \frac{x-2}{x}, & x \geq 2 \end{cases} \quad (1.35)$$

Вероятность попадания случайной величины в интервал $(2, 3)$ получается равной $p = 0,033$.

А вероятность того, что при четырех независимых испытаниях случайная величина ни разу не попадет в интервал $(2, 3)$:

$$(1 - p)^4 = q^4 = 0.667^4 = 0.198 \quad (1.36)$$

1.6. Задания для самостоятельной работы

Задача 1

Функция $F(x)$ задается выражением:

$$F(x) = \begin{cases} 1, & \text{если } x < 0; \\ x^3, & \text{если } 0 \leq x < 2; \\ x + 4, & \text{если } 2 \leq x < \frac{13}{4}; \\ 1, & \text{если } x \geq \frac{13}{4}. \end{cases}$$

Определить плотность $f(x)$ и вероятность попадания случайной величины X в интервал $(0, 1)$.

Задача 2

Системой уравнений определена плотность распределения случайной величины:

$$f(x) = \begin{cases} 2, & x < 4 \\ \sqrt{x} \\ \frac{1}{A}, & x \geq 4 \end{cases}$$

Определить коэффициент A , функцию распределения случайной величины $F(x)$, вероятность попадания случайной величины в интервал $(4, 5)$, вероятность того, что при четырех независимых испытаниях случайная величина ни разу не попадет в интервал $(4, 5)$.

Задача 3

Функция $F(x)$ задается выражением:

$$F(x) = \begin{cases} 0, & \text{если } x < -2; \\ \frac{x^2}{2}, & \text{если } -2 \leq x < 1; \\ x - 1, & \text{если } 1 \leq x < 4; \\ 1, & \text{если } x \geq 4. \end{cases}$$

Определить плотность $f(x)$ и вероятность попадания случайной величины X в интервал $(-1, 0)$.

Задача 4

Системой уравнений определена плотность распределения случайной величины:

$$f(x) = \begin{cases} 1, & x < 2 \\ x^2 * A, & x \geq 2 \end{cases}$$

Определить коэффициент A функцию распределения случайной величины $F(x)$, вероятность попадания случайной величины в интервал $(1, 2)$, вероятность того, что при четырех независимых испытаниях случайная величина ни разу не попадет в интервал $(1, 2)$.

Задача 5

Функция $F(x)$ задается выражением:

$$F(x) = \begin{cases} 2, & \text{если } x < -1,5; \\ 3 + x, & \text{если } -1,5 \leq x < 2; \\ x^2 - 3, & \text{если } 2 \leq x < 3; \\ 5, & \text{если } x \geq 3. \end{cases}$$

Определить плотность $f(x)$ и вероятность попадания случайной величины X в интервал $(-1, 0)$.

Задача 6

Системой уравнений определена плотность распределения случайной величины:

$$f(x) = \begin{cases} 0.85, & x < 1 \\ A + \frac{x}{A}, & x \geq 1 \end{cases}$$

Определить коэффициент A функцию распределения случайной величины $F(x)$, вероятность попадания случайной величины в интервал $(1, 2)$, вероятность того, что при четырех независимых испытаниях случайная величина ни разу не попадет в интервал $(0.5, 1)$.

2. Локальные максимумы и минимумы. Граничные максимумы и минимумы.

Задачи линейного программирования

Математический процессор MathCAD справляется с задачами линейного программирования при определении максимума и минимума функции, находит граничные значения, выполняя упрощения при выполнении задач линейного программирования.

Функции системы математических расчетов MathCAD, решающие проблемы определения экстремумов:

$\text{Maximize}(f, \text{var1}, \text{var2})$ – позволяет вернуть значения $\text{var1}, \text{var2}, \dots$ которые обеспечивают функции f максимальный параметр. Для использования этой функции сначала задается начальное приближение у каждой неизвестной и, в том случае, когда ограничения заданы, необходимо вначале ввести ключевое слово `Given`.

$\text{Minimize}(f, \text{var1}, \text{var2})$ – позволяет вернуть значения $\text{var1}, \text{var2}, \dots$ которые обеспечивают функции f минимальный параметр. Для использования этой функции сначала задается начальное приближение у каждой неизвестной и, в том случае, когда ограничения заданы, необходимо вначале ввести ключевое слово `Given`.

Когда решается задача линейного программирования при поиске экстремума, сначала задается целевая определяющая функция, затем рассматриваются ограничения:

$$f(x) = 108 \cdot x_1 + 112 \cdot x_2 + 126 \cdot x_3 \rightarrow \max \quad (2.1)$$

$$\begin{cases} 0,8 \cdot x_1 + 0,5 \cdot x_2 + 0,6 \cdot x_3 \leq 800 \\ 0,4 \cdot x_1 + 0,4 \cdot x_2 + 0,9 \cdot x_3 \leq 600 \\ 0,1 \cdot x_2 + 0,1 \cdot x_3 \leq 120 \\ x_j \geq 0 \end{cases}$$

При выполнении задания поиска экстремума при помощи функции Maximize (f, var1, var2):

- записывается функция цели, которую необходимо максимизировать;
- происходит присвоение первоначальных значений $x_1 = x_2 = x_3 = 1$;
- далее записывается система уравнений ограничений с условием неотрицательности;
- вводится ключевое слово Given (Дано);
- с использованием функции Maximize (f, var1, var2) находится максимум целевой функции;
- выполняется нахождение точек экстремума целевой функции;
- определяется численное значение в точки экстремума целевой функции.

2.1. Пример выполнения задания

Применяется функция системы Maximize (f, var1, var2) математических расчетов MathCAD, вычисляется экстремум – возвращаются значения var1, var2, ... которые обеспечивают функции f максимальный параметр.

$$f(x) = 108 \cdot x_1 + 112 \cdot x_2 + 126 \cdot x_3 \rightarrow \max \quad (2.2)$$

В системе MathCAD записывается функция цели, которую необходимо максимизировать

$$f(x_1, x_2, x_3) := 108x_1 + 112x_2 + 12x_3$$

Заполняются начальные значения, заданные примером.

$$\begin{cases} 0,8 \cdot x_1 + 0,5 \cdot x_2 + 0,6 \cdot x_3 \leq 800 \\ 0,4 \cdot x_1 + 0,4 \cdot x_2 + 0,9 \cdot x_3 \leq 600 \\ 0,1 \cdot x_2 + 0,1 \cdot x_3 \leq 120 \\ x_j \geq 0 \end{cases} \quad (2.3)$$

В системе MathCAD вводятся начальные значения (рисунок 2.1), система ограничений и набирается ключевое слово Given.

$x1 := 1$	$x2 := 1$	$x3 := 1$
Given		
$x1 \geq 0$	$x2 \geq 0$	$x3 \geq 0$
$0.8x1 + 0.5x2 + 0.6x3 \leq 800$		
$0.4x1 + 0.4x2 + 0.9x3 \leq 600$		
$0.1x2 + 0.1x3 \leq 120$		

Рисунок 2.1. Ввод исходных данных задачи

Затем осуществляется нахождение максимального значения функции цели – рисунок 2.2. Ответ при решении нахождения оптимального значения будет выведен в виде матрицы результатов. Подставив полученные численные значения в уравнение функции, можно увидеть результат, при котором обозначается оптимальное решение функции цели.

$$x := \text{Maximize}(f, x1, x2, x3)$$

$$x = \begin{pmatrix} 250 \\ 1.2 \times 10^3 \\ 0 \end{pmatrix}$$

$$f(250, 1.2 \cdot 10^3, 0) = 1.614 \times 10^5$$

Рисунок 2.2. Вычисление оптимального значения функции

Общий вид документа решения задачи представлен на рисунке 2.3.

$$f(x1, x2, x3) := 108x1 + 112x2 + 12x3$$

$$x1 := 1 \quad x2 := 1 \quad x3 := 1$$

Given

$$x1 \geq 0 \quad x2 \geq 0 \quad x3 \geq 0$$

$$0.8x1 + 0.5x2 + 0.6x3 \leq 800$$

$$0.4x1 + 0.4x2 + 0.9x3 \leq 600$$

$$0.1x2 + 0.1x3 \leq 120$$

$$x := \text{Maximize}(f, x1, x2, x3)$$

$$x = \begin{pmatrix} 250 \\ 1.2 \times 10^3 \\ 0 \end{pmatrix}$$

$$f(250, 1.2 \cdot 10^3, 0) = 1.614 \times 10^5$$

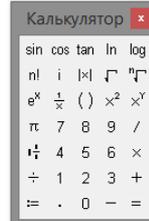
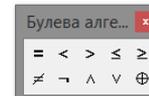
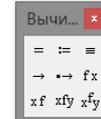


Рисунок 2.3. Общий вид документа решения задачи

2.2. Задания для самостоятельной работы

Задача 1

Найти, используя встроенные функции процессора MathCAD, экстремумы уравнений способом линейного программирования:

$$\begin{cases} x_1 + 2 \cdot x_2 \leq 14 \\ -5 \cdot x_1 + 3 \cdot x_2 \leq 15 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$
$$f(x) = x_1 + x_2 \rightarrow \max(\min)$$

Задача 2

Найти, используя встроенные функции процессора MathCAD, экстремумы уравнений способом линейного программирования:

$$\begin{cases} 2 \cdot x_1 + x_2 \leq 6 \\ x_1 + 3 \cdot x_2 \leq 9 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$
$$f(x) = x_1 + 4 \cdot x_2 \rightarrow \max(\min)$$

Задача 3

Найти, используя встроенные функции процессора MathCAD, экстремумы уравнений способом линейного программирования:

$$\begin{cases} -4 \cdot x_1 + 2 \cdot x_2 \leq 4 \\ x_1 + x_2 \leq 6 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$
$$f(x) = 2 \cdot x_1 + 3 \cdot x_2 \rightarrow \max(\min)$$

Задача 4

Найти, используя встроенные функции процессора MathCAD, экстремумы уравнений способом линейного программирования:

$$\begin{cases} -4 \cdot x_1 + 2 \cdot x_2 \leq 4 \\ x_1 + x_2 \leq 6 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$
$$f(x) = 2 \cdot x_1 + 3 \cdot x_2 \rightarrow \max(\min)$$

Задача 5

Найти, используя встроенные функции процессора MathCAD, экстремумы уравнений способом линейного программирования:

$$\begin{cases} x_1 + x_2 \leq 10 \\ -2 \cdot x_1 + x_2 \leq 2 \\ x_1 - 2 \cdot x_2 \leq 8 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$
$$f(x) = 2 \cdot x_1 + x_2 \rightarrow \max(\min)$$

Задача 6

Найти, используя встроенные функции процессора MathCAD, экстремумы уравнений способом линейного программирования:

$$\begin{cases} x_1 + x_2 \leq 20 \\ -x_1 + x_2 \leq 15 \\ x_1 - 3 \cdot x_2 \leq 9 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$
$$f(x) = 6 \cdot x_1 + x_2 \rightarrow \max(\min)$$

2.3. Матричный вид решения задач линейного программирования

При выполнении задач оптимизации представление условия задачи предпочтительнее реализовать в виде матрицы. Матричное решение задач линейного программирования хорошо представляется в системе математического процессора MathCAD. Автоматизированный редактор алгебраических вычислений позволяет быстро выполнять решение, а прекрасный интерфейс представления данных в MathCAD организует процесс расчета визуально понятным математику, далекому от программирования.

Maximize (f, var1, var2) и Minimize (f, var1, var2) - основные функции поиска экстремальных значений.

Maximize (f, var1, var2) – используется при поиске максимального значения f, возвращая значения var1, var2. Использование данной функции

и ключевого слова `Given` требует указания начального приближения для каждой неизвестной системы уравнений и указания ограничений.

`Minimize (f, var1, var2)` – используется при поиске максимального значения f , возвращая значения $var1, var2$. Использование данной функции и ключевого слова `Given` требует указания начального приближения для каждой неизвестной системы уравнений и указания ограничений.

Нахождение экстремального значения функции в задаче линейного программирования:

$$f(x) = 18 \cdot x_1 + 12 \cdot x_2 + 16 \cdot x_3 \rightarrow \max \quad (2.4)$$

$$\begin{cases} 0.6 \cdot x_1 + 0.4 \cdot x_2 + 0.6 \cdot x_3 \leq 150 \\ 0.3 \cdot x_2 + 0.9 \cdot x_3 \leq 20 \\ -0.2 \cdot x_1 + 0.1 \cdot x_2 + 0.1 \cdot x_3 \leq 120 \\ x_i \geq 0 \end{cases}$$

осуществляется в следующем порядке:

Нахождение экстремума с помощью функции

`Maximize (f, var1, var2)`

- указывается функция цели, подлежащая максимизации;
- присваивается начальное значение переменным $x_1 = x_2 = x_3 = 1$;
- обозначается система ограничений и заносятся условия неотрицательности;
- ключевое слово `Given` запускает решение задачи;
- для поиска максимума целевой функции – используется функция `Maximize (f, var1, var2)`;
- находятся точки экстремума функции;

- по точкам рассчитывается значение функции.

2.4. Пример выполнения задания

Система математических расчетов MathCAD хорошо справляется с нахождением экстремума при использовании матричной формы записи.

$$f(x) = 18 \cdot x_1 + 12 \cdot x_2 + 16 \cdot x_3 \rightarrow \max \quad (2.5)$$

$$\begin{cases} 0,6 \cdot x_1 + 0,4 \cdot x_2 + 0,6 \cdot x_3 \leq 150 \\ x_1 + 0,3 \cdot x_2 + 0,9 \cdot x_3 \leq 20 \\ -0,2 \cdot x_1 + 0,1 \cdot x_2 + 0,1 \cdot x_3 \leq 120 \\ x_j \geq 0 \quad i = 1, 2, 3 \end{cases}$$

В системе MathCAD вводятся начальные значения числа ограничений и числа неизвестных переменных, задается исходная переменная. Затем набирается столбец коэффициентов функции цели в виде вектора размерностью $n \cdot 1$ и матрица размерностью $m \cdot n$ системы ограничений задачи.

Вектор b представляет собой столбец размерностью $m \cdot 1$ правой части системы ограничений. Вводится целевая функция $f(x)$, элементы которой представляются в матричной форме. И происходит обозначение первоначального значения задаваемых переменных. Появляется вектор первых значений x .

$$\begin{array}{l} \underline{m} := 3 \quad \underline{n} := 3 \\ \underline{\text{ORIGIN}} := 1 \\ \underline{c} := \begin{pmatrix} 18 \\ 12 \\ 16 \end{pmatrix} \quad \underline{a} := \begin{pmatrix} 0.6 & 0.4 & 0.6 \\ 0 & 0.3 & 0.9 \\ -0.2 & 0.1 & 0.1 \end{pmatrix} \quad \underline{b} := \begin{pmatrix} 150 \\ 20 \\ 120 \end{pmatrix} \quad \underline{x} := \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \\ \underline{f(x)} := \sum_{j=1}^n (c_j \cdot x_j) \quad x_n := 0 \end{array}$$

Рисунок 2.4. Ввод исходных данных в матричной форме

Поиск экстремума в матричной форме приводит так же к заданию системы ограничений матричными элементами, введению условия неотрицательности, поиску максимума функции цели и определению самой точки экстремума и значению функции цели в этой точке.

Общий вид документа решения задачи представлен на рис. 2.5.

$$\begin{array}{l}
 \underline{m} := 3 \quad \underline{n} := 3 \\
 \underline{\text{ORIGIN}} := 1 \\
 \underline{c} := \begin{pmatrix} 18 \\ 12 \\ 16 \end{pmatrix} \quad \underline{a} := \begin{pmatrix} 0.6 & 0.4 & 0.6 \\ 0 & 0.3 & 0.9 \\ -0.2 & 0.1 & 0.1 \end{pmatrix} \quad \underline{b} := \begin{pmatrix} 150 \\ 20 \\ 120 \end{pmatrix} \quad \underline{x} := \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \\
 f(x) := \sum_{j=1}^n (c_j \cdot x_j) \quad x_n := 0 \\
 \text{GIVEN} \\
 \underline{a} \cdot \underline{x} \leq \underline{b} \quad \underline{x} \geq 0 \\
 \underline{x} := \text{Maximize}(f, \underline{x}) \\
 \underline{x} = \begin{pmatrix} 205.556 \\ 66.667 \\ 0 \end{pmatrix} \quad f(x) = 4.5 \times 10^3
 \end{array}$$

Рисунок 2.5. Общий вид решения задачи в матричной форме

2.5. Задания для самостоятельной работы

Задача 1

Найти экстремум, используя матричные формы записи процессора MathCAD:

$$f(x) = 108 \cdot x_1 + 112 \cdot x_2 + 126 \cdot x_3 \rightarrow \max$$

$$\begin{cases} 0,8 \cdot x_1 + 0,5 \cdot x_2 + 0,6 \cdot x_3 \leq 800 \\ 0,4 \cdot x_1 + 0,4 \cdot x_2 + 0,9 \cdot x_3 \leq 600 \\ 0,1 \cdot x_2 + 0,1 \cdot x_3 \leq 120 \\ x_j \geq 0 \quad i = 1, 2, 3 \end{cases}$$

Задача 2

Найти экстремум, используя матричные формы записи процессора

MathCAD:

$$\begin{aligned} f(x) &= 6 \cdot x_1 + 4 \cdot x_2 + 8 \cdot x_3 \rightarrow \max \\ \begin{cases} 0,1 \cdot x_1 + 0,4 \cdot x_2 + 0,11 \cdot x_3 \leq 1 \\ 0,2 \cdot x_1 + 0,1 \cdot x_2 + 0,13 \cdot x_3 \leq 9 \\ 0,3 \cdot x_1 + 0,14 \cdot x_3 \leq 5 \\ x_j \geq 0 \quad i = 1, 2, 3 \end{cases} \end{aligned}$$

Задача 3

Найти экстремум, используя матричные формы записи процессора

MathCAD:

$$\begin{aligned} f(x) &= 24 \cdot x_1 + 15 \cdot x_2 + 38 \cdot x_3 \rightarrow \max \\ \begin{cases} 0,1 \cdot x_2 + 0,13 \cdot x_3 \leq 70 \\ 0,2 \cdot x_1 + 0,4 \cdot x_2 + 0,4 \cdot x_3 \leq 30 \\ 0,8 \cdot x_1 + 0,2 \cdot x_2 + 0,8 \cdot x_3 \leq 40 \\ x_j \geq 0 \quad i = 1, 2, 3 \end{cases} \end{aligned}$$

Задача 4

Найти экстремум, используя матричные формы записи процессора

MathCAD:

$$f(x) = 14 \cdot x_1 + 35 \cdot x_2 + 38 \cdot x_3 \rightarrow \max$$

$$\begin{cases} 0,1 \cdot x_2 + 0,3 \cdot x_3 \leq 700 \\ 0,5 \cdot x_1 + 0,4 \cdot x_3 \leq 300 \\ 0,8 \cdot x_1 + 0,2 \cdot x_2 + 0,8 \cdot x_3 \leq 400 \\ x_j \geq 0 \quad i = 1, 2, 3 \end{cases}$$

Задача 5

Найти экстремум, используя матричные формы записи процессора

MathCAD:

$$f(x) = 106 \cdot x_1 + 104 \cdot x_2 + 107 \cdot x_3 \rightarrow \max$$

$$\begin{cases} 0,2 \cdot x_1 + 0,1 \cdot x_3 \leq 23 \\ 0,2 \cdot x_2 + 0,3 \cdot x_3 \leq 29 \\ 0,9 \cdot x_1 + 0,2 \cdot x_2 + 0,4 \cdot x_3 \leq 17 \\ x_j \geq 0 \quad i = 1, 2, 3 \end{cases}$$

Задача 6

Найти экстремум, используя матричные формы записи процессора

MathCAD:

$$f(x) = 166 \cdot x_1 + 141 \cdot x_2 + 178 \cdot x_3 \rightarrow \max$$

$$\begin{cases} 0,3 \cdot x_1 + 0,8 \cdot x_2 + 0,1 \cdot x_3 \leq 32 \\ 0,63 \cdot x_1 + 0,5 \cdot x_2 \leq 22 \\ 0,9 \cdot x_1 + 0,2 \cdot x_2 + 0,4 \cdot x_3 \leq 16 \\ x_j \geq 0 \quad i = 1, 2, 3 \end{cases}$$

3. Идентификация стационарных точек функции одной переменной

Функция, которая обладает свойством непрерывности в каждой точке x_i , принадлежащей области определения данной функции, является *непрерывной функцией*. Если рассматривать две произвольные точки x_1 и x_2 функции (причем $x_1 \leq x_2$) и на всем протяжении выполнения функциональной зависимости соблюдаются неравенства $f(x_1) \leq f(x_2)$, то функция является монотонной: *монотонно возрастающая функция*, если $f(x_1) \leq f(x_2)$, и *монотонно убывающая функция*, если $f(x_1) \geq f(x_2)$.

Функция $f(x)$, описывающая некоторый процесс при исследовании физических задач, является целевой функцией. Если необходимо рассматривать определенный отрезок зависимости $a \leq x \leq b$, когда точка x^* является единственной точкой экстремума функции $f(x)$ на отрезке $a \leq x \leq b$, $f(x)$ представляется *униmodalной* на данном интервале. Если $x^* \leq x_1 \leq x_2$, функция соответствует неравенству $f(x^*) \leq f(x_1) \leq f(x_2)$. Если же $x^* \geq x_1 \geq x_2$, функция соответствует неравенству $f(x^*) \geq f(x_1) \geq f(x_2)$.

Определенная на множестве G функция достигает своего глобально максимума в точке $x^* \in G$ только в том случае, если $f(x^*) \geq f(x)$ при всех $x \in G$. Локальный максимум (относительный максимум) в точке определения функции $x^* \in G$ только тогда, если справедливо неравенство $f(x^*) \geq f(x)$ для всех x , удаленных от x^* на расстояние меньшее δ , $\delta > 0$. Для всех x , удовлетворяющих условию $|x - x^*| < \delta$, должно быть выполнено неравенство:

$$f(x^*) \geq f(x) \quad (3.1)$$

Если функция является *униmodalной*, тогда локальный максимум автоматически является глобальным максимумом. Если же функция не является *униmodalной*, то есть возможность наличия нескольких локальных

оптимумов, глобальный максимум можно найти только путем нахождения всех локальных максимумов и определения наибольшего из них.

3.1. Максимизация нелинейной унимодальной функции

При выполнении задачи необходимо определить максимальное значение нелинейной функции $f(x)$ при ограничениях $a \leq x \leq b$. Переменная x изменяет свое значение в границах a и b . Рассмотрение значений функции в заданном интервале приводит к необходимости проверки локального оптимума не только в стационарных точках, но и в точках, определяющих границы интервала.

Решение задачи производится следующим образом:

1. Приравнивается $\frac{df}{dx} = 0$, находятся все стационарные точки.
2. Определяются стационарные точки, которые расположены в интервале $[a, b]$ и сами a и b точки, находящиеся на границах интервала.
3. Находятся наибольшие значения функции в точках, вычисленных ранее. Полученное значение будет соответствовать глобальному максимуму.
4. При необходимости определения глобального минимума алгоритм выполнения задачи идентичен, но находится минимум функции.

3.2. Пример выполнения задания

Рассматривается функция, определенная на всей оси. Находятся стационарные точки, и они классифицируются.

$$f(x) = 5 \cdot x^6 - 36 \cdot x^5 + \frac{165}{2} \cdot x^4 - 60 \cdot x^3 + 36 \quad (3.2)$$

Первая производная исследуемой функции будет обращена в ноль в точках, значит эти точки являются стационарными.

$$\begin{aligned}\frac{df(x)}{dx} &= 30 \cdot x^5 - 180 \cdot x^4 + 330 \cdot x^3 - 180 \cdot x^2 \\ &= 30 \cdot x^2(x-1)(x-2)(x-3) \\ x_0 &= 0, x_1 = 1, x_2 = 2, x_3 = 3\end{aligned}\tag{3.3}$$

Вторая производная исследуемой функции будет также рассчитана, и найдены значения в стационарных точках.

$$\frac{d^2f(x)}{dx^2} = 150 \cdot x^4 + 720 \cdot x^3 - 990 \cdot x^2 - 360 \cdot x\tag{3.4}$$

$$\begin{aligned}x_0 = 0, f(x) = 36, \frac{d^2f(x)}{dx^2} = 0; \quad x_1 = 1, f(x) = 27,5, \frac{d^2f(x)}{dx^2} = 60; \\ x_2 = 2, f(x) = 44, \frac{d^2f(x)}{dx^2} = -120; \quad x_3 = 3, f(x) = 5,5, \frac{d^2f(x)}{dx^2} = 540.\end{aligned}$$

Изучение полученных значений позволяет сделать вывод, что точки локальных минимумов функции: $x_1 = 1$ и $x_3 = 3$. Точка же $x_2 = 2$ является точкой локального максимума.

Для определения точки $x_0 = 0$ вычисляется третья производная функции:

$$\frac{d^3f(x)}{dx^3} = (600 \cdot x^3 + 2160 \cdot x^2 - 1980 \cdot x - 360)|_{x=0} = -360\tag{3.5}$$

Полученная производная, отличная от нуля, является точкой перегиба.

3.3. Пример выполнения задачи в математическом процессоре

С использованием блока Given - Find производится максимизация функции на интервале:

$$-2 \leq x \leq 5$$

$$f(x) = -x^3 + 5 \cdot x^2 + 3 \cdot x + 7$$

Находятся стационарные точки, производится их идентификация:

$$f3(x) := -x^3 + 5 \cdot x^2 + 3 \cdot x + 7 \quad \text{xgr1} := -2 \quad \text{xgr2} := 5 \quad \text{x3} := \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Given

—————>

$$\frac{d}{dx3} f3(x3) = 0 \quad \text{xgr2} \geq \text{x3} \geq \text{xgr1} \quad \text{кор} := \text{Find}(x3)$$

$$\text{кор}_0 = -0.277 \quad \text{кор}_1 = 3.61 \quad \text{кор} = \begin{pmatrix} -0.277 \\ 3.61 \end{pmatrix}$$

Рисунок 3.1. Ввод исходных и граничных данных для решения задачи

Производится определение точки глобального максимума.

Значение рассчитывается в точке $\text{кор}_0 = -0.277$ $\text{кор}_1 = 3.61$.

Стационарная точка и граница интервала расчета показана на рисунке 3.2. Максимальное значение функция принимает в точке $\text{кор}_0 = -0.277$.

$$\text{кор}_0 = -0.277 \quad \text{кор}_1 = 3.61 \quad \text{кор} = \begin{pmatrix} -0.277 \\ 3.61 \end{pmatrix}$$

Стационарные точки

Границы интервала расчета

$$f3(\text{кор}_0) = 6.574$$

$$f3(\text{xgr1}) = 29$$

$$f3(\text{кор}_1) = 35.945$$

$$f3(\text{xgr2}) = 22$$

Рисунок 3.2. Определение значений точек граничных в интервале и величины максимума

Далее задаются данные для построения графика функции, производится нанесение данных стационарной точки:

$$\begin{aligned}
 &x3_max := xgr2 + 1 && x3_min := xgr1 - 1 \\
 &\text{Шаг расчета} && \Delta x3 := 0.01 \\
 &\text{Определение количества точек} && N3 := \frac{x3_max - x3_min}{\Delta x3} \\
 &\text{расчета} && \\
 &\text{Расчет построения векторов X3 и Y3 для построения кривой} && \\
 &i3 := 0..N3 && X3_{i3} := x3_min + \Delta x3 \cdot i3 && Y3 := f3(X3)
 \end{aligned}$$

Рисунок 3.3. Подготовка данных для построения графика

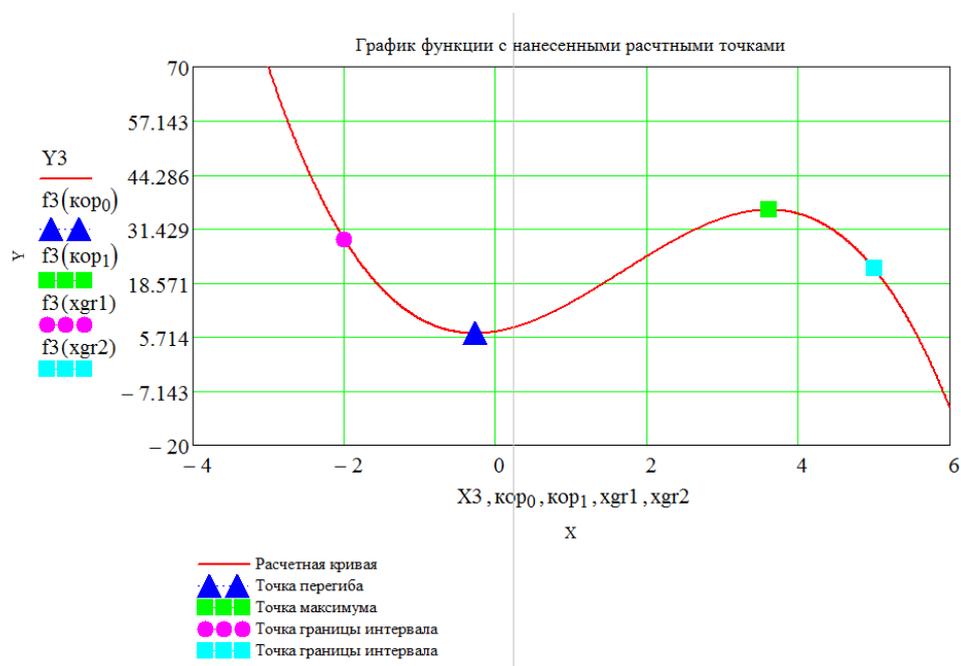


Рисунок 3.4. Итог построения функции с результатом максимизации

3.4. Задания для самостоятельной работы

Задача 1

С использованием блока Given - Find произвести минимизацию функции на заданном интервале при помощи математического процессора MathCAD:

$$-2 \leq x \leq 5$$

$$f(x) = -x^3 + 8 \cdot x^2 + 11 \cdot x + 3$$

Задача 2

С использованием блока Given - Find произвести минимизацию функции на заданном интервале при помощи математического процессора MathCAD:

$$-1 \leq x \leq 3$$

$$f(x) = x^3 + 2 \cdot x^2 + 9$$

Задача 3

С использованием блока Given - Find произвести минимизацию функции на заданном интервале при помощи математического процессора MathCAD:

$$-3 \leq x \leq 9$$

$$f(x) = x^3 - 3 \cdot x^2 + 2 \cdot x - 3$$

Задача 4

С использованием блока Given - Find произвести минимизацию функции на заданном интервале при помощи математического процессора MathCAD:

$$-2 \leq x \leq 6$$

$$f(x) = x^3 - 3 \cdot x^2 - 6 \cdot x$$

Задача 5

С использованием блока Given - Find произвести минимизацию функции на заданном интервале при помощи математического процессора MathCAD:

$$-2 \leq x \leq 6$$

$$f(x) = x^3 - 8 \cdot x^2 + 2 \cdot x - 5$$

Задача 6

С использованием блока Given - Find произвести минимизацию функции на заданном интервале при помощи математического процессора MathCAD:

$$-2 \leq x \leq 6$$

$$f(x) = x^3 - 4 \cdot x^2 + 2 \cdot x + 7$$

4. Математическая модель целочисленной задачи линейного программирования

Решение сложных математических задач не всегда выполняется с возможностью округления нецелых чисел, часто результат должен быть представлен только целочисленно – число животных, людей, транспортных средств, траекторий полетов самолета, количества планет,... Данный тип задач необходимо решать способами целочисленного линейного программирования.

Математическое линейное программирование, когда на экстремальные задачи устанавливается условие дискретности переменных с конечной областью допустимых решений, является дискретным линейным программированием. Если назначается условие целочисленности, то понимается целочисленное линейное программирование. При выполнении программы требуется буквальное указание, что результат поиска оптимальных значений может получать только целые значения.

Линейное программирование целочисленной задачи математической модели при этом:

$$f(\bar{x}) = \sum_{j=1}^n a_{ij} \cdot x_j \rightarrow \max(\min) \quad (4.1)$$

Ограничения при этом выглядят следующим образом.

$$\sum_{j=1}^n a_{ij} \cdot x_j \leq (\geq) b_i, i = 1 \dots m \quad (4.2)$$

$$x_j \geq 0, j = 1 \dots n$$

$$x_j - \text{целые}, j = 1 \dots n_1$$

Получается, что заранее определен ряд значений, которые может принять переменная x_j . Причем, если $n_1 < n$, задача линейного программирования является частично целочисленной, если же $n_1 = n$ – полностью целочисленной.

Технические и экономические задачи чаще всего являются именно дискретными и носят целочисленный характер. Физическая неделимость элементов расчета представлена тем, что нельзя изготовить 1,3 детали или построить 2,7 жилых дома...

Решение проблемы обычными способами, например симплекс методом, с последующим округлением до целых чисел может внести серьезные искажения в оптимальность итогового значения (например, автоматическое округление до 103 может привести к необходимости закупки еще 97 лишних элементов товара). При решении задач оптимизации классически являются задачи целочисленного программирования: задача о наполнении ранца школьными предметами (или задача о рюкзаке), задача о раскрое материала, задача о коммивояжере.

Если задача линейного программирования имеет две переменные, то она может быть решена графическим методом. В системе координат находится область допустимых решений, строится вектор C и линия уровня. Линия уровня перемещается по направлению вектора C . При поиске максимума находится наиболее удаленная от начала координат точка и ее координаты. Если координаты искомой точки нецелочисленные, в части допустимых решений строится целочисленная решетка и на ней определяются целые числа, которые удовлетворяют системе ограничений и при этом наиболее близки к экстремальному нецелочисленному решению. Координаты полученной вершины будут целочисленным решением.

Целочисленный минимум находится аналогично, только координаты вершины целочисленной решетки, лежащей в области допустимых решений, будут в направлении вектора C к началу координат.

Линейное программирование целочисленного исчисления при решении графическими методами начинается с построения линий ограничений на плоскости. После определения точки пересечения вершины многоугольника с целевой функцией, удовлетворяющей условию оптимальности, внутри определенной части плоскости строится еще один многоугольник, вершины которого имеют только целочисленные координаты.

Точка пересечения линии уровня целевой функции и вершины полученного многоугольника является оптимальным решением задачи целочисленного линейного программирования.

4.1. Выполнение оптимального целочисленного исчисления методом графического решения в процессоре MathCAD

Представлена математическая модель задачи оптимизации. Линейное программирование целочисленного решения задачи будет производиться графическим методом.

$$\begin{aligned} f(x, y) &= 5 \cdot x + 4 \cdot y \rightarrow \min \\ &\begin{cases} x + y \leq 9 \\ x + 2 \cdot y \geq 9 \\ 2 \cdot x + y \geq 9 \end{cases} \\ x \geq 0, y \geq 0, & \quad x \in \mathbb{Z}, y \in \mathbb{Z} \end{aligned} \quad (4.3)$$

Устанавливается минимизируемая целевая функция и задается система ограничений.

Производится построение допустимых областей значений для всех трех ограничений. В первую очередь вводится целевая функция

$$f(x, y) = 5 \cdot x + 4 \cdot y.$$

Затем записываются ограничения, явно обозначающие границы рассматриваемой области

$$\begin{aligned}
 x + y &\leq 9 \\
 x + 2 \cdot y &\geq 9. \\
 2 \cdot x + y &\geq 9
 \end{aligned}$$

Позже выполняется выражение переменной из всех ограничений и отображаются в координатных осях геометрические изображения ограничений.

$$f(x, y) := 5 \cdot x + 4 \cdot y \quad x + y \leq 9 \quad x + 2 \cdot y \geq 9 \quad 2 \cdot x + y \geq 9 \quad x \geq 0 \quad y \geq 0$$

$$\begin{aligned}
 x + y &= 9 & x + 2 \cdot y &= 9 & 2 \cdot x + y &= 9 \\
 -x + 9 && \frac{-1 \cdot x}{2} + \frac{9}{2} && -2 \cdot x + 9 &
 \end{aligned}$$

$$y1(x) := -x + 9 \quad y2(x) := \frac{-1 \cdot x}{2} + \frac{9}{2} \quad y3(x) := -2 \cdot x + 9$$

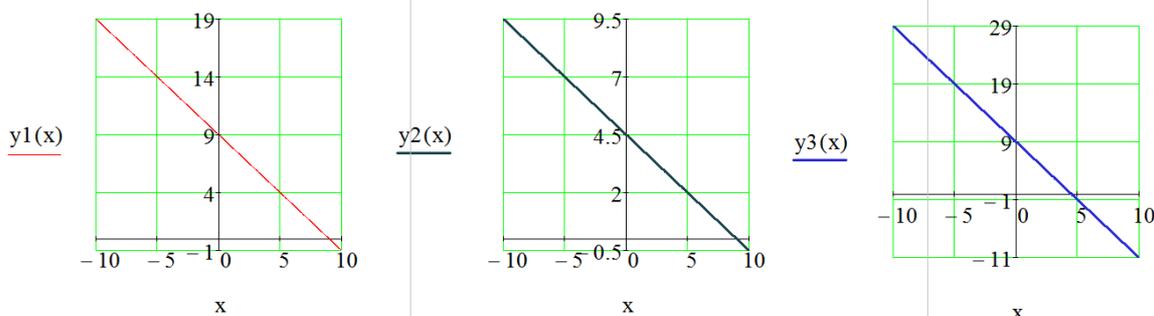
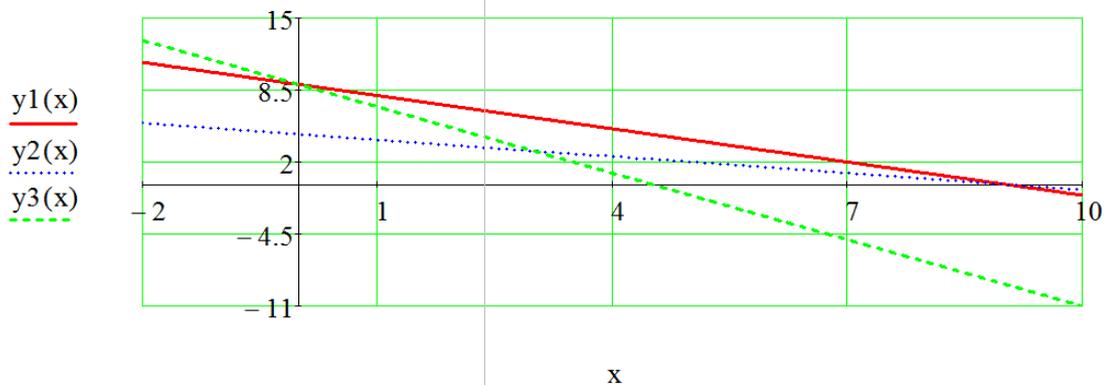


Рисунок 4.1. Ввод данных и построение линий ограничения области допустимых значений

Первый график ограничивает нижнюю полуплоскость, потому что ограничением задано "меньше или равно". Второе и третье ограничение используют неравенство "больше или равно", поэтому исследуются верхние полуплоскости.

Следующим шагом производится общее построение на одном графике всех трех прямых, в середине появляется плоскость включения области допустимых значений. Применяя производную константу, вводится уравнение линии, определяющей уровень целевой функции. Если в MathCAD

запустить трассировку, то можно определить координаты точки минимума, но достаточно точного указания значений добиться будет проблематично.



$$5 \cdot x + 4 \cdot y = C \quad \frac{-5}{4} \cdot x + \frac{1}{4} \cdot C$$

Рисунок 4.2. Получение графика с выделением области искомых значений

Для координат x и y находятся соответственно возможные значения экстремума

$$X_{kon}, Y_{kon}$$

Записывается программный модуль поиска и получения массива значений координат, численные величины которых могут выражены только в целочисленном варианте.

$y4(x, C) := \frac{-5}{4} \cdot x + \frac{1}{4} \cdot C$	$Formir(X_{kon}, Y_{kon}) :=$	<pre> i ← 0 for x ∈ 0 .. X_kon for y ∈ 0 .. Y_kon if (x + y ≤ 9) ∧ (x + 2·y ≥ 9) ∧ (2·x + y ≥ 9) X_{i,0} ← x X_{i,1} ← y i ← i + 1 </pre>
$fc := Formir(10, 10)$		X

Рисунок 4.3. Программный модуль поиска целочисленных решений.

Программный модуль, отображенный на рисунке 4.3, производит поиск решения задачи, когда необходимо из области графического решения выделить только целые значения искомых величин.

На рисунке 4.4 при помощи математического процессора MathCAD результативно решена задача, когда в области, ограниченной условиями, определены целочисленные значения вариантов решения задачи.

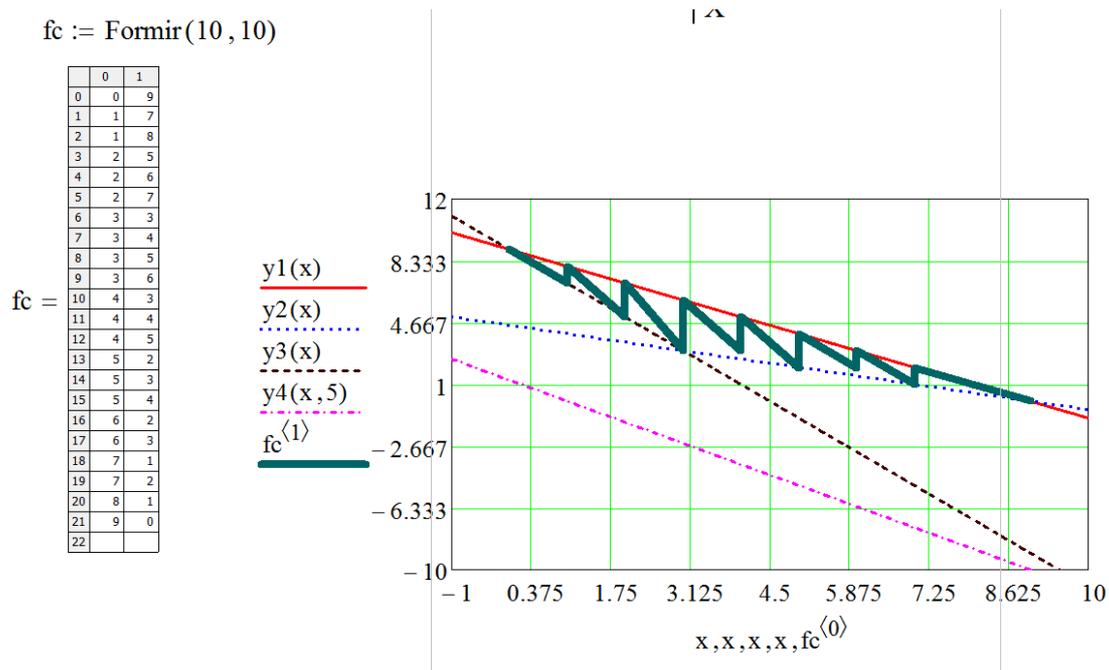


Рисунок 4.4. Координаты решений целых значений искомых величин

Табулирование позволяет и вывести все ответы искомых величин с их координатными определениями. Зигзаг на графике весьма успешно обозначает сколько целых значений находится на одной координатной прямой. Наибольшее число целочисленных решений задачи принадлежит величине, когда $x = 3$.

$$\begin{array}{l}
 \text{---} \\
 x := 0 \quad y := 0 \\
 \text{Given} \\
 x + y \leq 9 \quad x + 2 \cdot y \geq 9 \quad 2 \cdot x + y \geq 9 \\
 x \geq 0 \quad y \geq 0 \\
 \\
 z := \text{Minimize}(f, x, y) \\
 \\
 f(z_0, z_1) = 27 \\
 \\
 z = \begin{pmatrix} 3 \\ 3 \end{pmatrix}
 \end{array}$$

Рисунок 4.5. Расчет точки минимума

После рассмотрения построенных линий уровня определяется, куда направлен антиградиент. Видно, что в пересечении линий и их минимального значения в точке $z = (3,3)$ можно найти целое число минимального решения задачи. Оно равно $f(3,3) = 27$.

4.2. Задания для самостоятельного решения задачи оптимизационного целочисленного исчисления графическими методами

Задача 1

Выполнить математическую модель задачи оптимизации. При помощи целочисленного программирования решить задачу графическим методом. Установить минимальное значение пересечения трех линий.

$$\begin{aligned} f(x, y) &= 5 \cdot x + 4 \cdot y \rightarrow \min \\ &\begin{cases} x + y \leq 9 \\ x + 2 \cdot y \geq 9 \\ 2 \cdot x + y \geq 9 \end{cases} \\ x \geq 0, y \geq 0, & \quad x \in \mathbb{Z}, y \in \mathbb{Z} \end{aligned}$$

Задача 2

Выполнить математическую модель задачи оптимизации. При помощи целочисленного программирования решить задачу графическим методом. Установить минимальное значение пересечения трех линий.

$$\begin{aligned} f(x, y) &= 3 \cdot x + 2 \cdot y \rightarrow \min \\ &\begin{cases} x + y \leq 7 \\ x + 2 \cdot y \geq 7 \\ 2 \cdot x + y \geq 7 \end{cases} \\ x \geq 0, y \geq 0, & \quad x \in \mathbb{Z}, y \in \mathbb{Z} \end{aligned}$$

Задача 3

Выполнить математическую модель задачи оптимизации. При помощи целочисленного программирования решить задачу графическим методом. Установить минимальное значение пересечения трех линий.

$$\begin{aligned} f(x, y) &= 7 \cdot x + y \rightarrow \min \\ &\begin{cases} x + y \leq 14 \\ 3 \cdot x - 5 \cdot y \leq 15 \\ 5 \cdot x + y \geq 21 \end{cases} \\ x \geq 0, y \geq 0, & \quad x \in \mathbb{Z}, y \in \mathbb{Z} \end{aligned}$$

Задача 4

Выполнить математическую модель задачи оптимизации. При помощи целочисленного программирования решить задачу графическим методом. Установить минимальное значение пересечения трех линий.

$$\begin{aligned} f(x, y) &= 2 \cdot x + y \rightarrow \min \\ &\begin{cases} x - 2 \cdot y \geq 4 \\ 5 \cdot x + 2 \cdot y \geq 10 \\ 7 \cdot x + y \leq 23 \end{cases} \\ x \geq 0, y \geq 0, \quad x \in \mathbb{Z}, y \in \mathbb{Z} \end{aligned}$$

Задача 5

Выполнить математическую модель задачи оптимизации. При помощи целочисленного программирования решить задачу графическим методом. Установить минимальное значение пересечения трех линий.

$$\begin{aligned} f(x, y) &= 3 \cdot x + 4 \cdot y \rightarrow \min \\ &\begin{cases} x + y \leq 11 \\ x + 2 \cdot y \geq 14 \\ 5 \cdot x + y \geq 13 \end{cases} \\ x \geq 0, y \geq 0, \quad x \in \mathbb{Z}, y \in \mathbb{Z} \end{aligned}$$

Задача 6

Выполнить математическую модель задачи оптимизации. При помощи целочисленного программирования решить задачу графическим методом. Установить минимальное значение пересечения трех линий.

$$\begin{aligned} f(x, y) &= 8 \cdot x + 3 \cdot y \rightarrow \min \\ &\begin{cases} x + y \leq 19 \\ x + 2 \cdot y \geq 12 \\ 2 \cdot x + y \geq 24 \end{cases} \\ x \geq 0, y \geq 0, \quad x \in \mathbb{Z}, y \in \mathbb{Z} \end{aligned}$$

5. Решение задач со многими переменными при помощи линейного программирования. Метод Жордана-Гаусса - полное исключение неизвестных

Если при канонической записи основной задачи линейного программирования присутствуют две свободные переменные, то она решается графическим способом. Целевая функция выражается через свободные переменные, и после алгебраических преобразований и получения задачи линейного программирования выполняется графическое решение.

Методом Жордана-Гаусса решаются усложненные задачи. Задача линейного программирования дает оптимальное решение. Симплекс метод основан на упорядоченном переборе опорных решений и последовательном их улучшении.

Имеется система m линейных уравнений, в которой n неизвестных, причем $m \leq n$:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2, \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m, \end{cases} \quad (5.1)$$

Матрица данной системы $A = (a_{ij})_{m \times n}$ обозначается через $X = (x_1, x_2, \dots, x_n)^T$.

Столбец неизвестных записывается как $A_0 = (b_1, b_2, \dots, b_m)^T$. Тогда система уравнений записывается в виде:

$$A * X = A_0 \quad (5.2)$$

Когда матрица A содержит в каждом столбце ровно по одному ненулевому элементу, причем в разных строках, тогда переменные называются базисными. Свободные переменные (остальные) приводят решение системы уравнений к поиску коэффициентов, так как, задача решает систему

квадратной матрицы A^{-1} . При умножении матричного уравнения на обратную матрицу, получается $A0$ решение системы:

$$E * X = A^{-1} \cdot A_0 \quad (5.3)$$

Метод алгебраических дополнений для нахождения обратной матрицы весьма проблематичный и долгий математический процесс, проще использовать численный метод. Производя элементарные преобразования, находится через некоторое число шагов решение. В уравнениях находятся коэффициенты неизвестных, и на них делится уравнение и неизвестное исключается. После исключения неизвестных в каждом из уравнений система достаточно легко решается и выполняется обратный процесс поиска неизвестных. При этом будет получен вывод, что либо система несовместна, либо же найдено решение. Метод Жордана Гаусса называется и также методом исключения неизвестных.

5.1. Пример решения задачи

Найти оптимальное решение задачи линейного программирования, при которых выполняются условия:

$$\begin{aligned} x_j &\geq 0 \\ f(x_1, x_2, x_3, x_4) &= 2 \cdot x_1 + 2 \cdot x_2 - x_3 + x_4 \rightarrow \max \end{aligned} \quad (5.4)$$

если должны выполняться условия поставленной задачи:

$$\begin{cases} x_1 + 2 \cdot x_2 - x_3 - x_4 = 2 \\ 2 \cdot x_1 + x_2 + 2 \cdot x_4 = 1 \\ x_1, x_2, x_3, x_4 \geq 0 \end{cases} \quad (5.5)$$

Производится решение задачи методом Жордана-Гаусса:

JM (M, a, b) M – исходная расширенная матрица

a – номер строки разрешающего элемента

b – номер столбца разрешающего элемента

$$\begin{array}{l}
 \text{ORIGIN} := 1 \\
 \text{JG}(M, a, b) := \left| \begin{array}{l}
 \text{for } i \in 1 \dots \text{rows}(M) \\
 \quad \text{for } j \in 1 \dots \text{cols}(M) \\
 \quad \quad \left| \begin{array}{l}
 N_{i,j} \leftarrow \frac{M_{i,j}}{M_{a,b}} \quad \text{if } i = a \\
 N_{i,j} \leftarrow M_{i,j} - M_{i,b} \cdot \frac{M_{a,j}}{M_{a,b}} \quad \text{otherwise}
 \end{array} \right. \\
 \quad \quad \text{N}
 \end{array} \right.
 \end{array}$$

Рисунок 5.1. Программный модуль решения матрицы JM (M, a, b) M методом Жордана-Гаусса

Далее задается расширенная матрица, выполненная с элементами правых частей уравнений:

$$\begin{array}{l}
 \underline{A} := \left(\begin{array}{cc|ccc}
 1 & 2 & -1 & -1 & 2 \\
 2 & 0 & 1 & 2 & 1
 \end{array} \right) \\
 \\
 \text{B1} := \text{JG}(A, 1, 2) \qquad \text{B1} = \left(\begin{array}{cc|ccc}
 0.5 & 1 & -0.5 & -0.5 & 1 \\
 2 & 0 & 1 & 2 & 1
 \end{array} \right) \\
 \\
 \text{B2} := \text{JG}(B1, 2, 1) \qquad \text{B2} = \left(\begin{array}{cc|ccc}
 0 & 1 & -0.75 & -1 & 0.75 \\
 1 & 0 & 0.5 & 1 & 0.5
 \end{array} \right)
 \end{array}$$

Рисунок 5.2. Определяются базисные переменные x_1, x_2 и свободные переменные x_3, x_4

Производятся преобразования и из полученной матрицы В2 выполняются уравнения, задающие ограничения:

$$\begin{cases} x_2 - 0,75 \cdot x_3 - x_4 = 0,75 \\ x_1 + 0,5 \cdot x_3 + x_4 = 0,5 \end{cases} \quad (5.6)$$

$$f(x_1, x_2, x_3, x_4) = 2 \cdot x_1 + 2 \cdot x_2 - x_3 + x_4 \rightarrow \max$$

Базисные переменные выражаются через свободные переменные:

$$\begin{cases} x_2 = 0,75 \cdot x_3 + x_4 + 0,75 \\ x_1 = -0,5 \cdot x_3 - x_4 + 0,5 \\ x_1 \geq 0 \quad x_2 \geq 0 \end{cases} \quad (5.7)$$

Можно производить и дальнейшие преобразования системы уравнений:

$$\begin{cases} 0,75 \cdot x_3 + x_4 \geq -0,75 \\ -0,5 \cdot x_3 - x_4 \geq -0,5 \end{cases} \leftrightarrow \begin{cases} 0,75 \cdot x_3 + x_4 \geq -0,75 \\ 0,5 \cdot x_3 + x_4 \leq 0,5 \end{cases} \quad (5.8)$$

Целевая функция принимает вид:

$$f(x_3, x_4) = 2 \cdot (-0,5 \cdot x_3 - x_4 + 0,5) + 2 \cdot (0,75 \cdot x_3 + x_4 + 0,75) - x_3 + x_4 \rightarrow \max \quad (5.9)$$

$$f(x_3, x_4) = -0,5 \cdot x_3 + x_4 + 2,5 \rightarrow \max$$

Целевая функция с ограничениями принимает вид:

$$\begin{cases} f(x_3, x_4) = -0,5 \cdot x_3 + x_4 + 2,5 \rightarrow \max \\ 0,75 \cdot x_3 + x_4 \geq -0,75 \\ 0,5 \cdot x_3 + x_4 \leq 0,5 \end{cases} \quad (5.10)$$

Далее полученная задача может быть достаточно быстро решена способом, описанным в предыдущем параграфе.

5.2. Задания для самостоятельного решения задачи методом полного исключения неизвестных

Задача 1

Выполнить решение задачи линейного программирования методом Жордана-Гаусса оптимальное, при которых будут выполнены условия:

$$x_j \geq 0$$
$$\begin{cases} -x_1 + 2 \cdot x_2 + x_3 = 2 \\ x_1 + x_2 + x_4 = 2 \\ 2 \cdot x_1 + x_2 + x_3 + x_4 + 2 \cdot x_5 = 6 \end{cases}$$
$$f(x_1, x_2, x_3, x_4, x_5) = 3 \cdot x_2 + x_3 - x_4 + x_5 \rightarrow \max$$

Задача 2

Выполнить решение задачи линейного программирования методом Жордана-Гаусса оптимальное, при которых будут выполнены условия:

$$x_j \geq 0$$
$$\begin{cases} 2 \cdot x_1 + x_2 + x_3 + x_4 + 3 \cdot x_5 = 5 \\ 3 \cdot x_1 + 2 \cdot x_3 - x_4 + 6 \cdot x_5 = 7 \\ x_1 - x_3 + 2 \cdot x_4 + x_5 = 2 \end{cases}$$
$$f(x_1, x_2, x_3, x_4, x_5) = +3 \cdot x_3 - 2 \cdot x_4 - x_5 \rightarrow \max$$

Задача 3

Выполнить решение задачи линейного программирования методом Жордана-Гаусса оптимальное, при которых будут выполнены условия:

$$x_j \geq 0$$
$$\begin{cases} -x_1 + 2 \cdot x_2 + x_3 = 2 \\ 2 \cdot x_1 + 6 \cdot x_2 + 2 \cdot x_3 - x_4 + x_5 = 18 \\ x_1 - 2 \cdot x_2 + x_5 = 12 \end{cases}$$
$$f(x_1, x_2, x_3, x_4, x_5) = 6 \cdot x_1 + x_2 + x_4 + 2 \cdot x_5 \rightarrow \max$$

Задача 4

Выполнить решение задачи линейного программирования методом Жордана-Гаусса оптимальное, при которых будут выполнены условия:

$$\begin{aligned} & x_j \geq 0 \\ & \begin{cases} -2 \cdot x_1 + 3 \cdot x_3 + x_4 + x_5 = 5 \\ 3 \cdot x_1 + x_2 + x_3 + 6 \cdot x_4 + 2 \cdot x_5 = 9 \\ -x_1 + 2 \cdot x_3 - x_4 + 2 \cdot x_5 = 3 \end{cases} \\ & f(x_1, x_2, x_3, x_4, x_5) = -8 \cdot x_1 - x_2 \\ & \quad -x_3 + x_4 \rightarrow \max \end{aligned}$$

Задача 5

Выполнить решение задачи линейного программирования методом Жордана-Гаусса оптимальное, при которых будут выполнены условия:

$$\begin{aligned} & x_j \geq 0 \\ & \begin{cases} -x_1 + x_2 + x_3 = 2 \\ 5 \cdot x_1 + 2 \cdot x_2 + x_3 + x_4 + x_5 = 11 \\ 3 \cdot x_1 + 2 \cdot x_2 + x_5 = 6 \end{cases} \\ & f(x_1, x_2, x_3, x_4, x_5) = 6 \cdot x_1 - x_2 + \\ & \quad + 2 \cdot x_3 - x_4 + x_5 \rightarrow \max \end{aligned}$$

Задача 6

Выполнить решение задачи линейного программирования методом Жордана-Гаусса оптимальное, при которых будут выполнены условия:

$$\begin{aligned} & x_j \geq 0 \\ & \begin{cases} x_1 - x_2 + x_3 = 1 \\ 2 \cdot x_1 + 2 \cdot x_2 + x_3 + x_4 + 2 \cdot x_5 = 12 \\ 2 \cdot x_1 + x_2 + x_5 = 4 \end{cases} \\ & f(x_1, x_2, x_3, x_4, x_5) = 7 \cdot x_1 + \\ & \quad + x_3 - x_4 + x_5 \rightarrow \max \end{aligned}$$

6. Поиск точки оптимума внутри заданного интервала численными методами

Поиск оптимальных значений задача часто стратегическая во многих сферах жизнедеятельности человека. Методы поиска оптимума делятся на три основных: методы точечного оценивания и методы исключения интервала, а так же смешанные методы. Оптимум функции определяется при помощи исключения подинтервалов, а, значит, сокращением числа интервалов поиска. Данный способ, который освобождает от необходимости полного перебора всех допустимых точек, называется методом исключения интервалов.

Метод основан на вычислении значений функции, причем, допускается, что функцию нельзя записать в аналитическом виде, но она должна быть дифференцируема. Однако, метод накладывает существенное ограничение: исследуемая функция должна быть унимодальная.

Методы исключения интервалов используют логическую структуру простого сравнения значений функции в двух рассматриваемых точках без учета разности между значениями функции.

Методы точечного оценивания учитывают относительные изменения функции и ее производных. Методы точечного оценивания наиболее эффективны, но требуют введения достаточных допущений. Исследуемые функции должны быть достаточно гладкими и позволять проводить аппроксимацию полиномом с оценкой координаты точки оптимума. В отличие от методов исключения интервалов методы точечного оценивания могут быть использованы только для исследования непрерывных функций. Использование теоремы Вейерштрасса позволяет аппроксимировать функцию полиномом достаточно высокого порядка. Координата точки оптимума рассматриваемой функции оценивается путем вычисления координаты

точки оптимума полинома. Качество решения задачи можно повысить повышением порядка полинома или же уменьшением интервала аппроксимации.

6.1. Методы исключения

К методам исключения интервала с последовательной стратегией относится метод деления интервала пополам. При каждой итерации исключается ровно половина текущего интервала неопределенности. Алгоритм решения задачи начинается с анализа величины функции в трех точках, равномерно распределенных на текущем интервале. Окончание процесса деления заканчивается в том случае, если длина текущего интервала неопределенности получается меньше заданной величины.

Начальные условия и варианты реализации метода деления интервала пополам показаны на рис. 6.1. В первом случае $f(x_1) \geq f(x_c)$ и $f(x_2) \geq f(x_c)$, во втором случае $f(x_1) \leq f(x_c)$ и $f(x_2) \geq f(x_c)$, в третьем случае $f(x_1) \geq f(x_c)$ и $f(x_2) \leq f(x_c)$.

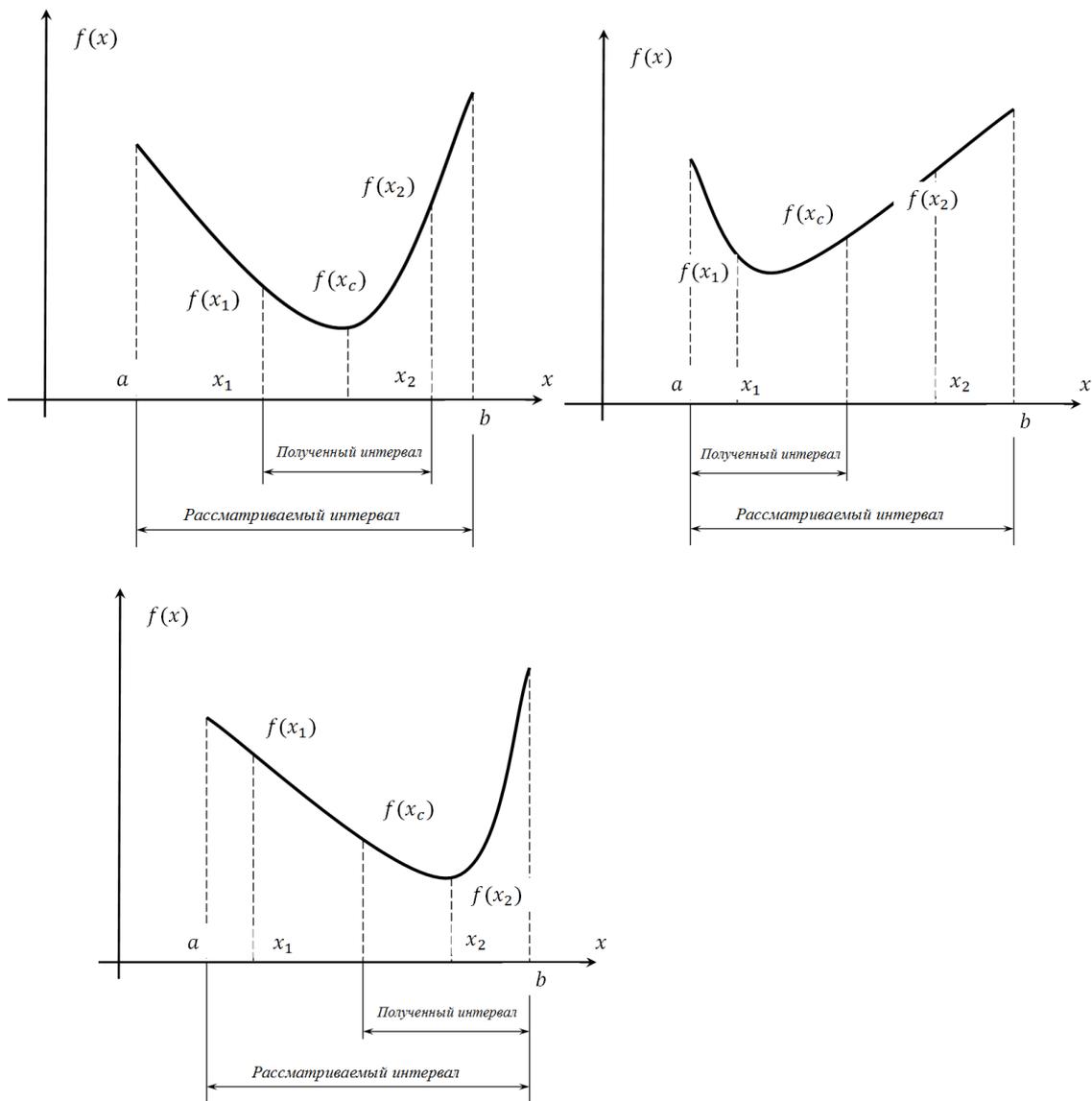


Рисунок 6.1. Начальные условия и варианты реализации метода деления интервала пополам

Процесс перебора интервалов основывается на анализе величины рассматриваемого интервала неопределенности.

6.2. Решение задачи поиска оптимума методом деления интервала пополам

Исследуется функция:

$$f(x) = 3x^2 - 15x \quad (6.1)$$

Задается интервал неопределенности и требуемая точность:

$$L_0 = [a_0; b_0], \Delta > 0 \quad (6.2)$$

Определяется средняя точка:

$$x_c = \frac{a + b}{2}, \text{ при предположении, что } k = 0 \quad (6.3)$$

Рассматривается интервал и находится значение в средней точке:

$$[a_k; b_k] - |L_k| = |b_k - a_k|, f(x_c) \quad (6.4)$$

Вычисляются точки и значения функции в этих точках. Происходит деление отрезка и сравниваются полученные значения.

$$x_1 = a_k + \frac{|L_k|}{4}; \quad x_2 = b_k - |L_k|/4 \quad (6.7)$$

$f(x) := 3 \cdot x^2 - 15 \cdot x$	$a_0 := 0$	$b_0 := 10$	$\Delta L := 1$	$k := 0$		
$a_k := a_0$	$a_k = 0$	$b_k := b_0$	$b_k = 10$			
$x_{c_k} := \frac{a_k + b_k}{2}$		$x_{c_k} = 5$				
$L_k := (a_k \ b_k)$		$L_k = (0 \ 10)$	$mL_k := b_k - a_k $	$mL_k = 10$		
$y_k := a_k + \frac{mL_k}{4}$	$y_k = 2.5$	$z_k := b_k - \frac{mL_k}{4}$	$z_k = 7.5$			
$f_{y_k} := f(y_k)$	$f_{y_k} = -18.75$	$f_{x_{c_k}} := f(x_{c_k})$	$f_{x_{c_k}} = 0$	$f_{z_k} := f(z_k)$	$f_{z_k} = 56.25$	
$f_{y_k} < f_{x_{c_k}} = 1$						
$a_{k+1} := a_k$	$a_{k+1} = 0$	$b_{k+1} := x_{c_k}$	$b_{k+1} = 5$	$x_{c_{k+1}} := y_k$	$x_{c_{k+1}} = 2.5$	
$L_{k+1} := (a_{k+1} \ b_{k+1})$		$L_{k+1} = (0 \ 5)$	$mL_{k+1} := b_{k+1} - a_{k+1} $	$mL_{k+1} = 5$		
$mL_{k+1} > \Delta L = 1$						

Рисунок 6.2. Первый этап проводимых вычислений

На рисунке 6.2 показан первый этап проводимых вычислений. Так как результат не достигнут, производится следующее вычисление – рисунок 6.3.

$k := k + 1 = 1$					
$y_k := a_k + \frac{mL_k}{4}$	$y_k = 1.25$	$z_k := b_k - \frac{mL_k}{4}$	$z_k = 3.75$		
$f_{y_k} := f(y_k)$	$f_{y_k} = -14.063$	$f_{x_{c_k}} := f(x_{c_k})$	$f_{x_{c_k}} = -18.75$	$f_{z_k} := f(z_k)$	$f_{z_k} = -14.063$
$f_{y_k} < f_{x_{c_k}} = 0$					
$f_{y_k} > f_{x_{c_k}} = 1$					
$a_{k+1} := y_k$	$a_{k+1} = 1.25$	$b_{k+1} := z_k$	$b_{k+1} = 3.75$	$x_{c_{k+1}} := x_{c_k}$	$x_{c_{k+1}} = 2.5$
$L_{k+1} := (a_{k+1} \ b_{k+1})$	$L_{k+1} = (1.25 \ 3.75)$	$mL_{k+1} := b_{k+1} - a_{k+1} $		$mL_{k+1} = 2.5$	
$mL_{k+1} > \Delta L = 1$					

Рисунок 6.3. Второй этап проводимых вычислений

Великолепие программы математических расчетов MathCAD еще раз демонстрирует свои возможности, позволяя избежать рутинных операций по набору формул и проводить итерационные вычисления путем простого копирования и вставки формульных решений.

$k := k + 1 = 2$ $y_k := a_k + \frac{mL_k}{4} \quad y_k = 1.875 \quad z_k := b_k - \frac{mL_k}{4} \quad z_k = 3.125$	
$f_{y_k} := f(y_k) \quad f_{y_k} = -17.578 \quad f_{x_{c_k}} := f(x_{c_k}) \quad f_{x_{c_k}} = -18.75 \quad f_{z_k} := f(z_k) \quad f_{z_k} = -17.578$	
$f_{y_k} > f_{x_{c_k}} = 1$	

$a_{k+1} := x_{c_k} \quad a_{k+1} = 2.5 \quad b_{k+1} := b_k \quad b_{k+1} = 3.75 \quad x_{c_{k+1}} := z_k \quad x_{c_{k+1}} = 3.125$	
$L_{k+1} := (a_{k+1} \quad b_{k+1}) \quad L_{k+1} = (2.5 \quad 3.75) \quad mL_{k+1} := b_{k+1} - a_{k+1} \quad mL_{k+1} = 1.25$	
$mL_{k+1} > \Delta L = 1$	

$k := k + 1 = 3$ $y_k := a_k + \frac{mL_k}{4} \quad y_k = 2.813 \quad z_k := b_k - \frac{mL_k}{4} \quad z_k = 3.438$	
$f_{y_k} := f(y_k) \quad f_{y_k} = -18.457 \quad f_{x_{c_k}} := f(x_{c_k}) \quad f_{x_{c_k}} = -17.578 \quad f_{z_k} := f(z_k) \quad f_{z_k} = -16.113$	
$f_{y_k} > f_{x_{c_k}} = 0$	
$f_{z_k} > f_{x_{c_k}} = 1$	

$a_{k+1} := y_k \quad a_{k+1} = 2.813 \quad b_{k+1} := z_k \quad b_{k+1} = 3.438 \quad x_{c_{k+1}} := x_{c_k} \quad x_{c_{k+1}} = 3.125$	
$L_{k+1} := (a_{k+1} \quad b_{k+1}) \quad L_{k+1} = (2.813 \quad 3.438) \quad mL_{k+1} := b_{k+1} - a_{k+1} \quad mL_{k+1} = 0.625$	
$mL_{k+1} > \Delta L = 0$	
$y_{k+1} := a_{k+1} + \frac{mL_{k+1}}{4} \quad y_{k+1} = 2.969 \quad z_{k+1} := b_{k+1} - \frac{mL_{k+1}}{4} \quad z_{k+1} = 3.281$	

Рисунок 6.4. Третий и четвертый этапы проводимых вычислений

Далее формируется матрица точек расчета и значений функции в этих точках. Используется известная встроенная функция системы MathCAD *augment*.

Решение матричного представления показано на рисунке 6.5.

$$X := \text{augment}(a, y, xc, z, b)$$

$$X = \begin{pmatrix} 0 & 2.5 & 5 & 7.5 & 10 \\ 0 & 1.25 & 2.5 & 3.75 & 5 \\ 1.25 & 1.875 & 2.5 & 3.125 & 3.75 \\ 2.5 & 2.813 & 3.125 & 3.438 & 3.75 \\ 2.813 & 2.969 & 3.125 & 3.281 & 3.438 \end{pmatrix}$$

$$\xrightarrow{\hspace{1cm}} \\ FX := f(X)$$

$$FX = \begin{pmatrix} 0 & -18.75 & 0 & 56.25 & 150 \\ 0 & -14.063 & -18.75 & -14.063 & 0 \\ -14.063 & -17.578 & -18.75 & -17.578 & -14.063 \\ -18.75 & -18.457 & -17.578 & -16.113 & -14.063 \\ -18.457 & -18.091 & -17.578 & -16.919 & -16.113 \end{pmatrix}$$

Рисунок 6.5. Решение задачи при помощи функции *augment*

$$L_{k+1} = (2.813 \quad 3.438)$$

$$xc_k = 3.125$$

$$f(xc_k) = -17.578$$

$$N := 2 \cdot \text{last}(a) = 8$$

Рисунок 6.6. Вывод результата решения

И, наконец, рисунок 6.6 демонстрирует решение задачи - конечный интервал неопределенности, точку минимума и общее количество вычислений (в данном случае количество вычислений равно 8).

6.3. Решение задачи с использованием пользовательской функции

Рассчитываются значения координаты для пяти точек a, x_1, x_c, x_2, b , затем для четырех точек a, x_1, x_c, x_2, b каждой итерации и координаты последней итерационной процедуры.

$$\begin{aligned}
 M_{ayzb} := X &= \begin{pmatrix} 0 & 2.5 & 5 & 7.5 & 10 \\ 0 & 1.25 & 2.5 & 3.75 & 5 \\ 1.25 & 1.875 & 2.5 & 3.125 & 3.75 \\ 2.5 & 2.813 & 3.125 & 3.438 & 3.75 \\ 2.813 & 2.969 & 3.125 & 3.281 & 3.438 \end{pmatrix} \\
 \xrightarrow{\quad} \\
 f_{\text{sug_int}} := f(M_{ayzb}) &= \begin{pmatrix} 0 & -18.75 & 0 & 56.25 & 150 \\ 0 & -14.063 & -18.75 & -14.063 & 0 \\ -14.063 & -17.578 & -18.75 & -17.578 & -14.063 \\ -18.75 & -18.457 & -17.578 & -16.113 & -14.063 \\ -18.457 & -18.091 & -17.578 & -16.919 & -16.113 \end{pmatrix} \\
 X_{ABmin} := \begin{bmatrix} (M_{ayzb}^{\langle 0 \rangle})_{\text{last}(M_{ayzb}^{\langle 0 \rangle})} \\ (M_{ayzb}^{\langle 4 \rangle})_{\text{last}(M_{ayzb}^{\langle 4 \rangle})} \end{bmatrix} & \quad X_{ABmin} = \begin{pmatrix} 2.813 \\ 3.438 \end{pmatrix}
 \end{aligned}$$

Рисунок 6.7. Расчет координат последнего интервала итерационной процедуры

$$X_ABmin := \begin{bmatrix} (M_ayzb^{\langle 0 \rangle})_{last}(M_ayzb^{\langle 0 \rangle}) \\ (M_ayzb^{\langle 4 \rangle})_{last}(M_ayzb^{\langle 4 \rangle}) \end{bmatrix} \quad X_ABmin = \begin{pmatrix} 2.813 \\ 3.438 \end{pmatrix}$$

$$Y_ABmin := f(X_ABmin) \quad Y_ABmin = \begin{pmatrix} -18.457 \\ -16.113 \end{pmatrix}$$

Рисунок 6.8. Определение точек и значений последнего интервала итерационной процедуры

Затем определяются координаты оптимальной точки. На рисунках с 6.1 по 6.9 продемонстрирован листинг работы программы разрешения функции. Показан алгоритм нахождения минимальной точки и значения функции в этой точке с заданной точностью.

$$Y_ABmin := f(X_ABmin) \quad Y_ABmin = \begin{pmatrix} -18.457 \\ -16.113 \end{pmatrix}$$

$$Xmin := \frac{X_ABmin_0 + X_ABmin_1}{2} \quad Ymin := f(Xmin)$$

$$Xmin = 3.125 \quad Ymin = -17.578$$

Рисунок 6.9 Точка минимума и значение функции в точке минимума

Метод деления интервала пополам важен процесс назначения характеристики относительного уменьшения начального интервала неопределенности, из которого следует количество вычислений.

6.4. Задания для самостоятельной работы

Задача 1

Найти минимум функции методом деления отрезка пополам в интервале при точности расчета:

$$f(x) = 2x^2 - 12x$$

Интервал неопределенности и требуемая точность:

$$[0; 10], \Delta L = 1$$

Задача 2

Найти минимум функции методом деления отрезка пополам в интервале при точности расчета:

$$f(x) = 6x^2 - 11x$$

Интервал неопределенности и требуемая точность:

$$[0; 10], \Delta L = 1$$

Задача 3

Найти минимум функции методом деления отрезка пополам в интервале при точности расчета:

$$f(x) = 8x^2 - 2x$$

Интервал неопределенности и требуемая точность:

$$[0; 10], \Delta L = 1$$

Задача 4

Найти минимум функции методом деления отрезка пополам в интервале при точности расчета:

$$f(x) = 25x^2 - 3x$$

Интервал неопределенности и требуемая точность:

$$[0; 10], \Delta L = 1$$

Задача 5

Найти минимум функции методом деления отрезка пополам в интервале при точности расчета:

$$f(x) = 13x^2 - 4x$$

Интервал неопределенности и требуемая точность:

$$[0; 10], \Delta L = 1$$

Задача 6

Найти минимум функции методом деления отрезка пополам в интервале при точности расчета:

$$f(x) = 43x^2 - 38x$$

Интервал неопределенности и требуемая точность:

$$[0; 10], \Delta L = 1$$

7. Практические математические модели в задачах оптимизации

Освоение методов решения задач оптимизации необходимо для формирования теоретических знаний при программировании в области экономики, маркетинга, финансов и т.д.. Современные информационные технологии предлагают множество программных продуктов и платформ в помощь экономистам, но для решения инженерных задач, все-таки наиболее оптимальным является выбор математического процессора MathCAD. Удобный интерфейс и математически грамотное отображение решаемых уравнений представляет проектировщику быстрый вариант интуитивно понятного результата.

7.1. Задачи практического применения методов оптимизации в программировании

Первая задача предлагает классический поиск угла α , при котором объем конуса, вырезанного из бумаги, будет максимальным.

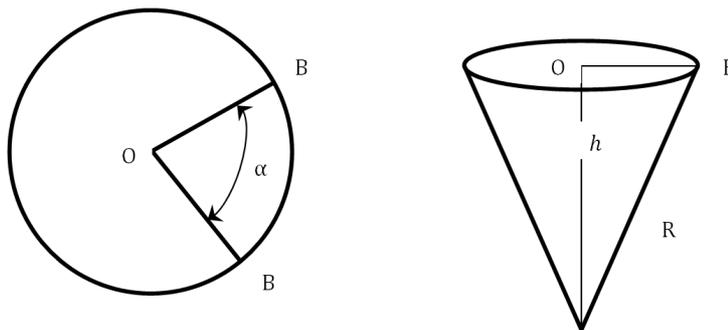


Рисунок 7.1. Выполнение конуса из листа

Задачу необходимо решить тремя способами:

- 1) использовать блок Given-Find;
- 2) использовать функцию solve;

3) использовать блок Given-Maximize.

Результат необходимо сравнить на графике функции.

Для начала необходимо обозначить общие для решения всеми способами моменты. Так, к ним будут относиться функции:

- расчета радиуса конуса;
- расчета высоты конуса;
- расчета объема конуса (целевая функция).

Общей переменной будет величина радиуса заготовки конуса.

Радиус как функция от угла α и радиуса заготовки R:

$$r(\alpha, R_{\text{ved}}) := R_{\text{ved}} \cdot \left(1 - \frac{\alpha}{2 \cdot \pi}\right)$$

Высота как функция от угла α и радиуса заготовки R:

$$h(\alpha, R_{\text{ved}}) := \sqrt{R_{\text{ved}}^2 - r(\alpha, R_{\text{ved}})^2}$$

Целевая функция - объем как функция от угла α и радиуса заготовки R:

$$V_{\text{ved}}(\alpha, R_{\text{ved}}) := \frac{1}{3} \cdot \pi \cdot r(\alpha, R_{\text{ved}})^2 \cdot h(\alpha, R_{\text{ved}})$$

Величина радиуса заготовки

$$R_{\text{ved}} := 1$$

Рисунок 7.2. Листинг задачи

Для того чтобы решить задачу с использованием блока Given-Find, необходимо рассчитать начальное приближение угла α в радианах

$$\begin{array}{l} \alpha := 3 \\ \text{Given } \alpha > 0 \quad \alpha < \pi \quad \frac{d}{d\alpha} V_{\text{ved}}(\alpha, R_{\text{ved}}) = 0 \quad \alpha_{\text{opt}} := \text{Find}(\alpha) \end{array}$$

Далее необходимо перевести полученное значение в радианы, для этого можно ввести дополнительную переменную $\text{grad} := 180/\pi$. На эту переменную нужно будет умножить начальное приближение.

$$\alpha_{\text{opt_drad}} := \alpha_{\text{opt}} \cdot \text{grad} \quad \alpha_{\text{opt_drad}} = 105.442$$

На основе полученного результата можно выполнить вычисления по целевой функции и получить результат.

$$V_{\max} := V_{\text{ved}}(\alpha_{\text{opt}}, R_{\text{ved}}) = 0.262$$

Для того чтобы выполнить символьное решение с использованием функции solve, необходимо вычислить значения первой производной объема по параметру α для функции без ограничения.

$$\frac{d}{d\alpha} V_{\text{ved}}(\alpha, R_{\text{ved}}) \text{ solve, } \alpha \rightarrow \begin{bmatrix} 2 \cdot \pi \\ \pi \cdot (\sqrt{2} + 2) \\ -\pi \cdot (\sqrt{2} - 2) \end{bmatrix} \quad \begin{array}{l} \alpha_{\text{opt1}} := 2 \cdot \pi = 6.283 \\ \alpha_{\text{opt2}} := \pi \cdot (\sqrt{2} + 2) = 10.726 \\ \alpha_{\text{opt3}} := -\pi \cdot (\sqrt{2} - 2) = 1.84 \end{array}$$

В результате будет получено три корня. Они будут являться значениями оптимального угла в градусах. Однако анализ результатов показывает, что первое и второе значения угла не могут быть практически реализованы, следовательно, оптимальным будет третий найденный угол.

$$\alpha_{\text{opt1_grad}} := \alpha_{\text{opt1}} \cdot \text{grad} = 360$$

$$\alpha_{\text{opt2_grad}} := \alpha_{\text{opt2}} \cdot \text{grad} = 614.558$$

$$\alpha_{\text{opt3_grad}} := \alpha_{\text{opt3}} \cdot \text{grad} = 105.442$$

Для того чтобы найти численное решение с использованием блока Given-Maximize, необходимо выполнить ряд операций.

$$V_{\text{ved2}}(\alpha_2, R_{\text{ved}}) := \frac{1}{3} \cdot \pi \cdot r(\alpha_2, R_{\text{ved}})^2 \cdot h(\alpha_2, R_{\text{ved}})^2$$

$$\alpha_2 := 3$$

$$\text{Given} \quad \alpha_2 > 0 \quad \alpha_2 < \pi \quad R_{\text{ved}} = 1$$

$$R_{\alpha_opt} := \text{Maximize}(V_{\text{ved}}, \alpha, R_{\text{ved}})$$

$$R_{\alpha_opt} = \begin{pmatrix} 1.84 \\ 1 \end{pmatrix}$$

$$R_{\text{ved3}} := R_{\alpha_opt_1} = 1$$

$$\alpha_3_opt := R_{\alpha_opt_0} = 1.84$$

Для того чтобы решить задачу графическим способом, необходимо построить график зависимости объема конуса от угла α через один градус и нанести точку с максимальным объемом ведра.

Минимальное и максимальное значение угла:

$$\alpha_{\text{min}} := 0 \quad \alpha_{\text{max}} := 2 \cdot \pi = 6.283$$

Радиус заготовки:

$$R_{\text{ved}} = 1$$

Расчетный шаг по углу:

$$\Delta\alpha := 0.01 \cdot \pi = 0.031$$

Количество точек расчета:

$$N := \frac{\alpha_{\text{max}} - \alpha_{\text{min}}}{\Delta\alpha}$$

Дискретный аргумент:

$$i := 0..N$$

Вектор-угол расчетных углов:

$$\alpha_i := \alpha_{\text{min}} + \Delta\alpha \cdot i$$

Вектор радиуса конуса:

$$r_{\text{vec}}(\alpha, R_{\text{ved}}) := \overrightarrow{\left[R \cdot \left(1 - \frac{\alpha}{2 \cdot \pi} \right) \right]}$$

Вектор высоты конуса:

$$h_{\text{vec}}(\alpha, R_{\text{ved}}) := \sqrt{R_{\text{ved}}^2 - r_{\text{vec}}(\alpha, R_{\text{ved}})^2}$$

Вектор оптимизационной функции:

$$V_{\text{opt}}(\alpha, R_{\text{ved}}) := \overrightarrow{\left(\frac{1}{3} \cdot \pi \cdot r_{\text{vec}}(\alpha, R_{\text{ved}})^2 \cdot h_{\text{vec}}(\alpha, R_{\text{ved}})^2 \right)}$$

В результате получится график:

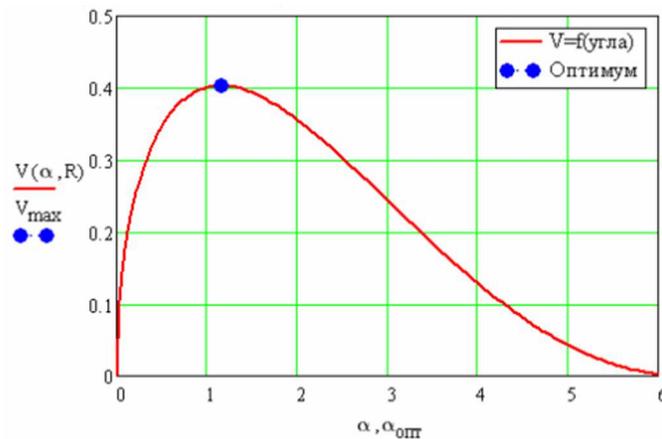


Рисунок 7.3. Решение задачи графическим способом

Вторая задача предлагает рассчитать вырезку из картонки для коробки. Отношение высоты вырезаемого квадрата к высоте всей картонки.

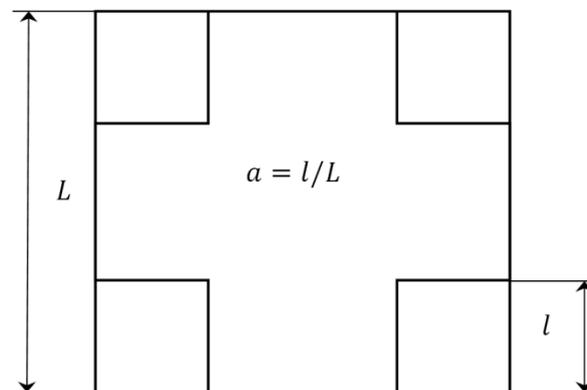
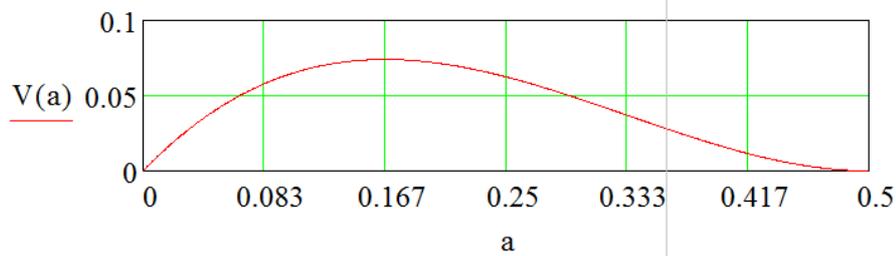


Рисунок 7.4. Выкройка коробки из картонки

$$V(a) := [a \cdot (1 - 2 \cdot a)^2]$$



$$\left(\begin{array}{c} \text{Minimum} \\ \text{aop} \end{array} \right) := \frac{d}{da} V(a) \text{ solve, } a \rightarrow \left(\begin{array}{c} \frac{1}{2} \\ \frac{1}{6} \end{array} \right)$$

$$\text{aop} = 0.167$$

Рисунок 7.5. Решение задачи

Третья задача предлагает рассчитать вырезку из картонки для коробки. Отношение высоты вырезаемого квадрата к высоте всей картонки.

$$\left(\begin{array}{c} R \\ H \\ L \end{array} \right) := \left(\begin{array}{c} 1 \\ 1 \\ 1 \end{array} \right) \text{ m}$$

$$V(R, H, L) := \left| \begin{array}{l} \text{"Полусфера"} \\ \frac{4}{3} \cdot \pi \cdot R^3 \\ 2 \end{array} \right| + \left| \begin{array}{l} \text{"Цилиндр"} \\ \pi \cdot R^2 \cdot L \end{array} \right| + \left| \begin{array}{l} \text{"Конус"} \\ \frac{1}{3} \cdot \pi \cdot R^2 \cdot H \end{array} \right|$$

$$S(R, H, L) := \left| \begin{array}{l} \text{"Полусфера"} \\ 4 \cdot \pi \cdot R^2 \\ 2 \end{array} \right| + \left| \begin{array}{l} \text{"Цилиндр"} \\ 2\pi \cdot R \cdot L \end{array} \right| + \left| \begin{array}{l} \text{"Конус"} \\ \pi R \sqrt{R^2 + H^2} \end{array} \right|$$

Given
 $V(R, H, L) = 3m^3 \quad R \geq 0 \quad H \geq 0 \quad L \geq 0$

$$\begin{pmatrix} R \\ H \\ L \end{pmatrix} := \text{Maximize}(S, R, H, L) = \begin{pmatrix} 1.127 \\ 0 \\ 0 \end{pmatrix} m \quad S(R, H, L) = 11.976 m^2$$

Given
 $V(R, H, L) = 3m^3 \quad S(R, H, L) = 11m^2$

$$\begin{pmatrix} R \\ H \\ L \end{pmatrix} := \text{Find}(R, H, L) = \begin{pmatrix} 1.05 \\ 0.276 \\ 0.073 \end{pmatrix} m$$

"Бак" = "Полусфера" + "Цилиндр" + "Конус" solve, L → solve, L → solve, 0.0734425727723475·m

$$V = \begin{pmatrix} \frac{4}{3} \cdot \pi \cdot R^3 \\ \pi \cdot R^2 \cdot L \\ \frac{1}{3} \cdot \pi \cdot R^2 \cdot H \end{pmatrix}$$

$V := 3m^3$

$$V := 3m^3$$

$$S(R, H) := \begin{cases} L \leftarrow \frac{3V - 2\pi R^3 - \pi R^2 H}{3\pi R^2} \\ \text{if} \left(L \geq 0, \frac{1}{2} 4\pi R^2 + 2\pi R \cdot L + \pi R \sqrt{H^2 + R^2}, 0 \right) \end{cases}$$

Рисунок 7.6. решение задачи для вырезки коробки

7.2. Задания для самостоятельной работы

Задача 1

Найти минимум функции методом деления отрезка пополам в интервале при точности расчета:

$$f(x) = 2x^2 - 12x$$

Интервал неопределенности и требуемая точность:

$$[0; 10], \quad \Delta L = 1$$

Задача 2

Найти минимум функции методом деления отрезка пополам в интервале при точности расчета:

$$f(x) = 6x^2 - 11x$$

Интервал неопределенности и требуемая точность:

$$[0; 10], \Delta L = 1$$

Задача 3

Найти минимум функции методом деления отрезка пополам в интервале при точности расчета:

$$f(x) = 8x^2 - 2x$$

Интервал неопределенности и требуемая точность:

$$[0; 10], \Delta L = 1$$

Задача 4

Найти минимум функции методом деления отрезка пополам в интервале при точности расчета:

$$f(x) = 25x^2 - 3x$$

Интервал неопределенности и требуемая точность:

$$[0; 10], \Delta L = 1$$

Задача 5

Найти минимум функции методом деления отрезка пополам в интервале при точности расчета:

$$f(x) = 13x^2 - 4x$$

Интервал неопределенности и требуемая точность:

$$[0; 10], \Delta L = 1$$

Задача 6

Найти минимум функции методом деления отрезка пополам в интервале при точности расчета:

$$f(x) = 43x^2 - 38x$$

Интервал неопределенности и требуемая точность:

$$[0; 10], \Delta L = 1$$

8. Оптимизационные задачи в теории колебаний

Решение задач колебаний всегда является проблемой при решении вопросов оптимизации. Производятся процессы выявления либо параметров, влияющих на точность колебательных процессов, либо определяются величины и характеристики технических объектов, при изменении которых происходит снижение колебательных и вибрационных нагрузок. Особым разделом в науке стоит вопрос моделирования колебательных процессов в транспортных системах. Колебательные и вибрационные нагрузки при воздействии на человека приводят к крайне нежелательным последствиям, особенно при выполнении оператором рабочих функций. Ученые, посвятившие свои труды изучению колебаний и вибрации на транспорте, уже много лет успешно применяют современные информационные технологии, при этом, все-таки, наиболее оптимальным является выбор математического процессора MathCAD. Выполнение амплитудно-частотных и фазо-частотных характеристик система MathCAD производит с использованием встроенного функционала. И, именно после грамотного исследования частотных характеристик, появляется возможность выбора оптимальных параметров упруго-вязких элементов транспортных систем.

8.1. Оптимизационные задачи динамики. Вертикальные колебания технических систем.

Упрощенная модель вертикальных перемещений машины, у которой имеется кабина и кузов, представлена на рисунке 8.1. Гашение колебаний производится при помощи пружин и амортизаторов, причем задача подбора характеристик жесткости пружин и коэффициентов сопротивления амортизаторов – проблема, над которой бьются уже много десятилетий ученые автомобилестроители многих стран мира.

Оптимизация подбора характеристик упруго-вязкой подвески должна приводить к комфорту работы оператора транспортного средства и приятному перемещению пассажиров. Система, показанная на рисунке 8.1 демонстрирует модель для расчета и построения математических дифференциальных уравнения колебательных процессов.

Формулы 8.1 классические колебательные системы.

$$\begin{cases} m_1 * z_1'' + b_1 * z_1' + b_2 * (z_1' - z_2') + c_1 * z_1 + c_2 * (z_1 - z_2) = b_1 * \eta \\ m_2 * z_2'' + b_2 * (z_2' - z_1') + c_2 * (z_2 - z_1) = 0 \end{cases} \quad (8.1)$$

где m_1 – масса ходовой части транспортного средства,

m_2 – масса кабины пилота,

c_1 и c_2 – жесткость пружин подвесок,

b_1 и b_2 – коэффициент сопротивления амортизаторов,

$\eta(t)$ – возмущение со стороны пути,

z_1, z_1', z_1'' – перемещение, скорость и ускорение первого тела,

z_2, z_2', z_2'' – перемещение, скорость и ускорение второго тела.

MathCAD предлагает встроенный процессор для решения задачи Коши. После приведения системы уравнений к нормальной форме системы дифференциальных уравнений первого порядка.

$$\begin{cases} \dot{t}_0 = z \\ \dot{t} = z' \end{cases} \quad (8.2)$$

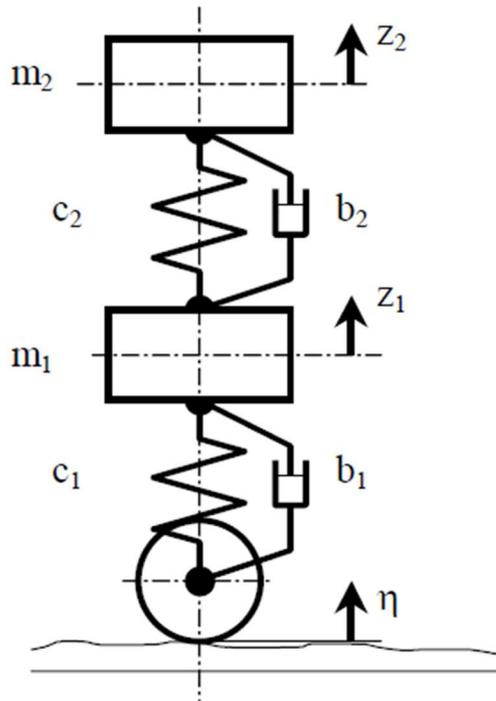


Рисунок 8.1. Расчетная двухмассовая схема колебаний машины

Выражаются и рассчитываются z_1'' и z_2'' .

$$\left\{ \begin{array}{l} t_0' = z_1' \\ t_1' = -\frac{1}{m_1} [c_1 * (z_1 - \eta(t)) + b_1 * (z_1' - \eta'(t)) + c_2 * (z_1 - z_2) + b_2 * (z_1' - z_2')] \\ t_0' = z_2' \\ t_1' = -\frac{1}{m_2} [c_2 * (z_2 - z_1) + b_2 * (z_2' - z_1')] \end{array} \right. \quad (8.3)$$

Решение уравнений в программной среде MathCAD производится после замены обозначений: $x_0 = z_1, x_1 = z_1', x_2 = z_2, x_3 = z_2'$.

Задавая нулевые начальные условия, с использованием встроенного функционала Rkadapt в MathCAD методом Рунге-Кутты производится расчет и построение частотных характеристик колебательных процессов.

Исходные данные

$m1 := 8.82$	$m2 := 25.823$	масса, т
$c1 := 7000$	$c2 := 2600$	жесткость, кН/м
$b1 := 60$	$b2 := 125$	демпфирование, кН.с/м
$V := 20$		скорость, м/с
$L := 25$		длина неровности, м
$A := 0.005$		амплитуда неровности, м
$\omega := V \cdot \frac{2 \cdot \pi}{L}$	$\omega = 5.027$	частота возмущения

Функции возмущения со стороны пути

$$\eta(t) := A \cdot \sin(\omega \cdot t) \quad d\eta(t) := A \cdot \omega \cdot \cos(\omega \cdot t)$$

Параметры интегрирования

$$t0 := 0 \quad t1 := 4 \quad N := 200 \quad h := \frac{t1 - t0}{N} \quad h = 0.02$$

Начальные условия $x := \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

Система дифференциальных уравнений

$$D(t, x) := \begin{bmatrix} x_1 \\ \frac{-1}{m1} \cdot [c1 \cdot (x_0 - \eta(t)) + b1 \cdot (x_1 - d\eta(t)) + c2 \cdot (x_0 - x_2) + b2 \cdot (x_1 - x_3)] \\ x_3 \\ \frac{-1}{m2} \cdot [c2 \cdot (x_2 - x_0) + b2 \cdot (x_3 - x_1)] \end{bmatrix}$$

Решение системы дифференциальных уравнений $Z := Rkadapt(x, t0, t1, N, D)$

Рисунок 8.2. Решение системы дифференциальных уравнений колебательного процесса

Решение в виде матрицы представлено на рисунке 8.3.

	0	1	2	3	4
0	0	0	0	0	0
1	0.02	$5.184 \cdot 10^{-5}$	$5.936 \cdot 10^{-3}$	$1.648 \cdot 10^{-6}$	$2.742 \cdot 10^{-4}$
2	0.04	$2.514 \cdot 10^{-4}$	0.014	$1.683 \cdot 10^{-5}$	$1.428 \cdot 10^{-3}$
3	0.06	$6.09 \cdot 10^{-4}$	0.021	$6.588 \cdot 10^{-5}$	$3.684 \cdot 10^{-3}$
4	0.08	$1.083 \cdot 10^{-3}$	0.026	$1.701 \cdot 10^{-4}$	$6.927 \cdot 10^{-3}$
5	0.1	$1.61 \cdot 10^{-3}$	0.027	$3.459 \cdot 10^{-4}$	0.011
6	0.119	$2.137 \cdot 10^{-3}$	0.026	$6.023 \cdot 10^{-4}$	0.015
7	0.139	$2.628 \cdot 10^{-3}$	0.024	$9.399 \cdot 10^{-4}$	0.019
8	0.159	$3.073 \cdot 10^{-3}$	0.021	$1.353 \cdot 10^{-3}$	0.023
9	0.179	$3.476 \cdot 10^{-3}$	0.019	$1.833 \cdot 10^{-3}$	0.026
10	0.199	$3.851 \cdot 10^{-3}$	0.018	$2.366 \cdot 10^{-3}$	0.028
11	0.219	$4.205 \cdot 10^{-3}$	0.017	$2.94 \cdot 10^{-3}$	0.03
12	0.239	$4.543 \cdot 10^{-3}$	0.017	$3.54 \cdot 10^{-3}$	0.031
13	0.259	$4.859 \cdot 10^{-3}$	0.015	$4.154 \cdot 10^{-3}$	0.031
14	0.279	$5.145 \cdot 10^{-3}$	0.013	$4.765 \cdot 10^{-3}$	0.03
15	0.299	$5.39 \cdot 10^{-3}$	0.011	$5.359 \cdot 10^{-3}$...

Рисунок 8.3. Решение задачи колебаний в виде матрицы

Вид решения задачи методом табулирования удобен в информационных системах своей наглядностью, выполняются графики по этим значениям, рисунки 8.4 и 8.5.

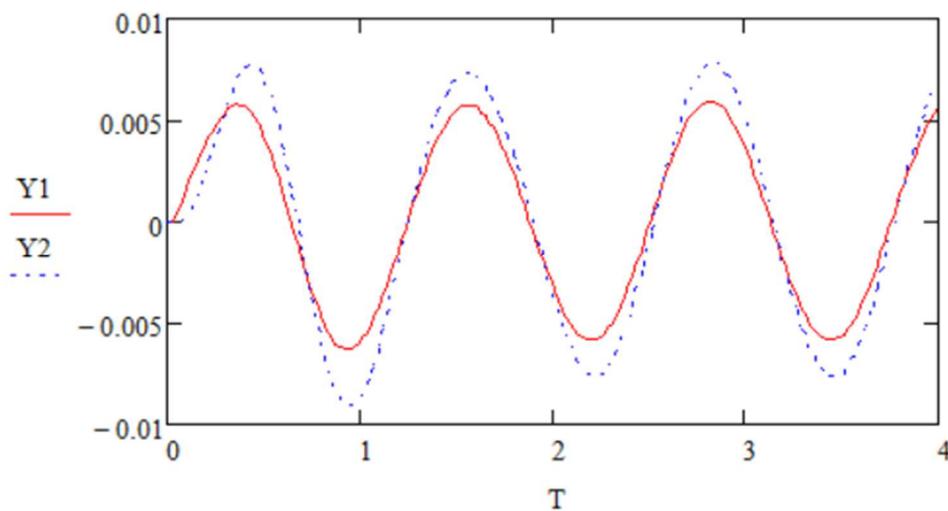


Рисунок 8.4. Графики частотных характеристик

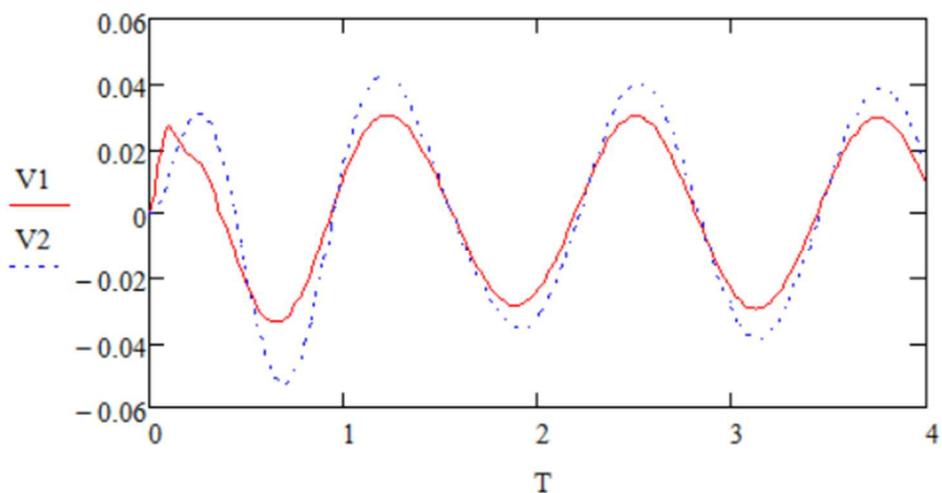


Рисунок 8.5. Графики частотных характеристик

Результаты фазовых характеристик показаны на рисунке 8.6.

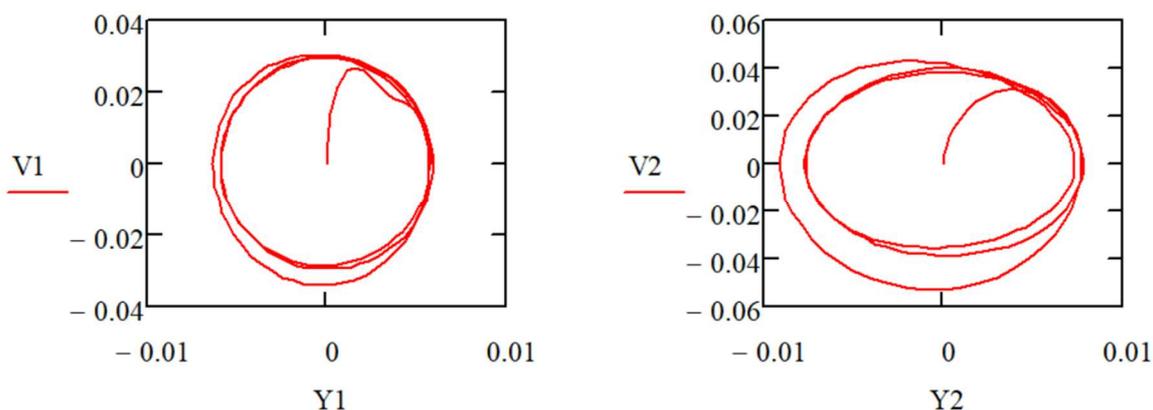


Рисунок 8.6 Графики фазовых характеристик.

Рисунок 8.7 представляет собой отображение документа расчетного процесса при выполнении задачи расчетов колебаний транспортных средств. Несмотря на простоту документа в математическом процессоре, MathCAD с использованием встроенных формульных зависимостей предоставляет ученому в кратчайшее время построить и решить сложную математическую модель оптимизационных задач колебательных процессов. Амплитуда колебаний, выступая в прямой зависимости от возмущающего процесса и скорости движения при помощи построенных графиков позво-

ляет выявить оптимальные решения определения таких характеристик как жесткости пружин и коэффициенты сопротивления амортизаторов.

Формирование выходных данных

$T := Z^{(0)}$		время
$Y1 := Z^{(1)}$	$V1 := Z^{(2)}$	перемещение и скорость первого тела
$Y2 := Z^{(3)}$	$V2 := Z^{(4)}$	перемещение и скорость второго тела

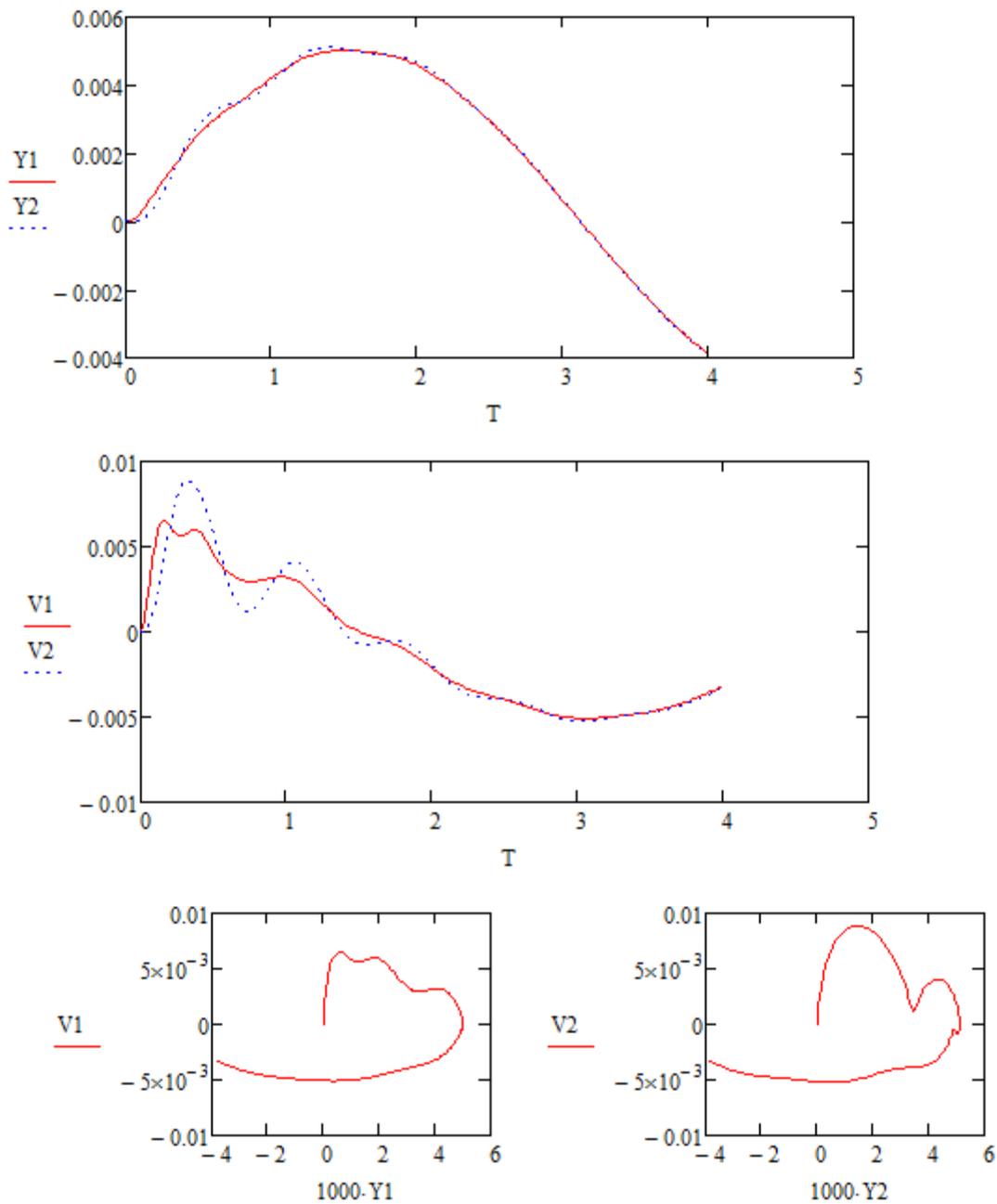


Рисунок 8.7. Графики характеристик колебательных процессов

Результаты полученных математических испытаний могут поставить в тупик ученого (как, например графики рисунка 8.7) и заставить производить подбор характеристик упруго-вязких подвесок для поиска наиболее оптимального решения задачи.

8.2. Расчет дифференциальных уравнений колебательных процессов и получение частотных характеристик

Возможность рассчитать системы дифференциальных уравнений в программе MathCAD позволяет определить оптимальные характеристики упруго-вязких элементов колебательных систем. В первую очередь системы дифференциальных уравнений колебательных процессов приводятся к форме Коши. Представление уравнений в компьютерной программе системы MathCAD не представляет труда и интуитивно понятна ученому, привыкшему решать задачи оптимизации в колебательных параметрах. Формальная процедура математического процессора великолепно отображает матричное решение задачи. Качество и точность решения зависит не от работы программиста, а от тщательного описания проблематики ученого в сфере колебаний технических транспортных систем. И, тем не менее, процессор MathCAD требует внимательности при проведении расчетов при введении переменных, их размерности и обозначений.

Исходные данные

$m1 := 9$ $j1 := 12$ $c1 := 3040$ $b1 := 30$ $a1 := 1.5$
 $m2 := 57$ $j2 := 70$ $c2 := 2660$ $b2 := 100$ $a2 := 3.725$
 $v := 30$

Неровность пути

$\underline{\omega} := 25$ $n0 := 0.005$ $\omega := 2 \cdot \pi \cdot \frac{v}{1}$
 $n1(t) := n0 \cdot \sin(\omega \cdot t)$ $n11(t) := \omega \cdot n0 \cdot \cos(\omega \cdot t)$
 $n2(t) := n1\left(t - \frac{2 \cdot a1}{v}\right)$ $n22(t) := n11\left(t - \frac{2 \cdot a1}{v}\right)$
 $n3(t) := n1\left(t - 2 \cdot \frac{a2}{v}\right)$ $n33(t) := n11\left(t - 2 \cdot \frac{a2}{v}\right)$
 $n4(t) := n1\left(t - 2 \cdot \frac{a2 + a1}{v}\right)$ $n44(t) := n11\left(t - 2 \cdot \frac{a2 + a1}{v}\right)$

Рисунок 8.8. Программа определения исходных данных.

После введения данных (рисунок 8.8), определения жесткости пружин и коэффициентов сопротивления амортизаторов, заносятся величины масс подрессоренной и не подрессоренной частей машин. Манипулируя введенными данными опять же решаются задачи оптимизации. Как и принято, решением является графическое представление частотных характеристик, внешний вид, которых позволяет ученым выбрать оптимальные значения характеристик упруго-вязких опор движителей машин.

Уравнение

$$\underline{H}(t,x) := \begin{bmatrix}
 x_1 \\
 \frac{1}{m1} \cdot \left[b1 \cdot (n11(t) + n22(t) - 2 \cdot x_1) + c1 \cdot (n1(t) + n2(t) - 2 \cdot x_0) \dots \right. \\
 \left. + b2 \cdot (x_5 - x_1 + x_{11} \cdot a2) + c2 \cdot (x_4 - x_0 + x_{10} \cdot a2) \right] \\
 x_3 \\
 \frac{1}{m1} \cdot \left[b1 \cdot (n33(t) + n44(t) - 2 \cdot x_3) + c1 \cdot (n3(t) + n4(t) - 2 \cdot x_2) \dots \right. \\
 \left. + b2 \cdot (x_5 - x_3 - x_{11} \cdot a2) + c2 \cdot (x_4 - x_2 - x_{10} \cdot a2) \right] \\
 x_5 \\
 \frac{1}{m2} \cdot \left[b2 \cdot (x_1 + x_3 - 2 \cdot x_5) + c2 \cdot (x_0 + x_2 - 2 \cdot x_4) \right] \\
 x_7 \\
 \frac{a1}{j1} \cdot \left[b1 \cdot (n11(t) - n22(t) - 2 \cdot x_7 \cdot a1) + c1 \cdot (n1(t) - n2(t) - 2 \cdot x_6 \cdot a1) \right] \\
 x_9 \\
 \frac{a1}{j1} \cdot \left[b1 \cdot (n33(t) - n44(t) - 2 \cdot x_9 \cdot a1) + c1 \cdot (n3(t) - n4(t) - 2 \cdot x_8 \cdot a1) \right] \\
 x_{11} \\
 \frac{a1}{j2} \cdot \left[b2 \cdot (x_1 - x_3 - 2 \cdot x_{11} \cdot a2) + c2 \cdot (x_0 - x_2 - 2 \cdot x_{10} \cdot a2) \right]
 \end{bmatrix}$$

Рисунок 8.9. Матричное решение задачи

Матричное представление системы дифференциальных уравнений в MathCAD, показанное на рисунке 8.9 прекрасно демонстрирует возможности информационного математического процессора как программного продукта, непосредственно предназначенного для ученых, производящих оптимизационные вычисления без возможности затрат времени на построение программных алгоритмов.

Далее производится решение системы уравнений, причем, опять же важным выступает необходимость скрупулезного описания параметрических данных, вводимых при определении характеристик колебательных

процессов. Даже небольшая ошибка может привести к весьма плачевным результатам.

Решение системы уравнений показано на рисунке 8.10. Используется адаптивный метод с автоматическим подбором шага.

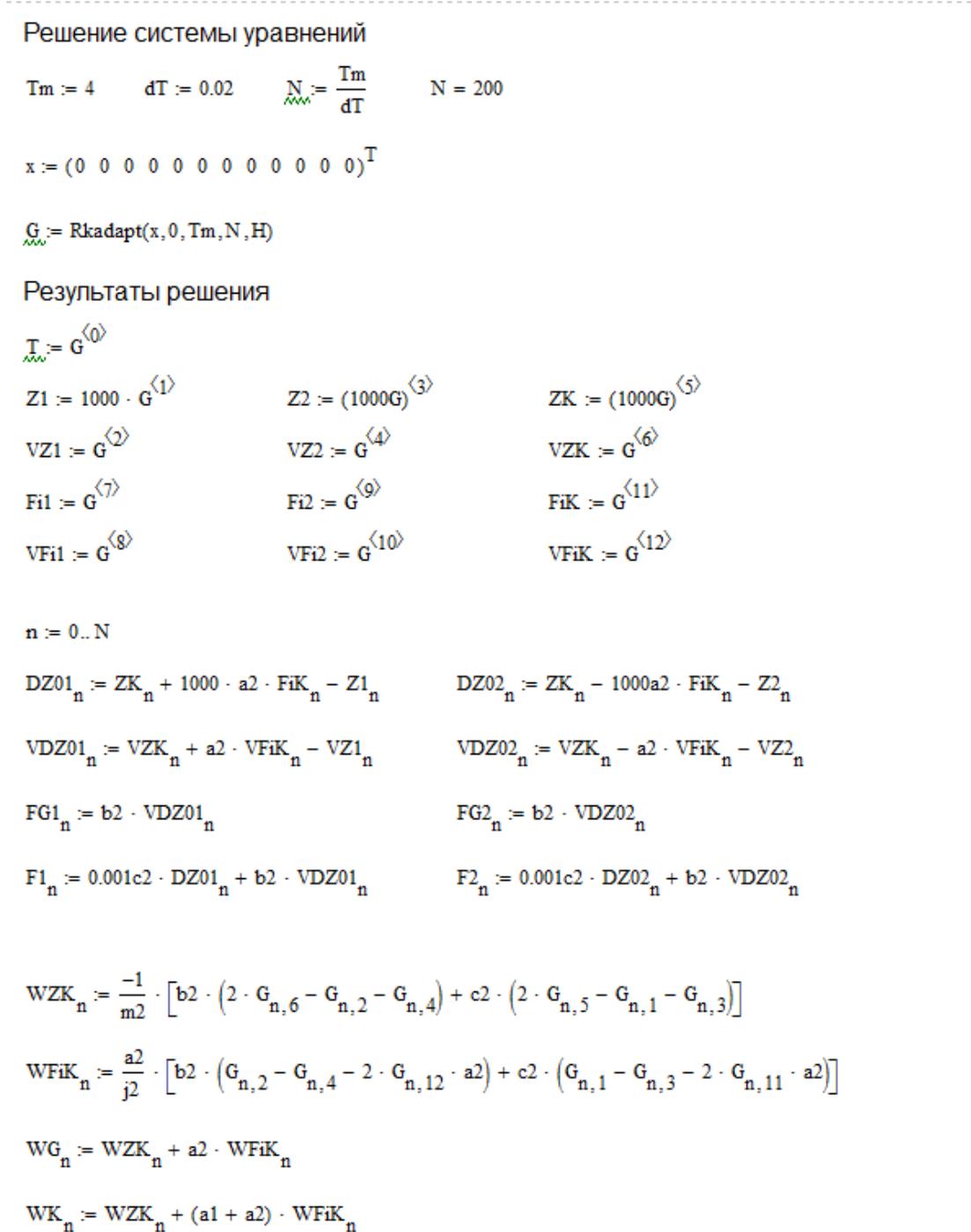


Рисунок 8.10. Решение задачи

В матрице рисунка 8.9 формируется решение задачи, которое опять же должно быть графическим. Для удобства преобразования результатов и построения графиков вводятся новые переменные. В массивы данных используются индексы и векторизация вычислений.

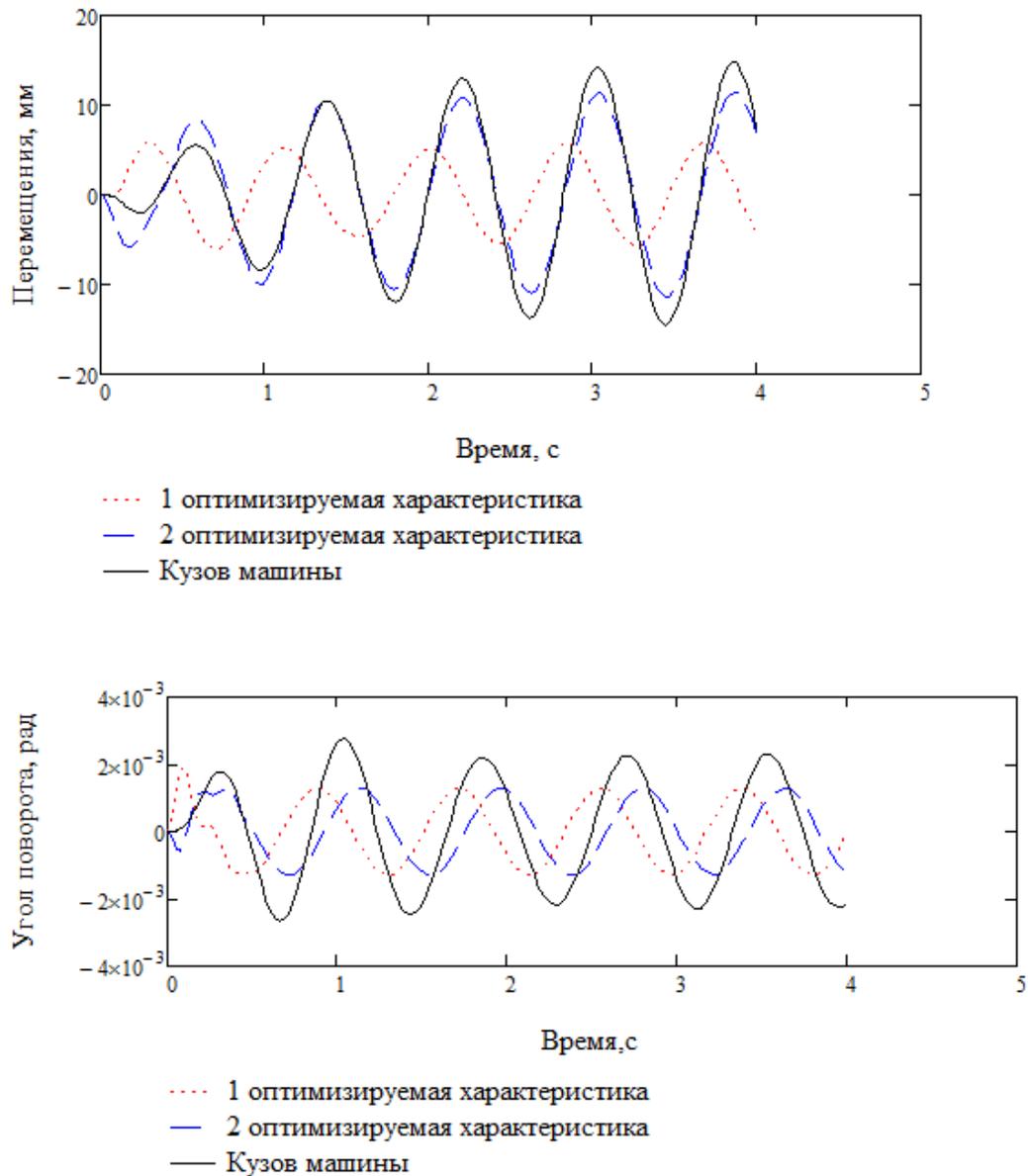


Рисунок 8.11. Частотные характеристики решения задачи и выполнения выбора оптимизационных данных упруго-вязкой подвески

Оптимизация частотных характеристик проводится методом Гаусса-Зейделя, когда главные критерии (в данном случае жесткость упругого элемента и коэффициент сопротивления амортизирующего) выбираются с учетом ограничений. Оптимизация выполняется в последовательности: четко определяется число коэффициентов для поиска результата, рассматривается и выбирается число неудачных попыток, начальное приближение вектора определяется инженерным методом (или просто устанавливается равным единице), после изменения коэффициентов изучаются характер изменения амплитудно-частотных характеристик. Например, на рисунке 8.10 после изменения коэффициентов происходит сглаживание колебательного процесса (нет разброса амплитуд частотных характеристик), следовательно выбор коэффициентов проведен верно. Полная оптимизация колебательных процессов будет выполнена после подбора характеристик упругих элементов таким образом, что улучшение целевой функции приведет к колебаниям в рамках гармоник в допустимых пределах.

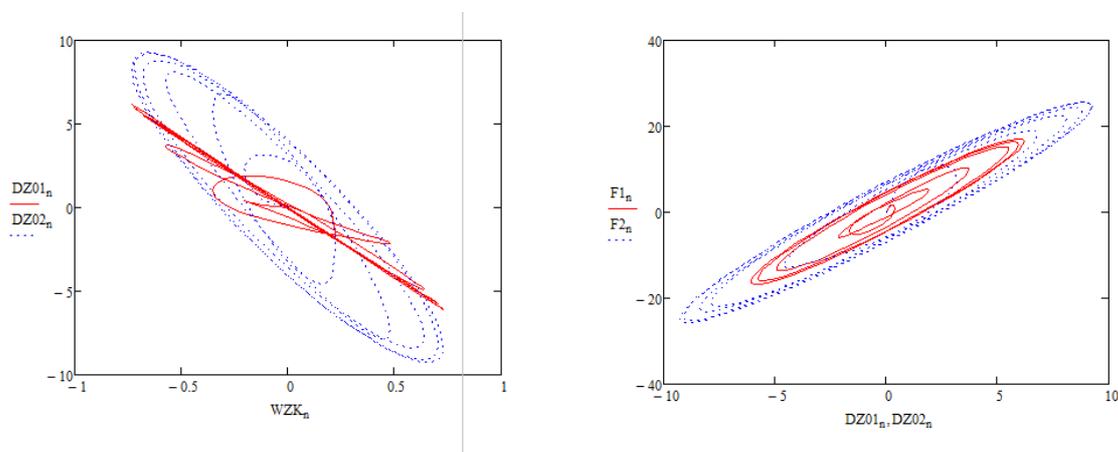


Рисунок 8.12. Фазовые характеристики решения задачи и выполнения выбора оптимизационных данных упруго-вязкой подвески

Алгоритм получения оптимальных значений по частотным характеристикам работает по методу релаксации переменных, когда процесс разбивается на этапы и на каждом из этапов оптимизация проводится по одной или нескольким группам переменных. Когда осуществляется переход

от предыдущего этапа к следующему выполняется замена переменных и групповая релаксация продолжается. В первой группе расчетов все коэффициенты принимаются равными единице, следовательно релаксация начинается со второй группы и процесс происходит до улучшения целевой функции. На рисунке 8.12 показано, что фазово-частотные левого рисунка демонстрируют отклонения от нормы, а после поправки коэффициентов получается исправленная картина правого графика.

8.3. Расчет контактных напряжений в контактных парах

Определение контактных сил методами математического моделирования в таких технических системах как контакт железнодорожного колеса и рельса позволяет выявить оптимальные величины нагружения вагона при выполнении ими рабочих операций. Комплексная математическая модель системы экипаж - тяговый привод - путь, охватывающая локомотивы с трехосными тележками, двухступенчатым рессорным подвешиванием, упругой связью колесных пар с рамой тележки и тележек с кузовом с учетом всех основных нелинейностей, присущих реальным локомотивам (решение показано на рисунке 8.13).

Исходные данные программы

$mas := 20.1$ Масса тележки, т
 $Iz := 26$ момент инерции тележки относительно вертикальной оси, т.м²
 $P := 112.5$ нагрузка от колеса на рельсы, кН $g = 9.807 \frac{m}{s^2}$
 $mk := 4 \cdot \frac{P}{g} - mas = \blacksquare$ приведенная масса кузова, т
 $a := 1.5$ полубаза тележки, м
 $ak := 3.725$ полубаза кузова, м
 $bs := 0.8$ половина расстояние между кругами катания, м
 $r0 := 0.625$ номинальный радиус бандажа по кругу катания, м
 $\gamma := 0.05$ наклон образующей бандажа
 $Car := 2000$ угловая жесткость связей тележки с кузовом, кН .м/рад
 $B := 0$ угловое демпфирование тележки, кН .м.с/рад
 $Cr := 80000$ жесткость рельса, кН/м
 $Koe := 235 \cdot P - 2.4 \cdot P^2 + 0.01P^3 = 1.03 \times 10^4$ коэффициент крипа по Мюллеру, кН
 $fm := 0.25$ максимальный коэффициент трения скольжения
 $Ks := \frac{Koe}{P \cdot fm} = 366.25$ относительный коэффициент крипа
 $Vk := 80$ скорость движения, км/ч
 $Va := \frac{Vk}{3.6} = 22.222$ м/с
 $Rkr := 600$ радиус кругой кривой, м
 $\rho1 := \frac{1}{Rkr}$ кривизна пути в круговой кривой, м⁻¹ $\rho1 = 0.0017$
 $hR := 0.100$ возвышение наружного рельса, м

Рисунок 8.13. Введение исходных данных

Скорости скольжения колеса по рельсу

$$u_x(j, \eta, \Psi, d\Psi, s) := \frac{1}{r_0} \cdot [\eta + (-1)^{j+1} \cdot \Psi \cdot a] \cdot \gamma + b_s \cdot \left(\frac{1}{V} \cdot d\Psi - \rho(s) \right)$$

$$u_y(j, d\eta, \Psi, d\Psi, s) := \Psi - \frac{1}{V} \cdot [d\eta + (-1)^{j+1} \cdot a \cdot (d\Psi + 0.5 \cdot \Psi)] + (-1)^{j+1} \cdot a \cdot \rho(s)$$

Силы трения колеса по рельсу

$$F_s(j, \eta, d\eta, \Psi, d\Psi, s) := \frac{K}{\sqrt{1 + Ks^2 \cdot (u_x(j, \eta, \Psi, d\Psi, s)^2 + u_y(j, \eta, \Psi, d\Psi, s)^2)}}$$

$$X(j, \eta, d\eta, \Psi, d\Psi, s) := F_s(j, \eta, d\eta, \Psi, d\Psi, s) \cdot u_x(j, \eta, \Psi, d\Psi, s)$$

$$Y(j, \eta, d\eta, \Psi, d\Psi, s) := F_s(j, \eta, d\eta, \Psi, d\Psi, s) \cdot u_y(j, \eta, \Psi, d\Psi, s)$$

Направляющая сила

$$Y_{ns}(\eta, s) := Cr \cdot \left[\eta + \frac{1}{2} \cdot (|\eta - \sigma(s)| - |\eta + \sigma(s)|) \right]$$

Символьный вектор, определяющий правые части уравнений в форме Коши

$$H(t, y) := \begin{bmatrix} y_1 \\ \frac{1}{mas} \cdot \left[(mas + mk) \cdot An(V \cdot t) + 2 \cdot Y(1, y_0, y_1, y_2, y_3, V \cdot t) \dots \right. \\ \left. + 2 \cdot Y(2, y_0, y_1, y_2, y_3, V \cdot t) \dots \right. \\ \left. + (-Yn(y_0 + a \cdot y_2, V \cdot t) - Yn(y_0 - a \cdot y_2, V \cdot t)) \right] \\ y_3 \\ \frac{1}{Iz} \cdot \left[C \cdot (ak \cdot \rho(V \cdot t) - y_2) - B \cdot y_3 \dots \right. \\ \left. + 2bs \cdot (-X(1, y_0, y_1, y_2, y_3, V \cdot t) - X(2, y_0, y_1, y_2, y_3, V \cdot t)) \dots \right. \\ \left. + 2a \cdot (Y(1, y_0, y_1, y_2, y_3, V \cdot t) - Y(2, y_0, y_1, y_2, y_3, V \cdot t)) \dots \right. \\ \left. + a \cdot (Yn(y_0 + a \cdot y_2, V \cdot t) - Yn(y_0 - a \cdot y_2, V \cdot t)) \right] \end{bmatrix}$$

Рисунок 8.14. Введение исходных данных

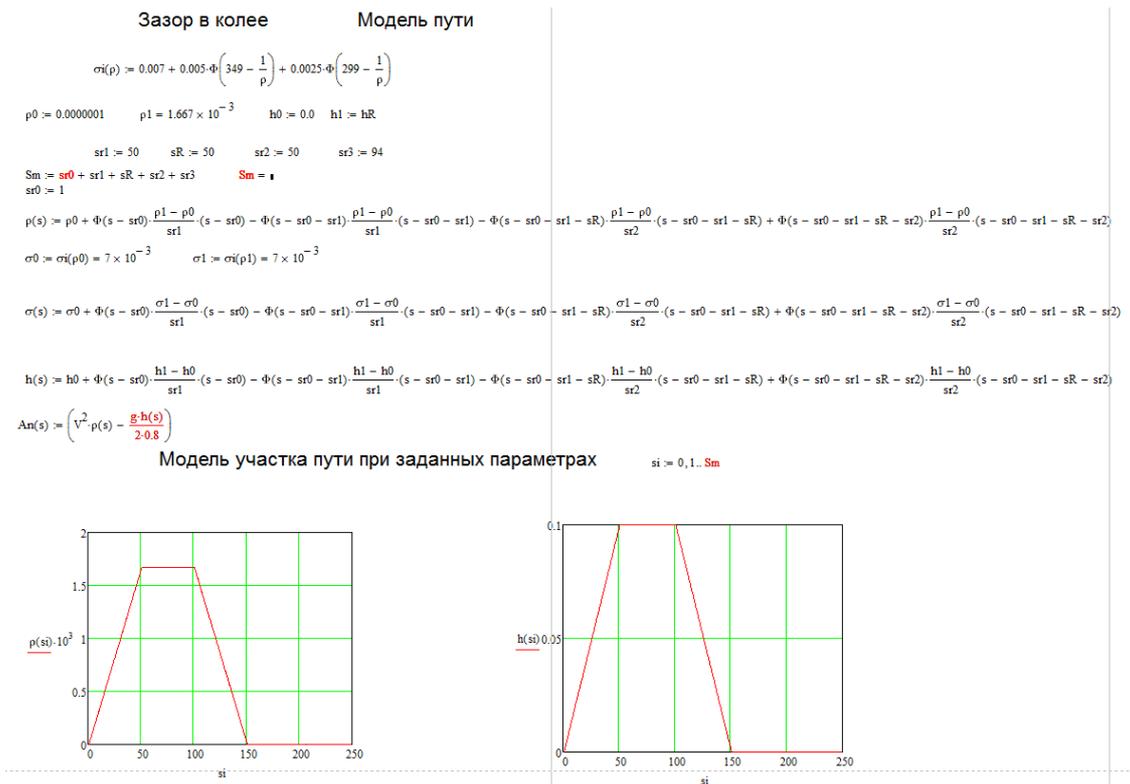


Рисунок 8.15. Получение модели участка пути по заданным параметрам

Особенностями проведенных вычислений явилось то, что графические решения дифференциальных уравнений и определения по графикам оптимальных коэффициентов является визитной карточкой математического процессора MathCAD. Для получения необходимых данных можно и не сохранять промежуточные итоги. Инженер, работающий в исследуемой области просто проводя манипуляции с исходными данными по виду изменения графических характеристик сразу же принимает оптимальное решение и фиксирует уже результирующие допустимые данные расчетов. Таким образом, компьютерные расчеты вполне результативно заменяют натуральный эксперимент.

Заключение

В представленном методическом пособии рассмотрен теоретический материал и практические задания для выполнения лабораторных работ курса автоматизированного проектирования и моделирования математических процессов. Путем синтеза структурированных моделей и непосредственного рабочего процесса по созданию модулей решения поставленных задач от создания и постановки проблематики вопроса до проработки расчетной, отчетной и конструкторской документации при помощи мощного редактора математического процессора MathCAD.

После изучения теоретического материала и проведения практических занятий студентам предложено самостоятельное выполнение полного курса оптимизационных примеров по автоматизации решения одного из алгоритмов линейного или нелинейного программирования. Студент должен разработать постановку задачи, осмыслить и выполнить полный набор отчетной документации по исследовательской работе математической задачи.

Выпускник направления 09.04.02 "Информационные системы и технологии" должен быть готов работать в любом направлении современного производства и выполнять разносторонние задачи: от математического моделирования до решения конструкторских инженерных задач.

Библиографический список

1. Абчук В.А. Справочник по исследованию операций / В.А. Абчук. - Москва: Воениздат, 1979. – 240 с.
2. Банди Б. Методы оптимизации. Вводный курс: Пер. с англ. / Б. Банди. – Москва: Радио и связь, 1988. – 128 с.
3. Вендеров А.М. CASE технологии. Современные методы и средства проектирования информационных систем / А.М. Вендеров. – Москва: Финансы и статистика, 2005. – 175 с. – ISSN: 5-279-01979-8
4. Данциг Дж. Линейное программирование: Пер. с англ. / Дж. Данциг – Москва: Прогресс. 1966. – 600 с.
5. Корн Г., Корн Т. Справочник по математике для научных работников и инженеров. Определения, теоремы, формулы / Г. Корн, Т. Корн. . – Москва: Наука, 1970. – 720 с.
6. Куклина И.Г., Куклина А.С., Молев М.Ю., Шишмакова Т.В. Автоматизация и информатизация в современных исследованиях / И.Г. Куклина, А.С. Куклина, М.Ю. Молев, Т.В. Шишмакова // Лучшая исследовательская работа 2022: материалы статей VII международного научно-исследовательского конкурса. – Пенза, 2022. – С. 24-28
7. Куклина А.С., Щербакова И.В. Особенности применения теории фронтального обучения в сочетании с современными возможностями концепции Blended Learning обучения / А.С. Куклина, И.В. Щербакова // Инновационные технологии в образовательной деятельности: материалы XXV международной научно-методической конференции. – Нижний Новгород, 2023. – С. 84-86
8. Куклина И.Г. Сравнение аппаратов объектно-ориентированного программирования в проектировании автотехники // Проблемы транспортных и технологических комплексов: сб. статей III Международ-

- ной научно-практической конференции. – Нижний Новгород, 2012. – НГТУ. – С. 119-122 – ISSN: 2307-1400
9. Куклина А.С., Куклина И.Г. Резистентная сингулярность узкого социума в мире / А.С. Куклина, И.Г. Куклина // Культура. Наука. Образование: актуальные вопросы. – БОУ ВО Чувашский государственный институт культуры и искусств. Минкультуры Чувашии Чебоксары, 2020. – С. 23-32
 10. Мину М. Математическое программирование. Теория и алгоритмы: Пер. с франц. / М. Мину. – Москва: Наука, 1990. – 487 с.
 11. Понтрягин Л.С. Математическая теория оптимальных процессов /Л.С. Понтрягин. - Москва: Наука, 1983. – 283 с.
 12. Сухарев, А.Г. Курс методов оптимизации. / А.Г. Сухарев, А.В. Тимохов, В.В. Федоров. – Москва: Наука, 1986. – 328 с.
 13. Хедли Дж. Нелинейное и динамическое программирование: Пер. с англ. /Дж. Хедли. – Москва: Мир, 1969. – 424 с.
 14. Черемных С.В. Моделирование и анализ систем. IDEF - технологии / С.В. Черемных, И.О. Семенов, В.С. Ручкин – Москва: Финансы и статистика, 2006. – 192 с. – ISSN: 5-279-02564-X
 15. Шеер А.В. Моделирование бизнес-процессов / А.В. Шеер – Москва: Весть-МетаТехнология, 2000 – 222 с. – ISSN: 5-89163-049-4
 16. Эльсгольц Л.Э. Дифференциальные уравнения и вариационное исчисление /Л.Э. Эльсгольц. - М.: Наука, 1969. – 146 с.

Куклина Ирина Геннадьевна

МЕТОДЫ ОПТИМИЗАЦИИ ЛИНЕЙНОГО
И НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Учебное пособие

Редактор:
Н.В. Викулова

Подписано в печать формат 60x90 1/16. Бумага газетная. Печать трафаретная.
Уч. изд. л. 6,0. Усл. печ. л. 6,3 Тираж 300 экз. Заказ №

Федеральное государственное образовательное учреждение высшего образования
«Нижегородский государственный архитектурно-строительный университет»
603950, Нижний Новгород, ул. Ильинская, 65.
Полиграфический центр ННГАСУ, 603950, Нижний Новгород, ул. Ильинская, 65
<http://www.nngasu.ru>, srec@nngasu.ru