Министерство науки и высшего образования Российской Федерации

Государственное образовательное учреждение высшего профессионального образования

«Алтайский государственный технический университет им. И. И. Ползунова»

Факультет информационных технологий

Кафедра Прикладная математика

Отчет защищен с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель         А. В. Сорокин

(подпись) (и.о.фамилия)

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_ 2022 г.

дата

Отчет

по дисциплине

ОСНОВЫ МОДЕЛИРОВАНИЯ

лабораторная работа №3

Математические модели в задаче линейного программирования

название работы

ЛР 09.03.04.23.003 О

обозначение документа

Студент группы гр. ПИ-02                                                     Чередов Р.А.

*(подпись*)

Преподаватель доцент, к.т.н.               А. В. Сорокин

должность, ученое звание и.о., фамилия

БАРНАУЛ 2022

# **Аннотация**

Задачи линейного программирования представляют собой оптимизационные задачи, описываемые линейными математическими моделями. В общем виде постановка оптимизационной задачи математического программирования состоит в определении таких значений переменных х1, х2, ... , хп, при которых целевая функция достигает наибольшего или наименьшего значения, а сами переменные удовлетворяют одновременно системе ограничений

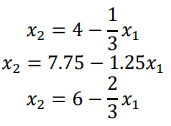
Оглавление

[**Аннотация** 2](#_Toc154646241)

[**Задание** 4](#_Toc154646242)

[**Математическая модель** 5](#_Toc154646243)

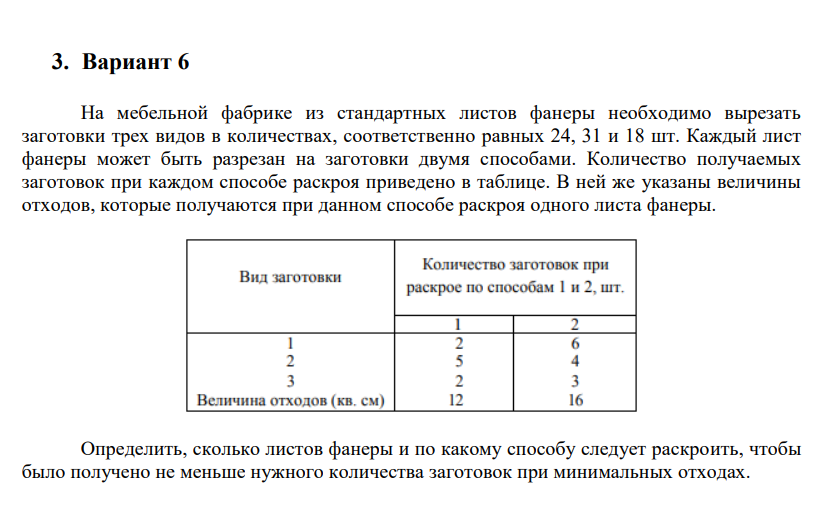
[**Графический метод** 6](#_Toc154646244)

[ 6](#_Toc154646245)

[**Заключение** 7](#_Toc154646246)

[**Вопросы при защите** 7](#_Toc154646247)

# **Задание**



# **Математическая модель**

X1-кол-во листов фанеры раскроенных по первому способу.

X2-кол-во листов фанеры раскроенных по второму способу.

Величина отходов по первому способу равна 12 условных единиц, по второму 16 условных единиц. В итоге величина отходов должна получиться минимальной. Исходя из этого, целевая функция будет выглядеть следующим образом:

F=12x1+16x2→min

В задании сказано, что заготовок 1-го вида должно получиться 24, 2-го вида – 31, 3- го вида – 18. Из этого вытекает 3 ограничения:

1. 2x1+6x2≥24

2. 5x1+4x2≥31

3. 2x1+3x2≥18

По смыслу задачи кол-во листов фанеры не может быть отрицательным, поэтому добавим еще 2 ограничения:

1. x1≥0

2. x2≥0

В итоге получается следующая математическая модель:

F=12x1+16x2→min

2x1+6x2≥24

5x1+4x2≥31

2x1+3x2≥18

x1≥0,x2≥0

# **Графический метод**

Для решения ЗЛП используется известный симплекс метод, основанный на поиске решения на границе области, описываемой системой неравенств. Алгоритм, пробегая по граням и вершинам многогранника, ищет ту точку множества, которая дает оптимальное решение.

Наглядным способом решения ЗЛП является графический метод.

**Решение**

Построим область допустимых решений ЗЛП, в которой ищется решение. Для этого нужно ограничения-неравенства превратить в равенства:

2x1+6x2=24

5x1+4x2=31

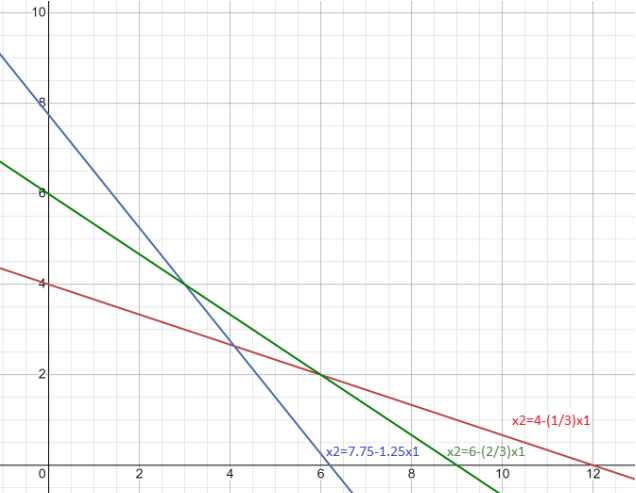
2x1+3x2=18

и построить соответствующие им прямые на плоскости с осями x1, x2. x1 будет соответствовать обычному x, а x2 – y.

Уравнение прямых:

# 

Построим данные прямые линии в декартовой системе координат x1O x2



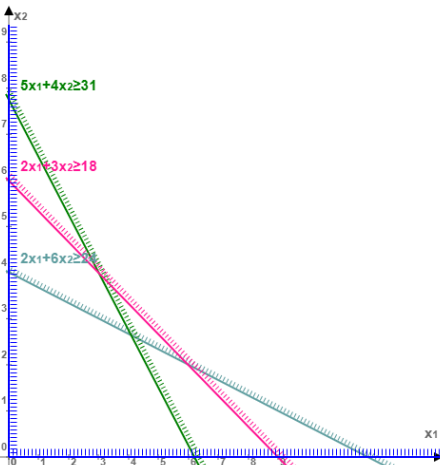
Подставим значения (x1, x2) возле каждой исследуемой прямой и определим с какой стороны от прямой находится допустимая область

Возьмем первое неравенство 2x1+6x2 ≥ 24 и точку (x1,x2)=(1,5), получим 32≥24, т.е. неравенство выполняется и допустимая область находится выше прямой 2x1+6x2=24

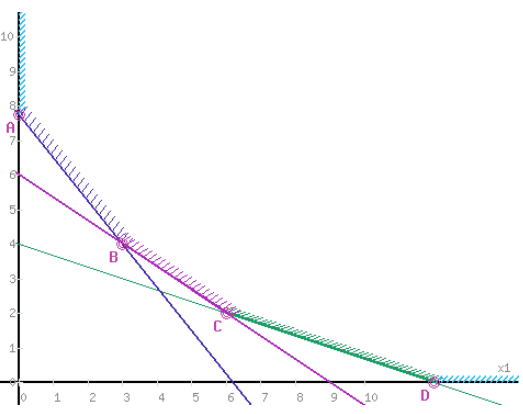
Возьмем второе неравенство 5x1+4x2 ≥ 31 и точку (x1,x2)=(3,5), получим 35≥31, т.е. неравенство выполняется и допустимая область находится выше прямой 5x1+4x2=31.

Возьмем третье неравенство 2x1+3x2 ≥ 18 и точку (x1,x2)=(1,6), получим 20≥18, т.е. неравенство выполняется и допустимая область находится выше прямой 2x1+3x2=18.

Сделаем штриховку по ту сторону линии, где выполняется соответствующее неравенство.

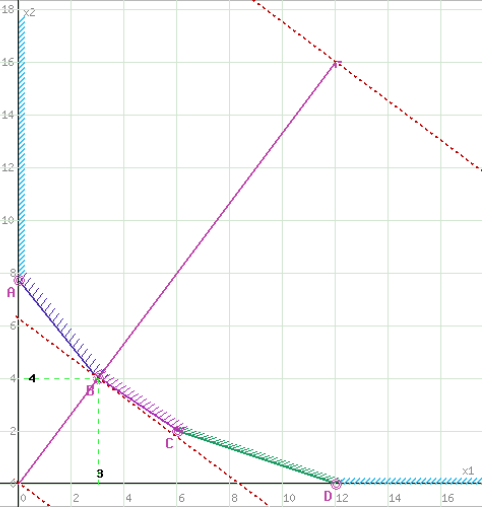


Таким образом, мы проверили расположение области допустимых решений относительно прямых линий, полученных из неравенств, и можем легко построить область допустимых решений, обозначим границы области допустимых решений:



Рассмотрим целевую функцию задачи F=12x1+16x2→min

Построим прямую, отвечающую значению функции F = 12x1+16x2 = 0. Вектор-градиент, составленный из коэффициентов целевой функции, указывает направление максимизации F(X). Начало вектора – точка (0; 0), конец – точка (12;16). Будем двигать эту прямую параллельным образом. Поскольку нас интересует минимальное решение, поэтому двигаем прямую до первого касания обозначенной области. На графике эта прямая обозначена пунктирной линией.



Прямая F(x) = const пересекает область в точке B (наиболее вероятная точка минимума функции цели).

Потенциальными точками максимума нашей задачи могут быть точки А, B, С и D. Проверим их. Координаты точек A и D легко определяются из уравнений прямых и равны соответственно (0,8) и (12,0).

Чтобы найти координаты точки B нужно решить систему уравнений, состоящую из двух уравнения (2) и (3):

5x1+4x2=31

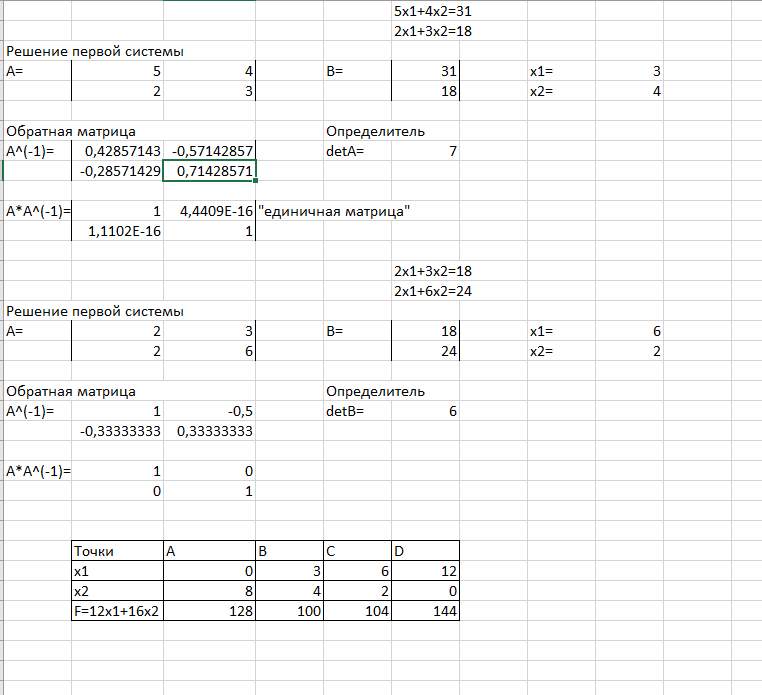
2x1+3x2=18

а для нахождения координат точки C, нужно решить систему уравнений, состоящую из двух уравнения (1) и (3)

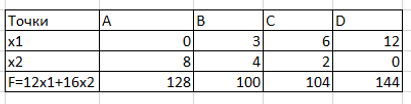
2x1+6x2=24

2x1+3x2=18

Для решения данных систем уравнений воспользуемся Excel



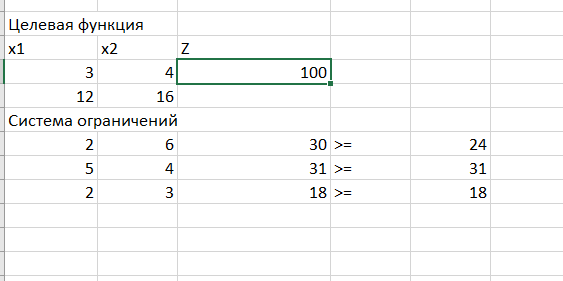
В результате мы получим координаты всех потенциальных точек, дающих максимум, функции цели F. Приведем их в таблице с вычисленной в Excel функцией цели:



Из таблицы видно, что оптимальной является точка B с координатами (3, 4) и значением функции цели F=100.

# **Решение ЗЛП в Excel**

Построим таблицу в Excel:

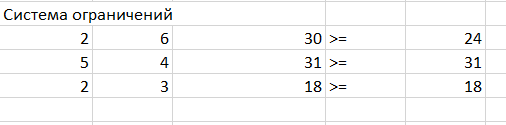


Далее во вкладке «Данные» выбираем пункт «Поиск решения»

Выбираем ячейку с целевой функцией, ставим галочку минимум, далее выбираем ячейки изменяемых переменных и добавляем ограничения при помощи кнопки Добавить. Также ставим галочку переменные без ограничений неотрицательные, выбираем, выбираем метод решения – симплекс-метод решения линейных задач.

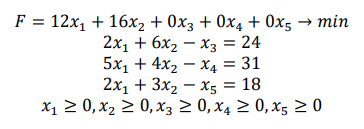
Итак, нажимаем «Найти решение», появляется окно результаты поиска решений, выбираем сохранить найденное решение

Результат:



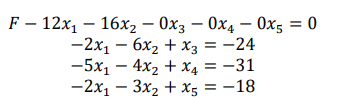
# **Решение ЗЛП табличным сиплекс-методом**

Чтобы решить задачу симплекс-методом добавить к выбранной модели искусственные переменные (обычно их 3, но может быть и другое количество), чтобы система ограничений приняла систему уравнений (x3, x4, x5):



Цель дальнейшего применения симплекс-метода – избавление от искусственных (дополнительных) переменных х3, х4, х5 выводом их из несвободных (базисных) переменных.

После введения добавочных переменных систему уравнений и линейную функцию записываем в виде расширенной системы:



Все добавочные переменные имеют неодинаковые знаки со свободными членами. Применим двойственный симплекс метод для решения задачи.

# **Программа**

import math  
from prettytable import PrettyTable  
  
limitations = [  
 [-2, -6, 1, 0, 0],  
 [-5, -4, 0, 1, 0],  
 [-2, -3, 0, 0, 1]  
]  
  
n = len(limitations)  
m = len(limitations[0])  
  
F = [-12, -16, 0, 0, 0]  
  
base = [3, 4, 5]  
  
free\_terms = [-24, -31, -18, 0]  
  
table = PrettyTable()  
field\_names = ["Базис", "Свободный член"]  
for i in range(m):  
 field\_names.append("x{}".format(i+1))  
field\_names.append("Оценочное отношение")  
table.field\_names = field\_names  
  
print("Исходная таблица:")  
while max(F) > 0 or min(free\_terms) < 0:  
 inf = [False] \* n  
  
 minus\_flag = False  
 row = min((i for i, num in enumerate(free\_terms[:-1]) if num < 0), default=None,key=lambda x: free\_terms[x])  
 col = -1  
 ocen\_otn = [0] \* n  
 if row != None:  
 print("\nЕсть отрицательные свободные члены!")  
 ocen\_otn = [0] \* m  
 inf = [False] \* m  
 minus\_flag = True  
 else:  
 col = max((i for i, num in enumerate(F) if num > 0), default=None, key=lambda x: F[x])  
 #col = max((i for i, num in enumerate(F) if num > 0), default=None, key=lambda x: abs(F[x]))  
 if not minus\_flag:  
 for i in range(n):  
 if limitations[i][col] == 0 or (limitations[i][col] < 0 and free\_terms[i] == 0) or (limitations[i][col] > 0 and free\_terms[i] < 0) or (limitations[i][col] < 0 and free\_terms[i] > 0):  
 inf[i] = True  
 elif free\_terms[i] == 0 and limitations[i][col] > 0:  
 ocen\_otn[i] = 0  
 else:  
 ocen\_otn[i] = free\_terms[i] / limitations[i][col]  
 else:  
 for j in range(m):  
 if limitations[row][j] == 0 or (limitations[row][j] < 0 and F[j] == 0) or (limitations[row][j] > 0 and F[j] < 0) or (limitations[row][j] < 0 and F[j] > 0):  
 inf[j] = True  
 elif F[j] == 0:  
 ocen\_otn[j] = 0  
 else:  
 ocen\_otn[j] = F[j] / limitations[row][j]  
# Вывод таблицы  
for i in range(n):  
 table\_row = ["x{}".format(base[i]), free\_terms[i]]  
 for j in range(m):  
 table\_row.append(limitations[i][j])  
 if minus\_flag:  
 table\_row.append(" ")  
 elif inf[i]:  
 table\_row.append("inf")  
 else:  
 table\_row.append(ocen\_otn[i])  
 table.add\_row(table\_row)  
  
table\_row = ["F", free\_terms[n]]  
for i in range(m):  
 table\_row.append(F[i])  
table\_row.append(" ")  
table.add\_row(table\_row)  
if minus\_flag:  
 table\_row = ["Оценочное отношение", ""]  
 for i in range(m):  
 if inf[i]:  
 table\_row.append("inf")  
 else:  
 table\_row.append(ocen\_otn[i])  
 table\_row.append("")  
 table.add\_row(table\_row)  
print(table)  
table.clear\_rows()  
if not minus\_flag:  
 biais = [(i, num) for i, num in enumerate(ocen\_otn) if (num > 0 or num < 0) and not inf[i]]  
 if len(biais) == 0:  
 print("Конечного минимума нет!")  
 break  
 min\_biais = min(biais, key=lambda x: x[1])  
 row = min\_biais[0]  
else:  
 biais = [(i, num) for i, num in enumerate(ocen\_otn) if (num > 0 or num < 0) and not inf[i]]  
 if len(biais) == 0:  
 print("Конечного минимума нет!")  
 break  
 min\_biais = min(biais, key=lambda x: x[1])  
 col = min\_biais[0]  
  
new\_limitations = [[0] \* m for \_ in range(n)]  
new\_free\_terms = [0] \* (n+1)  
new\_F = [0] \* m  
base[row] = col + 1  
new\_free\_terms[row] = free\_terms[row] / limitations[row][col]  
for i in range(m):  
 if i+1 in base:  
 new\_F[i] = 0  
 else:  
 new\_F[i] = F[i] - (F[col] \* limitations[row][i]) / limitations[row][col]  
 #print(F[i], F[col], limitations[row][j], limitations[row][col])  
for i in range(n):  
 for j in range(m):  
 if base[i] == j+1:  
 new\_limitations[i][j] = 1  
 elif j+1 in base:  
 new\_limitations[i][j] = 0  
 elif i == row:  
 new\_limitations[i][j] = limitations[i][j] / limitations[row][col]  
 else:  
 new\_limitations[i][j] = limitations[i][j] -(limitations[i][col] \* limitations[row][j]) / limitations[row][col]  
for i in range(n):  
 if i == row: continue  
 new\_free\_terms[i] = free\_terms[i] - (limitations[i][col] \* free\_terms[row]) / limitations[row][col]  
 new\_free\_terms[n] = free\_terms[n] - (F[col] \* free\_terms[row]) / limitations[row][col]  
  
limitations = new\_limitations  
F = new\_F  
free\_terms = new\_free\_terms  
# Вывод последней таблицы  
print("\nРезультат:")  
for i in range(n):  
 table\_row = ["x{}".format(base[i]), free\_terms[i]]  
 for j in range(m):  
 table\_row.append(limitations[i][j])  
 table\_row.append("")  
 table.add\_row(table\_row)  
  
table\_row = ["F", free\_terms[n]]  
for i in range(m):  
 table\_row.append(F[i])  
table\_row.append(" ")  
table.add\_row(table\_row)  
print(table)

# **Результат работы**



Результат работы программы совпал с результатом, полученным графическим методом и результатом, полученным в Excel.

# **Заключение**

Задачи линейного программирования представляют собой оптимизационные задачи, описываемые линейными математическими моделями. В общем виде постановка оптимизационной задачи математического программирования состоит в определении таких значений переменных х1, х2, ... , хп, при которых целевая функция достигает наибольшего или наименьшего значения, а сами переменные удовлетворяют одновременно системе ограничений.

В данной лабораторной работе были рассмотрены методы решения задач линейного программирования. А именно – графический метод, средствами Excel, табличный симплекс-метод.

В процессе выполнения лабораторной работы возникли трудности с пониманием двойственного симплекс-метода, но почитав немного учебник и немного поискав информацию в интернете – разобрался

# **Список использованных источников**

1. Сорокин А.В. Использование алгоритма Джонсона для решения задачи упорядочения. Методические указания. - Барнаул: АлтГТУ, 2015. – 32 с. – [Электронный ресурс]. – url: <http://elib.altstu.ru/eum/download/pm/Sorokin_alg_Johnson.pdf>
2. Алпатов, Ю.Н. Математическое моделирование производственных процессов [Электронный ресурс] : учеб. пособие / Ю.Н. Алпатов. — Электрон. дан. — Санкт Петербург : Лань, 2018. — 136 с. — Режим доступа: https://e.lanbook.com/book/107271. — Загл. с экрана.
3. Голубева, Н.В. Математическое моделирование систем и процессов [Электронный ресурс] : учебное пособие / Н.В. Голубева. — Электрон. дан. — Санкт-Петербург : Лань, 2016. — 192 с. — Режим доступа: https://e.lanbook.com/book/76825. — Загл. с экрана
4. Кудрявцев, Е.М. GPSS World. Основы имитационного моделирования различных систем [Электронный ресурс] / Е.М. Кудрявцев. — Электрон. дан. — Москва : ДМК Пресс, 2008. — 317 с. — Режим доступа: https://e.lanbook.com/book/1213. — Загл. с экрана.
5. Березовская, Е.А. Имитационное моделирование : учебное пособие / Е.А. Березовская ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего образования «Южный федеральный университет», Экономический факультет. - Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2018. - 76 с. : ил. - Библиогр. в кн. - ISBN 978-5 9275- 2426-6 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=499496 (21.04.2019)
6. Лисяк, Н.К. Моделирование систем : учебное пособие / Н.К. Лисяк, В.В. Лисяк ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего образования «Южный федеральный университет», Инженерно-технологическая академия. - Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2017. - Ч. 1. - 107 с. : ил. - Библиогр.: с. 101 - 102 - ISBN 978-5-9275- 2504-1 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=499733 (21.04.2019)
7. Аверченков, В.И. Основы математического моделирования технических систем : учебное пособие / В.И. Аверченков, В.П. Федоров, М.Л. Хейфец. - 3-е изд., стер. - Москва : Издательство «Флинта», 2016. - 271 с. : схем., ил. - Библиогр. в кн. - ISBN 978-5-9765-1278- 8 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=93344 (21.04.2019).
8. Теория систем массового обслуживания : учебное пособие / сост. А.В. Шапошников, В.В. Бережной, А.М. Лягин, А.А. Плетухина и др. - Ставрополь : СКФУ, 2017. - 134 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=483842 (24.04.2019).
9. Поршнев, С.В. Компьютерное моделирование физических процессов в пакете MATLAB [Электронный ресурс] : учебное пособие / С.В. Поршнев. — Электрон. дан. — Санкт Петербург : Лань, 2011. — 736 с. — Режим доступа: https://e.lanbook.com/book/650. — Загл. с экрана.
10. Материалы сайта "Интернет университет информационных технологий" , сетевой адрес: http://www.intuit.ru/, Костюкова Н. Основы математического моделирования. - НГУ, сетевой адрес: https://www.intuit.ru/studies/courses/66/66/info Материалы сайта "Единое окно доступа к образовательным ресурсам", Андриевский А.Б.,
11. Андриевский Б.Р., Капитонов А.А., Фрадков А.Л. Решение инженерных задач в среде Scilab. Учебное пособие. - СПб.: НИУ ИТМО, 2013. — 97 с. – [Электронный ресурс]. – url:http://window.edu.ru/resource/044/80044/files/itmo1329.pdf