



Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Робототехники и комплексной автоматизации

КАФЕДРА Системы автоматизированного проектирования (РК-6)

## **ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

Студент Никифорова Ирина Андреевна

Группа РК6-61б

Тип задания лабораторная работа

Тема лабораторной работы Интерполяция Лагранжа.  
Интерполяция кубическими сплайнами.

Студент	_____	<b>Никифорова И. А.</b>
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>
Преподаватель	_____	<b>Соколов А. П.</b>
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>
Преподаватель	_____	<b>Першин А. Ю.</b>
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>

Оценка \_\_\_\_\_

Москва, 2019 г.

## **Оглавление**

Задание на лабораторную работу	<b>2</b>
Цель выполнения лабораторной работы	<b>5</b>
Задачи, выполненные в процессе реализации лабораторной работы	<b>5</b>
Интерполяция Лагранжа	<b>6</b>
Интерполяция кубическими сплайнами	<b>13</b>
Заключение	<b>18</b>
Список использованных источников	<b>19</b>

## Задание на лабораторную работу

### Задание 1:

Дана рациональная функция, известная как аппроксимация Паде:

$$f_{n,m}(x) = \frac{\sum_{j=0}^m a_j x^j}{1 + \sum_{k=1}^n b_k x^k}, \text{ где } x \in [-1; 1]. \quad (1)$$

Требуется:

1. Разработать функцию  $l\_i(i, x, x\_nodes)$ , которая возвращает значение  $i$ -го базисного полинома Лагранжа, заданного на узлах с абсциссами  $x\_nodes$ , в точке  $x$ .
2. Написать функцию  $L(x, x\_nodes, y\_nodes)$ , которая возвращает значение интерполяционного полинома Лагранжа, заданного на узлах с абсциссами  $x\_nodes$  и ординатами  $y\_nodes$ , в точке  $x$ .
3. Сгенерировать 100 функции  $f_{n,m}(x)$ , где целые степени  $n, m \in [7; 15]$  и вещественные коэффициенты  $a, b \in [0; 1]$  генерируются случайным образом для каждой из функций.
4. Для нескольких из сгенерированных функций вывести на экран одновременно графики  $f_{n,m}(x)$  и соответствующего интерполяционного полинома  $L(x)$ , построенного по  $N$  равномерно расположенным узлам, где  $N$  выбирается по собственному усмотрению, но должно быть не меньше 5. На том же графике выведите  $L(x)$ , построенного по  $N$  чебышевским узлам.
5. Для каждой из функций, сгенерированных в предыдущем пункте, найдите интерполяционные полиномы  $L(x)$ , построенные по  $N \in \{1, 2, \dots, 30\}$  равномерно расположенным узлам и чебышевским узлам. Для каждого  $N$  рассчитайте расстояние между  $L(x)$  и  $f_{n,m}(x)$  в лебеговом пространстве  $L^\infty$ .

Рассмотрите несколько графиков зависимости этого расстояния для равномерных и чебышевских узлов от  $N$  и сделайте по ним вывод. Добавьте в отчет один характерный график, который наглядно демонстрирует верность вашего вывода.

6. Объясните, что такое аппроксимация Паде и почему предложенный метод генерации случайных функций  $f_{n,m}(x)$  позволяет до некоторой степени обобщить выводы предыдущего пункта на произвольные функции.

### **Задание 2:**

Требуется:

1. Разработать функцию *qubic\_spline\_coeff*(*x\_nodes*, *y\_nodes*), которая посредством решения матричного уравнения вычисляет коэффициенты естественного кубического сплайна.

2. Написать функции *qubic\_spline*(*x*, *qs\_coeff*) и *d\_qubic\_spline*(*x*, *qs\_coeff*), которые вычисляют соответственно значение кубического сплайна и его производной в точке (*qs\_coeff* обозначает матрицу коэффициентов).

3. На интернет-ресурсе <https://rp5.ru> найдите данные о температуре за один произвольный календарный месяц в произвольном городе, первая буква которого совпадает с первой буквой вашей фамилии. Проведите интерполяцию кубическими сплайнами по полученным данным и выведите результат на экран.

4. Из текущей выборки удалите половину узлов таким образом, что в выборке остаются измерения только в 02:00, 08:00, 14:00, 20:00. Проведите интерполяцию кубическими сплайнами по новой выборке и выведите результаты на экран.

5. Из последней выборки снова удалите половину узлов таким образом, что в выборке остаются измерения только в 02:00, 14:00, и проведите аналогичную интерполяцию кубическими сплайнами.

6. Сравните три полученные аппроксимации эволюции температуры. Оцените, насколько хорошо аппроксимации во втором и третьем случае предсказывают температуру в известные моменты времени, которые были отброшены при фильтрации узлов интерполяции. Также оцените, насколько хорошо эти аппроксимации предсказывают температуру, усредненную в течение дня (усредненная температура получается путем вычисления среднего арифметического известных значений температуры в течение каждого из дней).

## **Цель выполнения лабораторной работы**

### **Цель выполнения лабораторной работы –**

Изучить различные способы интерполяции данных и показать их использование в аналитическом и графическом видах.

Для задания 1: Продемонстрировать, что использование вместо равномерно распределенных узлов чебышевских узлов ускоряет сходимость аппроксимирующего полинома к произвольной непрерывной функции при увеличении степени полинома.

Для задания 2: Продемонстрировать использование кубических сплайнов для интерполяции реальной температуры, измеряемой на метеостанции, в случае различных интервалов измерений.

### **Задачи, выполненные в процессе реализации лабораторной работы**

1. Задание 1. Интерполяция Лагранжа
2. Задание 2. Интерполяция кубическими сплайнами

## 1. Интерполяция Лагранжа

Для выполнения этой части работы использовался язык Go версии 1.12 и его стандартные пакеты (`math`, `strconv` и др.). Для построения графиков была подключена библиотека `gonum.org/v1/plot` [1]. Все функции были разбиты на три пакета: `mathExtra`, `plotExtra` и `main` (файл `lagrPolynom.go`)

В соответствии с заданием была разработана функция  $l\_i(i, x, x\_nodes)$ , которая возвращает значение  $i$ -того базисного полинома Лагранжа, заданного на узлах с абсциссами  $x\_nodes$  в точке  $x$ . Для вычислений была использована формула (2) из лекций по вычислительной математике [2].

$$l_i(x) = \prod_{i \neq j} \frac{x - x_j}{x_i - x_j} \quad (2)$$

Также была написана функция  $L(x, x\_nodes, y\_nodes)$ . Для удобства при дальнейшем использовании, она возвращает не значение полинома Лагранжа, а анонимную функцию  $func(x)$ , вычисляющую это значение при заданном  $x$ . Вычисления выполнялись по формуле (3) из лекций по вычислительной математике [2].

$$L_{n-1}(x) = \sum_{i=1}^n f(x_i) l_i(x) = \sum_{i=1}^n f(x_i) \prod_{i \neq j} \frac{x - x_j}{x_i - x_j} \quad (3)$$

Далее была построена функция аппроксимации Паде  $f_{n,m}(x)$ . Для этого была написана функция `generatePadeApproxFunc()`, возвращающая функцию типа `func(x float64) float64`. Результирующая функция аппроксимации Паде принимает только значение абсциссы  $x$  типа `float64`. Необходимые для вычислений по формуле (1) целые степени  $n, m \in [7; 15]$  и вещественные коэффициенты  $a, b \in [0; 1]$  генерируются случайным образом для каждой из

функций в замыкании *generatePadeApproxFunc()* с помощью специальной функции *generateFuncConsts()*.

Для генерации сразу нескольких функций была также написана функция *generateSomePadeApproxFuncs(numOfFuncs)*. С помощью нее были сгенерированы  $N = 8$  функций аппроксимации Паде *funcsPade[N]*. В функции *createSomePadeLagrPlots(funcsPade[], numOfPoints)* для каждой из них были созданы функции  $f(x)$ , вычисляющие многочлены Лагранжа по равномерно распределенным и чебышёвским узлам для заданной абсциссы  $x$ . Там же функции были выведены на графики по три: функция Паде и соответствующие ей полиномы Лагранжа.

На полученных графиках (рис.1 — рис.8) можно увидеть, как именно полиномы Лагранжа приближают исходную функцию аппроксимации Паде. Были также отдельно выделены (окружностями) узлы Чебышёва на каждом из графиков.

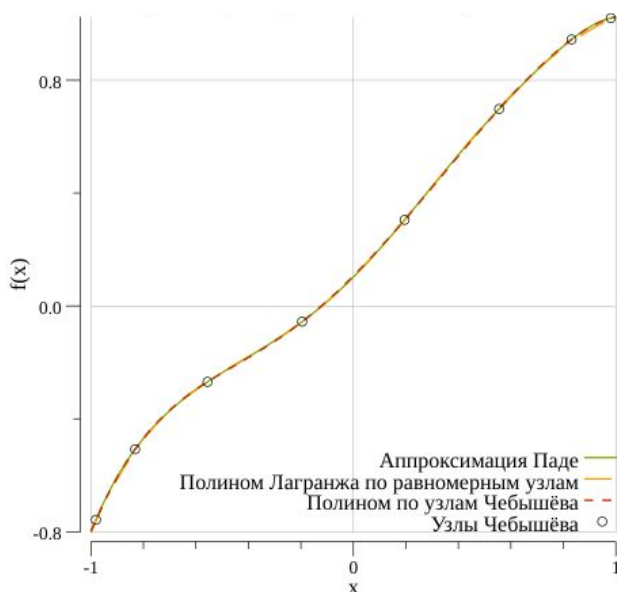


Рис. 1. Аппроксимация Паде и полиномы Лагранжа, вариант 1

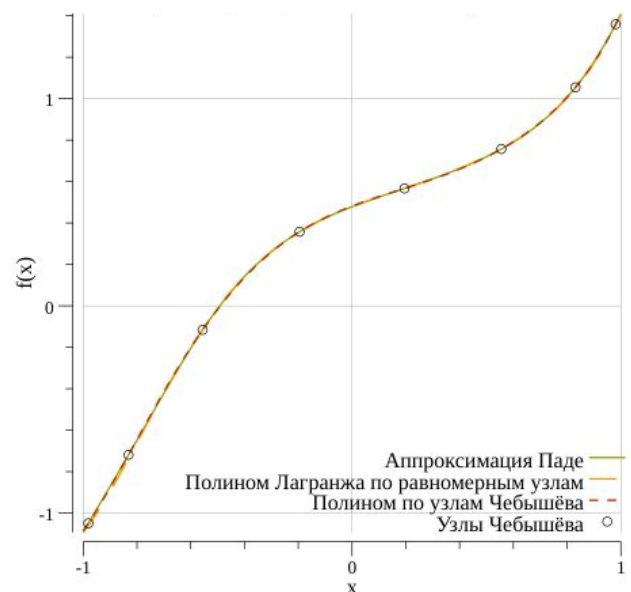


Рис. 2. Аппроксимация Паде и полиномы Лагранжа, вариант 2



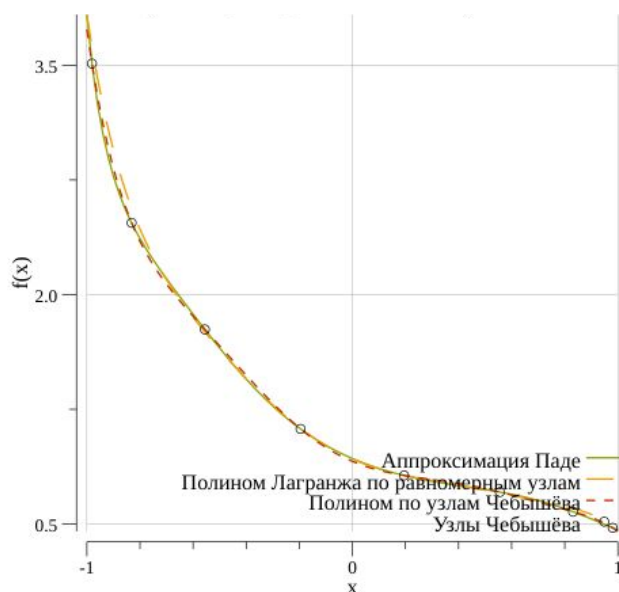


Рис. 3. Аппроксимация Паде и полиномы Лагранжа, вариант 3

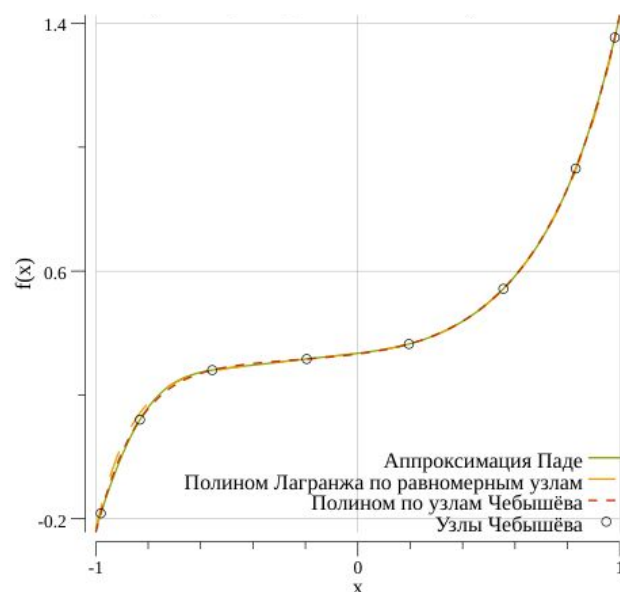


Рис. 4. Аппроксимация Паде и полиномы Лагранжа, вариант 4

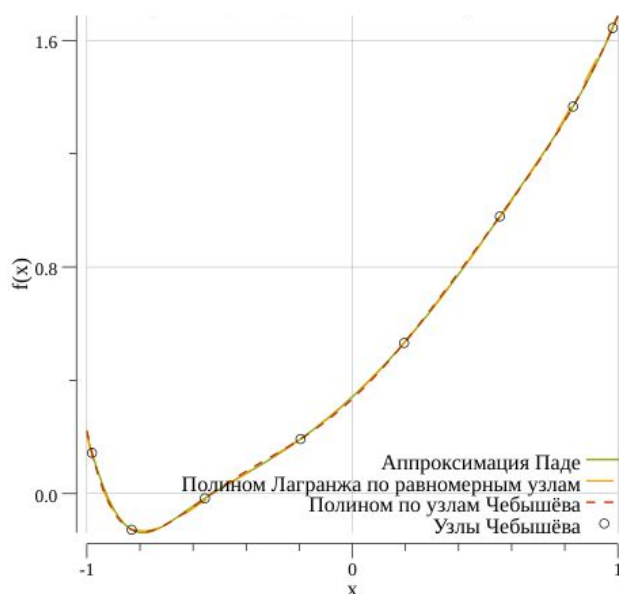


Рис. 5. Аппроксимация Паде и полиномы Лагранжа, вариант 5

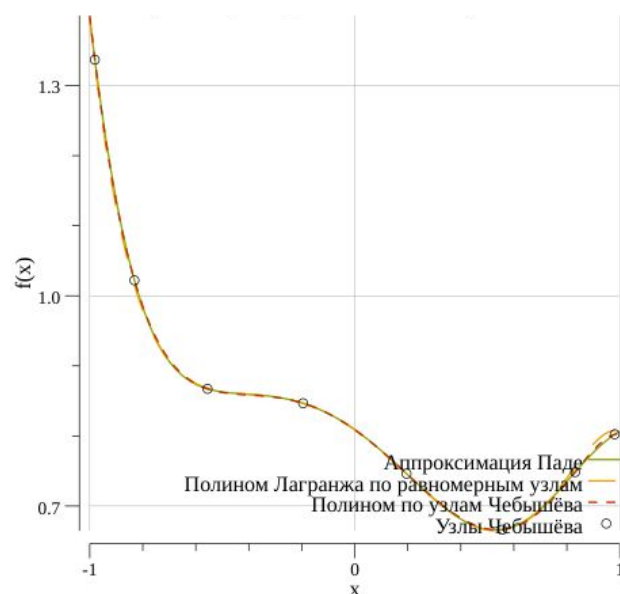


Рис. 6. Аппроксимация Паде и полиномы Лагранжа, вариант 6

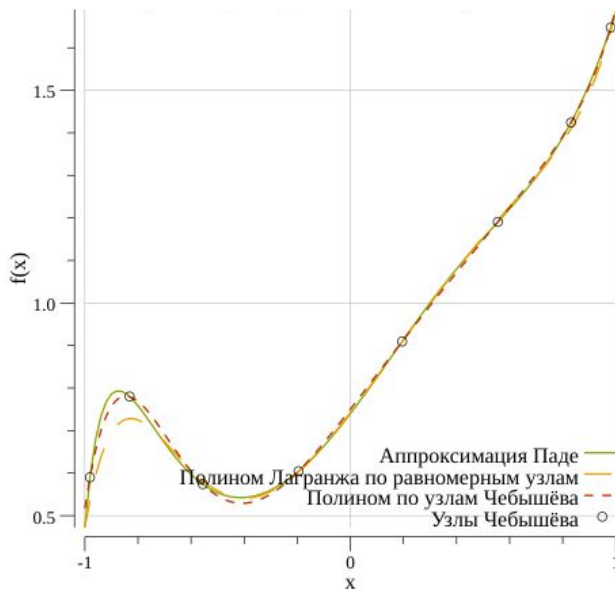


Рис. 7. Аппроксимация Паде и полиномы Лагранжа, вариант 7

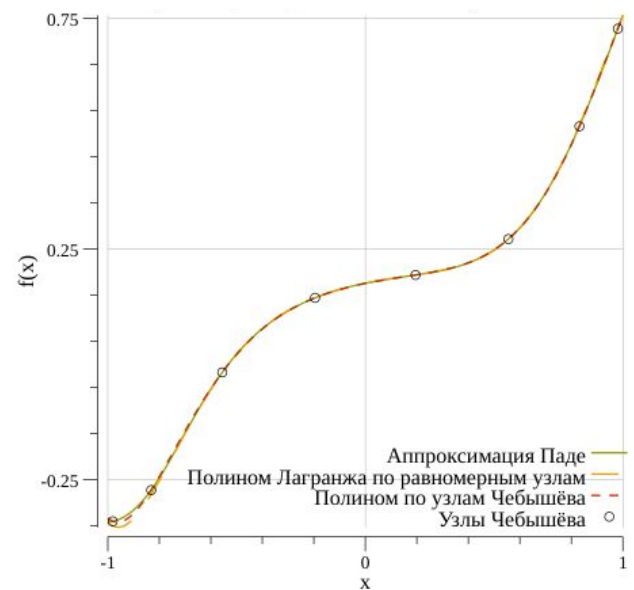


Рис. 8. Аппроксимация Паде и полиномы Лагранжа, вариант 8

Используя те же функции аппроксимации Паде, которые были сгенерированы ранее, был построен набор функций  $L(x)$ , созданных по  $N \in \{1, 2, \dots, 30\}$  равномерно расположенным узлам и, аналогично, по чебышёвским узлам. Для генерации полиномов Лагранжа была написана функция `createLagrFuncs(funcPade, start, end)`, принимающая функцию аппроксимации Паде, начало и конец диапазона значений количества узлов интерполяции. Она возвращает 2 среза функций, размерностями для обоих  $start - end + 1$ : для равномерно распределенных узлов и для чебышёвских узлов.

Также была написана функция `computeNorm(funcPade, funcLagr)`, вычисляющая равномерную норму между двумя функциями в лебеговом пространстве  $L^\infty$  как максимальную разницу ординат функций в заданном промежутке для абсцисс [3].

Далее эти функции были использованы для построения графиков зависимостей значения нормы от количества узлов расчета в функции `createNormPlot(funcPade, num)`. По полученным графикам (рис. 9 — рис. 16)

можно сделать выводы об общем виде зависимости отклонения интерполяции от реальной функции для равномерных и для чебышёвских узлов.

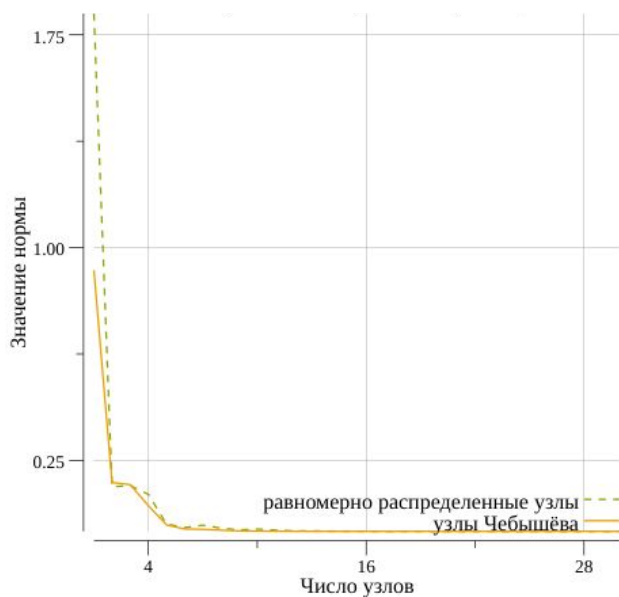


Рис. 9. Зависимость нормы от числа узлов для функции *funcspade[0]*

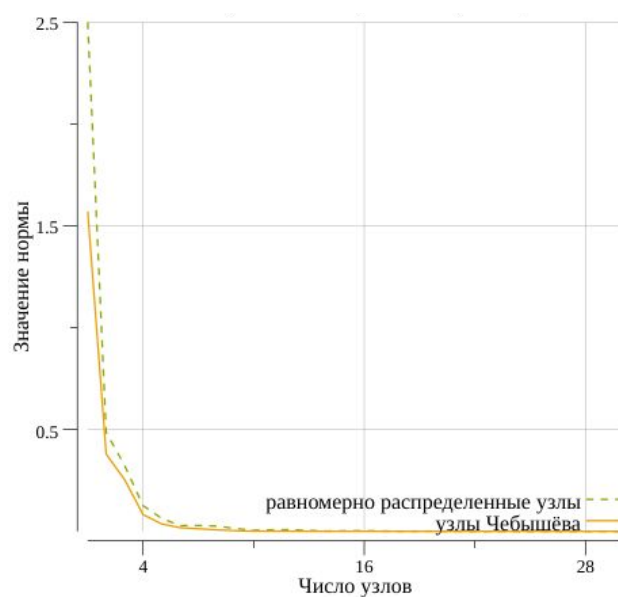


Рис. 10. Зависимость нормы от числа узлов для функции *funcspade[1]*

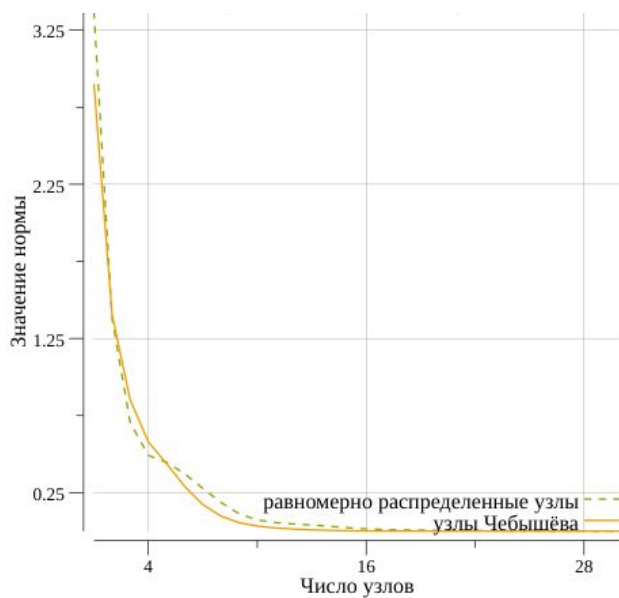


Рис. 11. Зависимость нормы от числа узлов для функции *funcspade[2]*

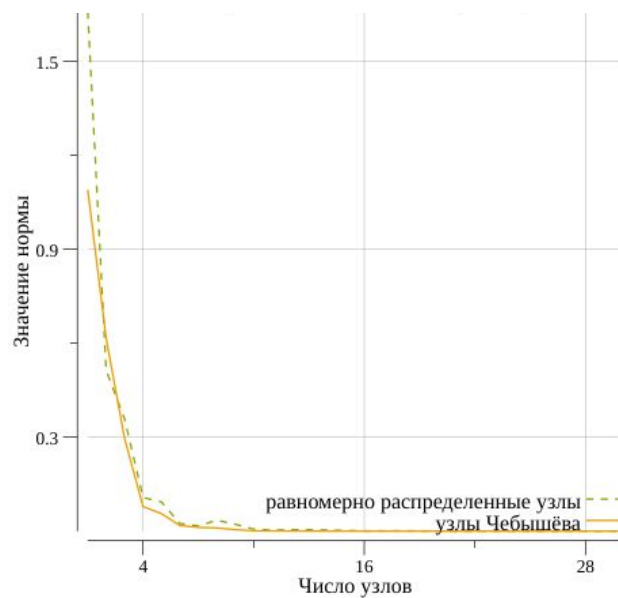


Рис. 12. Зависимость нормы от числа узлов для функции *funcspade[3]*

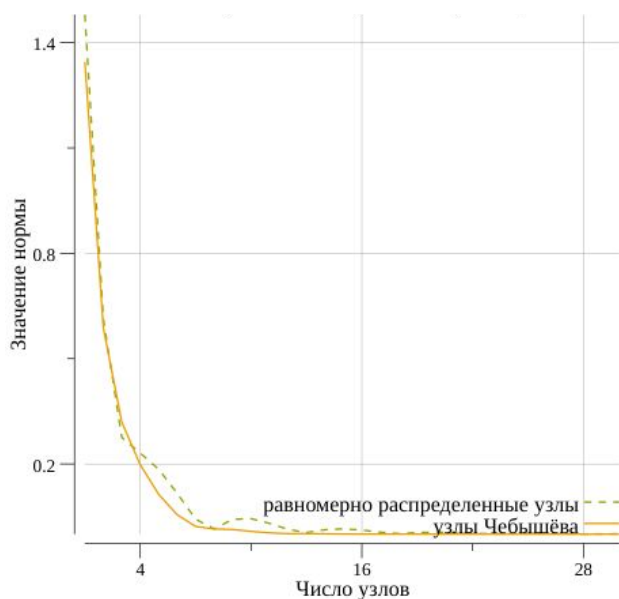


Рис. 13. Зависимость нормы от числа узлов для функции *funcspade[4]*

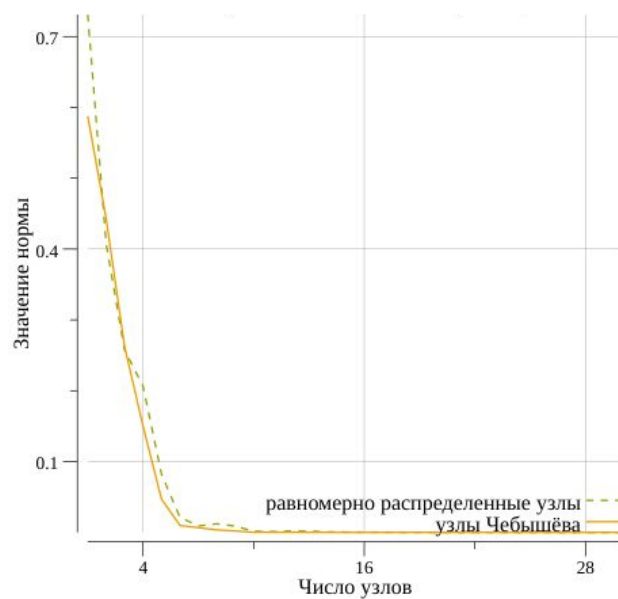


Рис. 14. Зависимость нормы от числа узлов для функции *funcspade[5]*

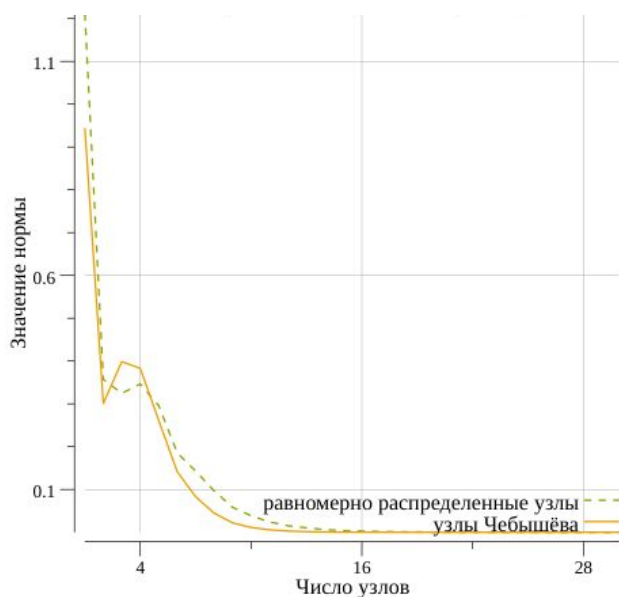


Рис. 15. Зависимость нормы от числа узлов для функции *funcspade[6]*

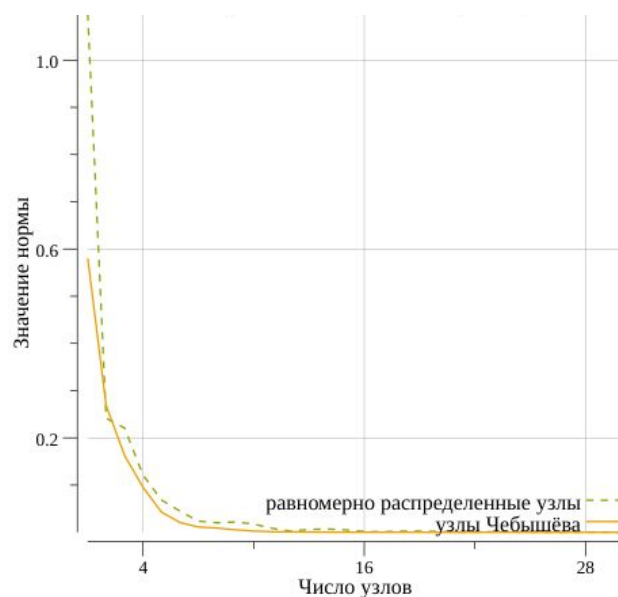


Рис. 16. Зависимость нормы от числа узлов для функции *funcspade[7]*

На каждом из графиков (рис. 9 - рис. 16) можно увидеть, что при небольшом количестве узлов отклонение часто бывает больше, чем при большом наборе узлов. Чебышёвские узлы, как видно, например, на рис. 11 — рис.14 и рис.16, позволяют интерполяционному полиному приблизиться к исходной функции лучше, чем равномерно расположенные узлы. Тем не менее, бывают и обратные ситуации, как например на рис.15. Здесь в одном из

интервалов, приближение по чебышёвским узлам оказывается гораздо менее точным, чем приближение по равномерно распределенным узлам. Участки, где приближение по чебышёвским узлам вызывает большую погрешность, есть на каждом из графиков. Обычно, они располагаются в области небольшого количества узлов. Исходя из этого, нельзя точно гарантировать при небольшом количестве узлов, какие именно полиномы Лагранжа будут давать лучшее приближение.

Однако, если обратить внимание на сходимость полиномов Лагранжа к исходной функции, можно заметить, что на всех приведенных примерах, использование чебышёвских узлов гарантирует лучшую сходимость при увеличении числа узлов.

Также, как известно, чебышевские узлы помогают более точно приближать функцию на тех участках, где полиномы Лагранжа, построенные по равномерным узлам, дают паразитные осцилляции. Это гарантировано тем, что будучи приведенными к нормированной форме они имеют наименьшие по модулю экстремумы среди всех нормированных полиномов той же степени [2]. Мы не можем наблюдать этот эффект на графиках, так как для этого необходимы более высокие значения количества узлов интерполяции.

Аппроксимация Паде, формула которой используется в данном задании в качестве интерполируемой функции, является классическим методом рациональной аппроксимации аналитических функций [4]. Разложимые в ряд Тейлора функции, фактически, могут быть представлены приближенно в виде аппроксимации Паде [5], а значит, с помощью этого метода можно, и в противоположную сторону, заменить аналитическую функцию на приближенную. Случайная генерация коэффициентов позволяет добиться получения и в знаменателе, и в числителе многочленов различных степеней, для того чтобы добиться большей вариативности в воспроизведении аналитических функций.

## 2. Интерполяция кубическими сплайнами

Для выполнения данного задания использовался язык Python версии 3.5.2 и его модули `numpy`, `matplotlib`, `csv`, `datetime` и другие. Весь функционал был разбит по назначению на три файла: `main.py`, `splines.py` и `data_parse.py`.

Была разработана функция `cubic_spline_coeff(x_nodes, y_nodes)`, решающая задачу получения коэффициентов кубического сплайна для передаваемых узлов. В этой функции сначала вычисляются значения  $h = x_{i+1} - x_i$  для каждого из подотрезков интерполирования. Далее выражаются коэффициенты  $a = f(x_i)$ . После того как эти значения найдены, составляется матричное уравнение относительно вектора коэффициентов  $c$ . Оно решается с помощью библиотечной функции `numpy.linalg.solve()`. После нахождения  $c$  через него выражаются векторы коэффициентов  $b$  и  $d$ . Функция возвращает объединенную матрицу коэффициентов, состоящую из 4 столбцов ( $a$ ,  $b$ ,  $c$  и  $d$  соответственно) и  $n-1$  строки, где  $n$  — количество точек интерполирования, а  $n-1$  — количество подотрезков.

Следующим этапом решения была разработка двух функций: `qubic_spline(x_nodes, qs_coeff)` и `d_qubic_spline(x_nodes, qs_coeff)`. Обе данные функции генерируют новые функции, зависящие от входных аргументов и имеющие сигнатуру типа `func(x)`.

Функция, возвращаемая функцией `qubic_spline(x_nodes, qs_coeff)`, рассчитывает значение кубического сплайна с заданными ранее коэффициентами `qs_coeff` и точками интерполяции `x_nodes`. При получении значения  $x$  она определяет подотрезок  $i$ , к которому он принадлежит и в соответствии с этим выбирает коэффициенты и вычисляет значение сплайна в данной точке.

Аналогично работает функция, возвращаемая из `d_qubic_spline(x_nodes, qs_coeff)`. Она вычисляет значение производной кубического сплайна в точке  $x$ .

Для дальнейшей интерполяции были получены данные о погоде в Нью-Йорке в районе Central Park [7] в формате csv. В функции *get\_data\_from\_csv()* с помощью модуля csv были выделены только нужные данные о времени и о погоде, после чего получены 2 списка *time* и *temperature* в формате datetime и int соответственно.

По этим спискам далее для наглядности был построен график зависимости температуры от времени без интерполяции (рис. 17).

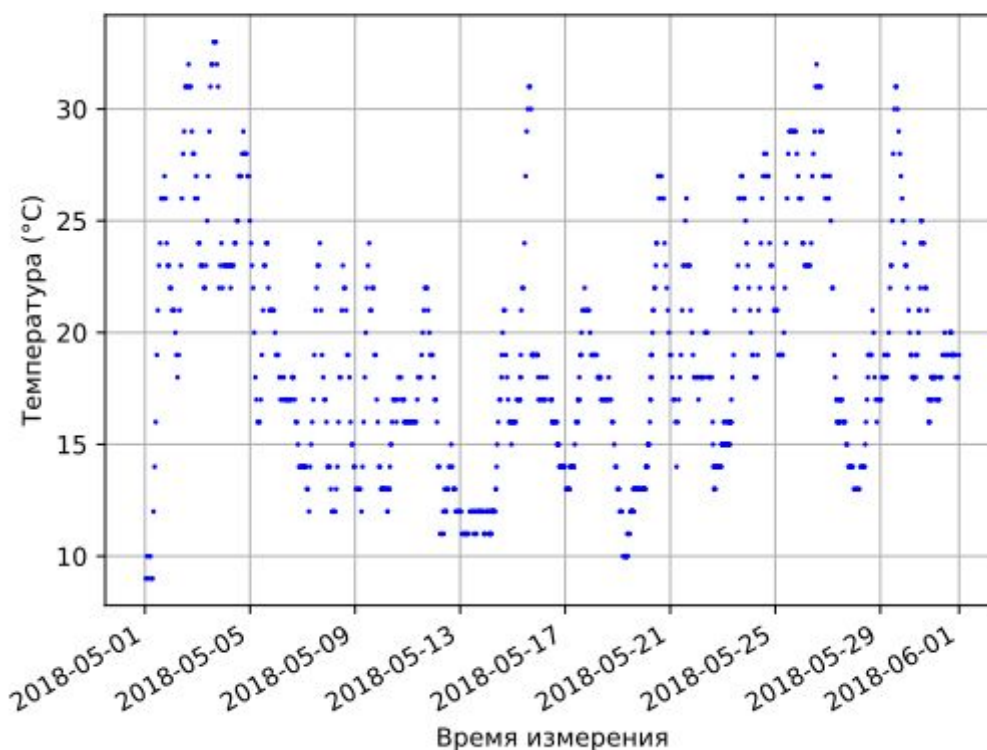


Рис. 17. Точки измерения значений температуры

Далее эти точки были использованы для получения кубического сплайна в функции *create\_spline(time, temperature)*. Внутри данной функции, реализована генерация функции сплайна по заданным точкам, а также его построение на графике в зависимости от времени измерения. При вызове этой функции с массивом всех точек в качестве аргумента был получен график интерполяции по всем точкам (рис.18).

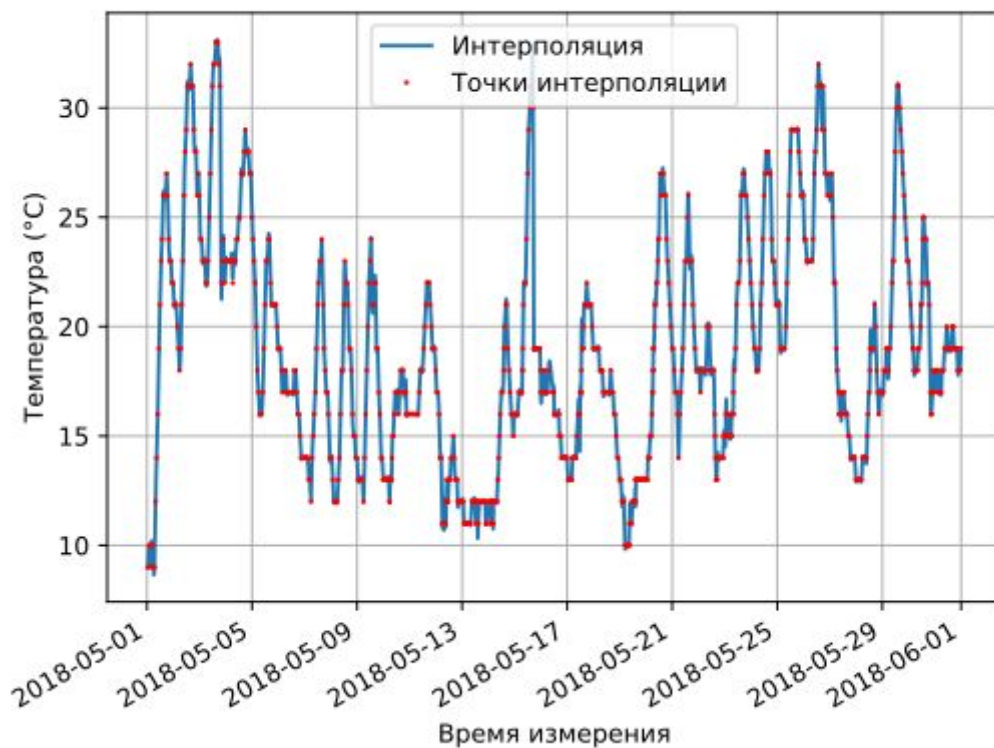


Рис. 18. Интерполяция кубическими сплайнами по всем точкам для температуры в Нью-Йорке в мае 2018 года

После этого, из набора всех точек должно было быть удалено некоторое неизвестное заранее количество точек таким образом, чтобы в выборке остались только точки, полученные в 3, 9, 15 или 21 час без учета минут. Для этого была модифицирована функция *create\_spline(time, temperature)*, которая после этого могла бы принимать еще один аргумент - *allowed* - список допустимых значений для выборки по часам.

Для удаления лишних элементов выборки была написана функция *get\_allowed\_point(time, temperature, allowed)*, которая вызывалась в измененной функции *create\_spline(time, temperature, allowed)*.

По новой выборке был построен график (рис. 19).

Аналогично был сформирован график только по измерениям в 3 и 15 часов (рис. 20).



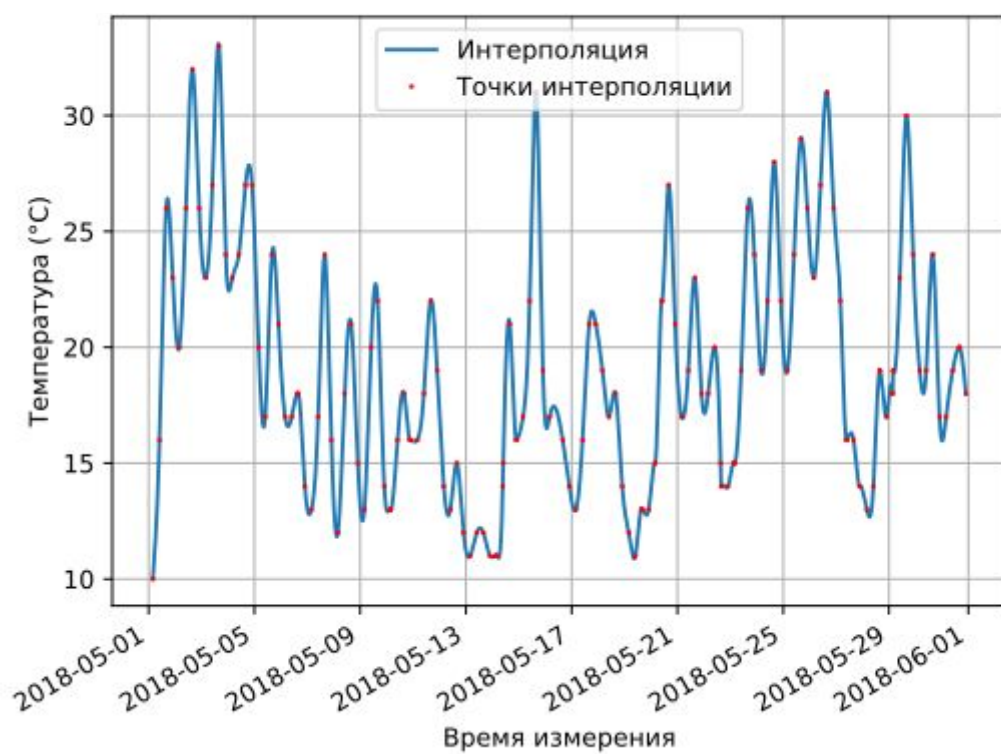


Рис. 19. Интерполяция по измерениям, сделанным в 3, 9, 15 или 21 час

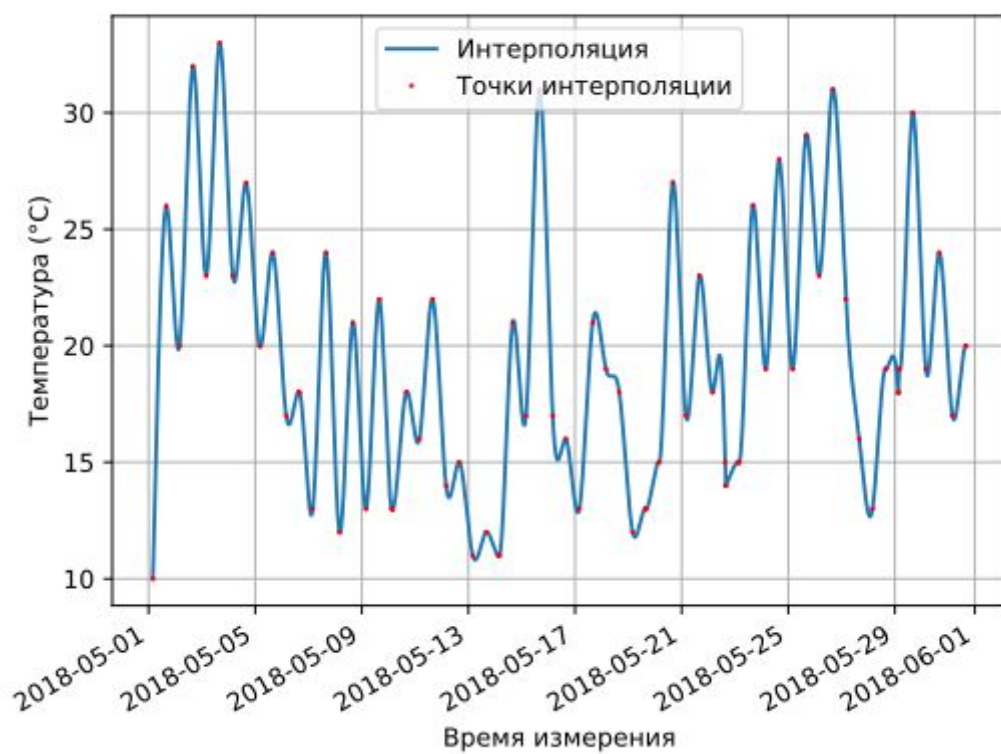


Рис. 20. Интерполяция по измерениям, сделанным в 3 или 15 часов

После этого было проведено сравнение всех полученных функций по значению равномерной нормы. Равномерная норма была получена, исходя из разницы измерений во всех известных 997 точках в исходных данных и по функции кубического сплайна для данной выборки. Аналогично была получена норма для температуры, усредненной по дням. Полученные результаты представлены в таблице 1.

Таблица 1. Равномерные нормы для различных данных

Часы выборки	Количество точек интерполяции	Равномерная норма для температуры в точках изначальных измерений, °C	Равномерная норма для среднесуточной температуры, °C
3, 9, 15, 21	152	10.003	4.5
3, 15	84	11.587	6.1

Исходя из результатов вычисления норм для температуры (таблица 1), можно сделать вывод, что чем меньше точек доступно для интерполяции функции, тем менее точными будут получаться результаты интерполяции в приведенном примере с синоптическими данными.

Как видно из графиков (рис. 18 - 20) и равномерной нормы для среднесуточной температуры (таблица 1), значение среднесуточной температуры, удастся предсказать точнее, чем значение функции в данной точке.

## **Заключение**

### **Задание 1. Интерполяция Лагранжа.**

В задании были сгенерированы функции для вычисления аппроксимации Паде и приближения её с помощью полиномов Лагранжа, построенных по равномерно распределенным узлам и чебышёвским узлам. Были построены и проанализированы графики этих интерполяций, а также графики равномерных норм в зависимости от количества узлов интерполяции. Были выявлены особенности применения различных типов вычисления узлов для полиномов Лагранжа, а именно, что применение узлов Чебышёва для интерполяции полиномами Лагранжа даёт более точный результат при большем количестве узлов, чем применение равномерно распределенных узлов.

### **Задание 2. Интерполяция кубическими сплайнами.**

Были созданы функции для получения функции кубического сплайна и его производной для любых корректных исходных данных. С помощью полученных функций были интерполированы синоптические данные при различных наборах интерполяционных точек. Было выяснено, что уменьшение количества точек интерполяции, увеличивает равномерную норму между интерполирующей и исходной функциями. Также был сделан вывод, что средние значения функции на некоторых допустимых интервалах, приближаются лучше, чем отдельно взятые её значения.

## Список использованных источников

1. *Библиотека для визуализации данных* [Электронный ресурс]. – Режим доступа: <https://github.com/gonum/plot>. – A repository for plotting and visualizing data. – (Дата обращения: 17.03.2019).
2. Першин А.Ю. *Лекции по вычислительной математике (черновик)* [Электронный ресурс] // Кафедра РК6 (Системы автоматизированного проектирования), МГТУ им. Н.Э. Баумана, 4 марта 2019 г.
3. В. А. Скворцов *Примеры метрических пространств* [Электронный ресурс] // Издательство Московского центра непрерывного математического образования, Москва, 2002
4. *Аппроксимация Паде* [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Аппроксимация\\_Паде](https://ru.wikipedia.org/wiki/Аппроксимация_Паде) – (Дата обращения: 17.03.2019).
5. Оганесян Д. А. *О рациональных функциях с двумя критическими точками максимальной кратности* [Электронный ресурс] // Московский государственный университет им. М. В. Ломоносова
6. Бейкер Дж., Грейвс-Моррис П. *Аппроксимации Паде*. — М.: Мир, 1986.
7. *Weather in New York / Central Park (AWS)* [Электронный ресурс]. - Режим доступа: [https://rp5.ru/Weather\\_in\\_New\\_York\\_Central\\_Park\\_\(AWS\)](https://rp5.ru/Weather_in_New_York_Central_Park_(AWS)) - (Дата обращения: 17.03.2019).