Исследование непараметрического метода восстановления функциональной зависимости с адаптивно-оптимальной настройкой

# Оглавление

[Оглавление 1](#_Toc134359298)

[Введение 1](#_Toc134359299)

[Постановка задачи 1](#_Toc134359300)

[Цели 2](#_Toc134359301)

[Теоретические сведения 2](#_Toc134359302)

[Разработка инструментария 6](#_Toc134359303)

[Исследовательская часть 8](#_Toc134359304)

[Какие были проблемы 16](#_Toc134359305)

[Сравнение с другими методами 17](#_Toc134359306)

[Дальнейшее развитие 58](#_Toc134359307)

[Выводы 59](#_Toc134359308)

[Список литературы 59](#_Toc134359309)

# Введение

В современном мире ежедневно собирается огромное количество информации, и с каждым днём объёмы растут. Одним из значимых факторов, влияющих на качество данных, являются шумы. Шумы искажают наблюдаемые значения, тем самым снижая точность результатов, получаемых вследствие их анализа. Существующие на сегодняшний день решения наряду с определёнными преимуществами обладают некоторыми недостатками и зачастую требуют от пользователя знаний, необходимых для подбора оптимальных параметров, и непосредственного участия в процессе аппроксимации. В данной работе мы реализуем алгоритм аппроксимации, за основу которого был взят алгоритм локальной полиномиальной регрессии с помощью метода наименьших квадратов из статьи М. К. Семёнова «Восстановление функциональной зависимости с минимальной вычислительной структурой». Этот алгоритм работает эффективнее стандартного метода наименьших квадратов, однако, как и большинство локальных методов требует предварительной настройки. Нашей целью будет исследовать данный алгоритм на предмет возможности, в различных аспектах улучшения и автоматизации его работы.

# Постановка задачи

* Разработать ПО для генерации разнообразных наборов данных (с разными шагами, видами шумов, исходными функциями, наличием выбросов и т.п.), над которыми будут проводиться эксперименты.
* Разработать ПО для анализа результатов работы алгоритма (средства визуализации и другие необходимые элементы).
* Изучить алгоритм, приведённый в научной публикации [1] и предложить его усовершенствования.
* Внести предложенные модификации в метод и исследовать результаты.
* Провести сравнительный анализ эффективности модифицированного алгоритма с альтернативными(другими) методами сглаживания.

# Цели

* Исследовать ранее разработанный алгоритм аппроксимации данных на предмет возможности внесения модификаций для улучшения его скорости, точности и/или робастности и сравнить модифицированную версию с другими методами аппроксимации.

# Теоретические сведения

**Метод наименьших квадратов**

Пусть имеется набор точек, отражающий зависимость между двумя переменными. Попытаемся представить интересующую нас зависимость с помощью кривой линии. Разумеется, такая линия может дать только приближенное представление о форме реальной статистической связи. Постараемся сделать это приближение наилучшим. Оно будет тем лучше, чем меньше исходные данные будут отличаться от соответствующих точек, лежащих на линии. Степень близости может быть выражена величиной суммы квадратов отклонений реальных значений от значений, расположенных на кривой.

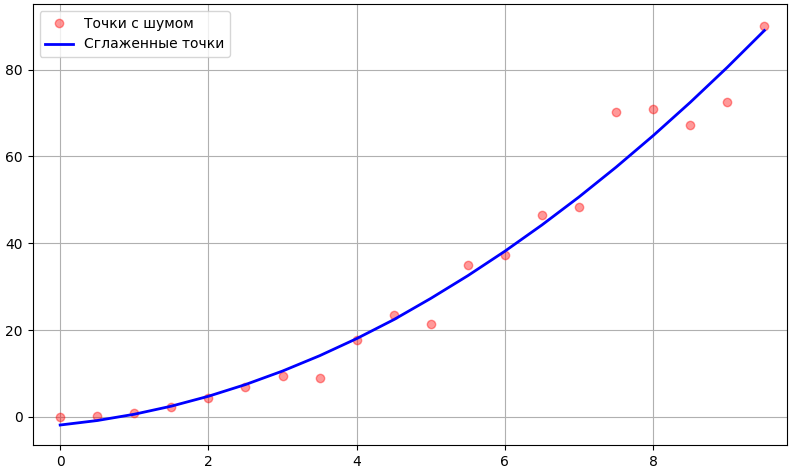


Рисунок 1 – Пример кривой, проведённой через точки, имеющие нормально распределённое отклонение от истинного значения с помощью МНК

Пусть имеется n значений некоторой переменной y (это могут быть результаты наблюдений, экспериментов и т. д.) и соответствующих переменных x. Задача заключается в том, чтобы взаимосвязь между y и x аппроксимировать некоторой функцией f(x,b), известной с точностью до некоторых неизвестных параметров b, то есть фактически найти наилучшие значения параметров b, максимально приближающие значения f(x,b) к фактическим значениям y. Фактически это сводится к случаю «решения» переопределенной системы уравнений относительно b:

Соответственно, отклонения наблюдаемых значений y от модельных f(x,b) предполагается уже в самой модели. Сущность классического МНК заключается в том, чтобы найти такие параметры b, при которых сумма квадратов отклонений (ошибок, для регрессионных моделей их часто называют остатками регрессии) будет минимальной:

где RSS — англ. Residual Sum of Squares определяется как:

В общем случае решение этой задачи может осуществляться численными методами оптимизации (минимизации). В этом случае говорят о нелинейном МНК (NLS или NLLS — англ. Non-Linear Least Squares). Во многих случаях можно получить аналитическое решение. Для решения задачи минимизации необходимо найти стационарные точки функции RSS(b), продифференцировав её по неизвестным параметрам b, приравняв производные к нулю и решив полученную систему нормальных уравнений:

Например, для полинома второй степени эта система будет выглядеть следующим образом

Исходным этапом для этого является подбор вида функции, отображающей статистическую связь. Тип функции в каждом конкретном случае можно подобрать путем анализа на исходных данных подходящей, т. е. достаточно хорошо приближающей эти данные, линии, однако на практике вид зависимости между переменными известен не всегда, а подбор отдельной функции для набора данных зачастую является трудоёмким процессом.

Автор публикации «Восстановление функциональной зависимости с минимальной вычислительной структурой»[1] для решения этой проблемы ушел от параметризованного подхода решения задачи и локализовал метод наименьших квадратов, что дало возможность работать с любыми данными, никак не учитывая предварительную информацию о поведении, формах, законах и любых других вводных, которые на практике как правило отсутствуют. Её суть заключается в том, что при сглаживании -й точки в качестве области определения выбирается интервал значений в окрестности этой точки такой длины, чтобы он хорошо аппроксимировался параболой. Такой подход позволяет восстанавливать данные без подбора аппроксимирующей функции, корректируя длину интервала аппроксимации.

Полагая, что аппроксимируемая точка находится в середине интервала аппроксимации, будем называть плечом количество точек с каждой из сторон от аппроксимируемой точки. Тогда .

Уравнение параболы с плечом , аппроксимирующей -ю точку будет иметь вид

где неизвестные параметры находятся с помощью МНК в результате решения следующей системы нормальных уравнений:

После вычисления коэффициентов значения оценки можно найти по формуле

Визуализация процесса аппроксимации представлена на следующем графике

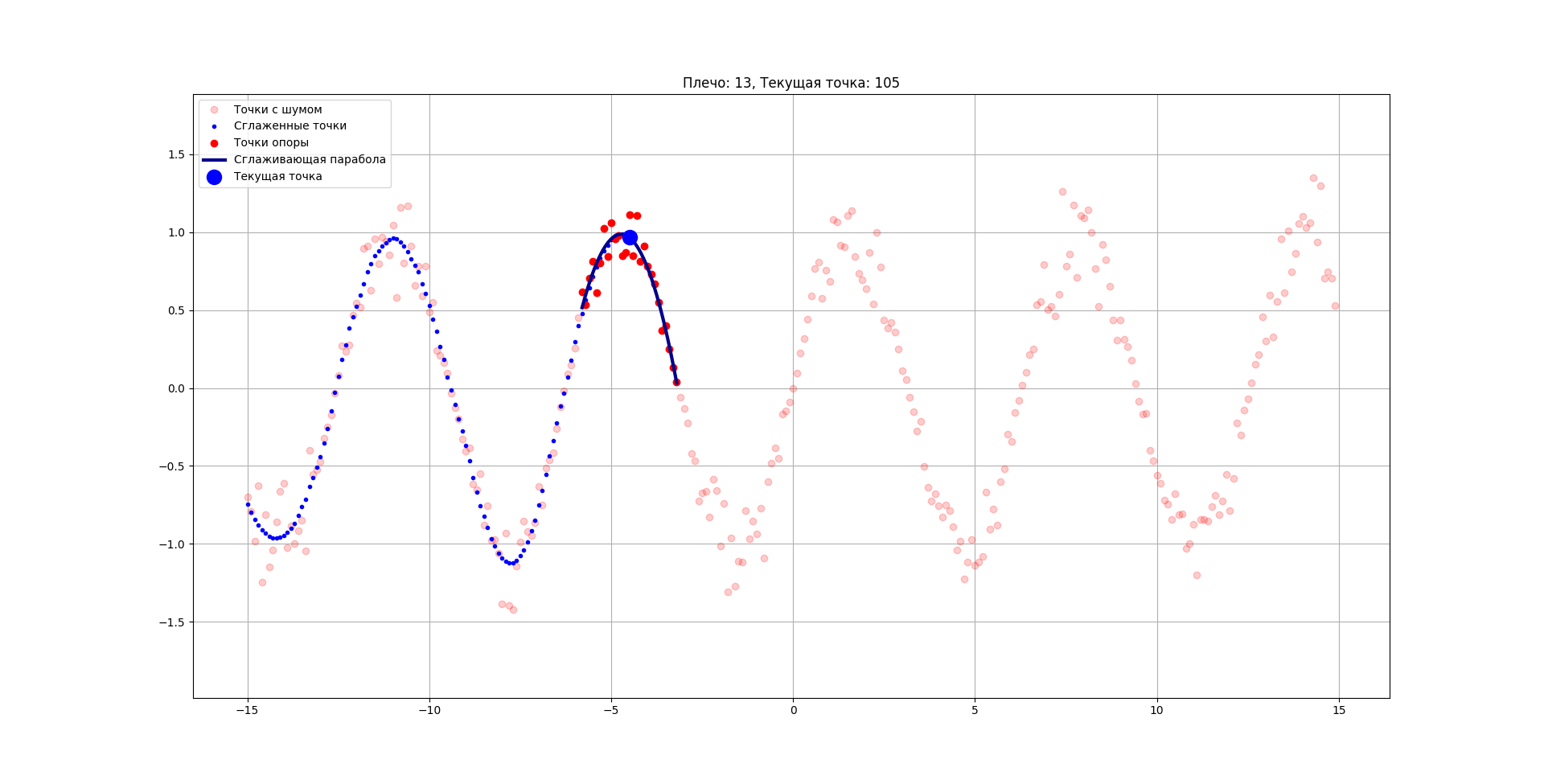


Рисунок 2 – Визуализация процесса аппроксимации

При данном подходе задача сводится к поиску оптимального плеча, которому соответствует наименьшее отклонение оценок от реальных значений функции. Минимизируемая функция при этом приобретает вид

Где L – длина промежутка аппроксимации, – реальное значение функции, – оценка полученная при аппроксимации параболой с плечом L, n – количество точек в сглаживаемом наборе данных. Оптимальное плечо соответственно определяется как

# Разработка инструментария

Для программной реализации всех необходимых элементов был выбран язык Python3 ввиду обилия научных библиотек и возможности быстрой разработки прототипа программы. После окончания разработки планируется провести рефакторинг программы с использованием других языков обеспечивающих лучшее быстродействие.

В рамках работы была разработано оконное приложение, имеющее два основных блока: генератор и аппроксиматор. От генератора требовалась возможность максимально гибкой генерации различных наборов данных с шумами. От аппроксиматора требовались возможности практического применения различных алгоритмов (приведённого в статье[1] алгоритма, модифицированного алгоритма и альтернативных существующих алгоритмов аппроксимации) для восстановления функциональной зависимости, а также качественной визуализации процесса восстановления и сопутствующих процессов.

**Генератор**

Генератор позволяет создавать наборы точек в двумерном пространстве на основе элементарных математических функций. Точки записываются в таблицу Microsoft Excel. Генератор обладает большим количеством настраиваемых параметров, одними из которых являются функция на основе которой генерируются точки, интервал на котором будут лежать точки, количество точек. Распределение точек на интервале можно регулировать - они могут быть распределены равномерно или со случайным шагом. Помимо этого, присутствует возможность генерации промежутков, заполненных лишь частично, процент их заполненности может меняться динамически в заданных границах. После генерации на полученный набор накладывается шум. Он может быть в процентах от изначального значения или в виде абсолютной величины, может иметь равномерное или нормальное распределение с заданными параметрами. К обычным шумам можно также добавить выбросы, частота которых регулируема, а их значения так же могут быть относительны или абсолютны. Интерфейс данного блока программы представлен на рисунке ниже.

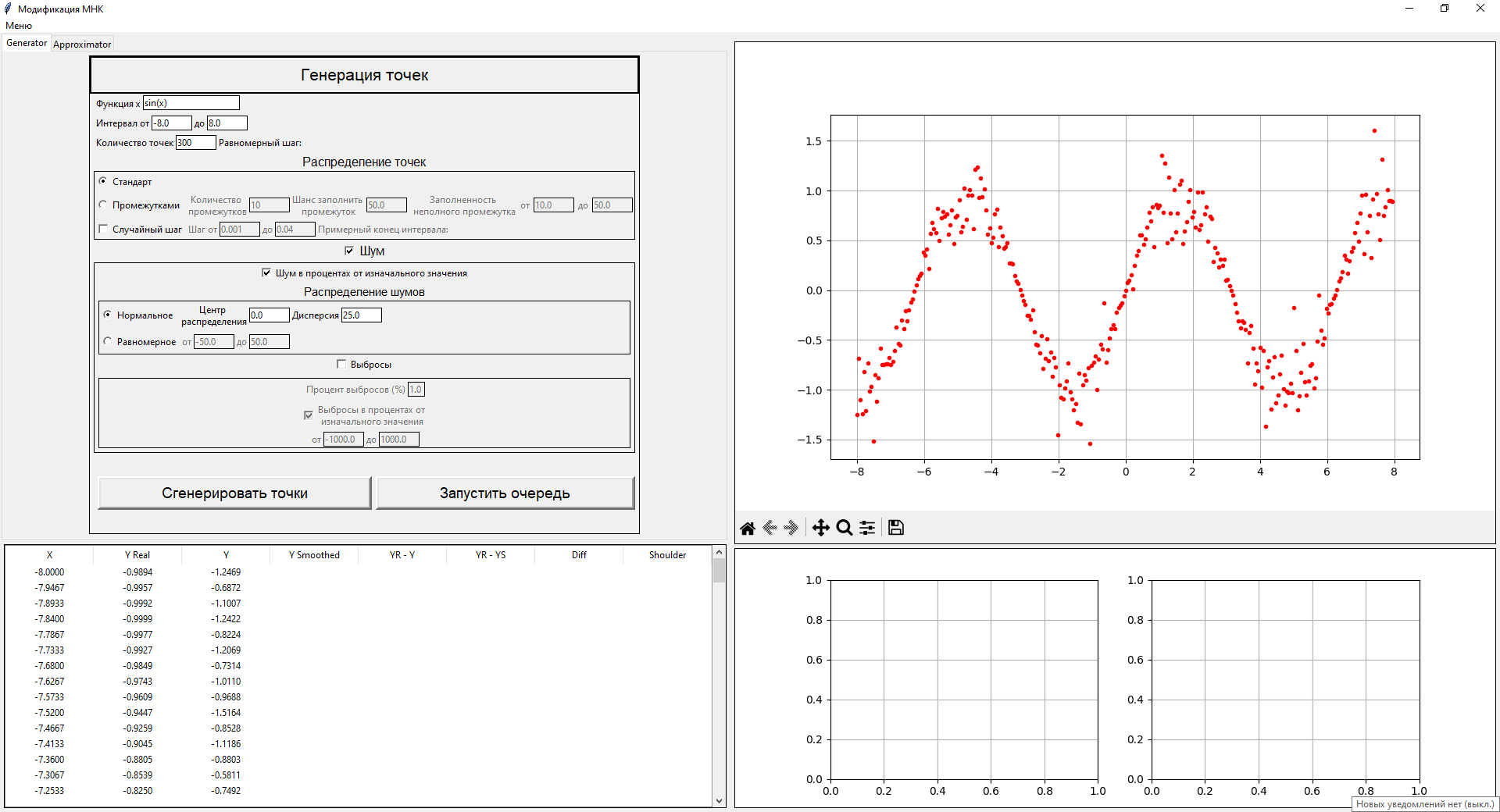


Рисунок 3 - Интерфейс блока "Генератор" программы

**Аппроксиматор**

В блоке аппроксиматор был программно реализован приведённый в статье[1] алгоритм возможностью задания различных параметров аппроксимации, таких как выбор вида плеча аппроксимации – фиксированное, или динамическое, мануальное задание величины плеча, возможность нахождения оптимального фиксированного плеча с помощью последовательного перебора всех значений, дихотомии или метода золотого сечения (параметры которых задаются вручную чтобы не попасть в локальный минимум). Реализована возможность сглаживания данных динамическим плечом с подбором оптимального вектора плеч с помощью перебора всех вариантов, случайного перебора, дихотомии, метода золотого сечения, а также генетического алгоритма. Присутствует возможность остановить процесс подбора оптимального плеча, не дожидаясь его полного завершения чтобы получить наилучший вычисленные на текущий момент результат. Кроме этого для динамического и фиксированного плеча имеется возможность сглаживать концы графика с помощью перемещения точки амортизации от центра к краю. Реализована возможность выбирать разные методы подсчёта отклонения такие как сумма квадратов, сумма модулей и среднеквадратическое отклонение, присутствует возможность не учитывать при подсчёте отклонения концы графика. Параметры визуализации процесса аппроксимации позволяют соединить сглаженные точки линией, показать настоящую функцию с помощью которой были сгенерированы точки с шумами, автоматически сохранять график с результатом аппроксимации по выбранному пути. Предоставлена возможность аппроксимации данных альтернативными алгоритмами в рамках одного графика. Существует возможность показать анимацию процесса аппроксимации в реальном времени. Интерфейс блока аппроксиматор приведён на рисунке ниже.

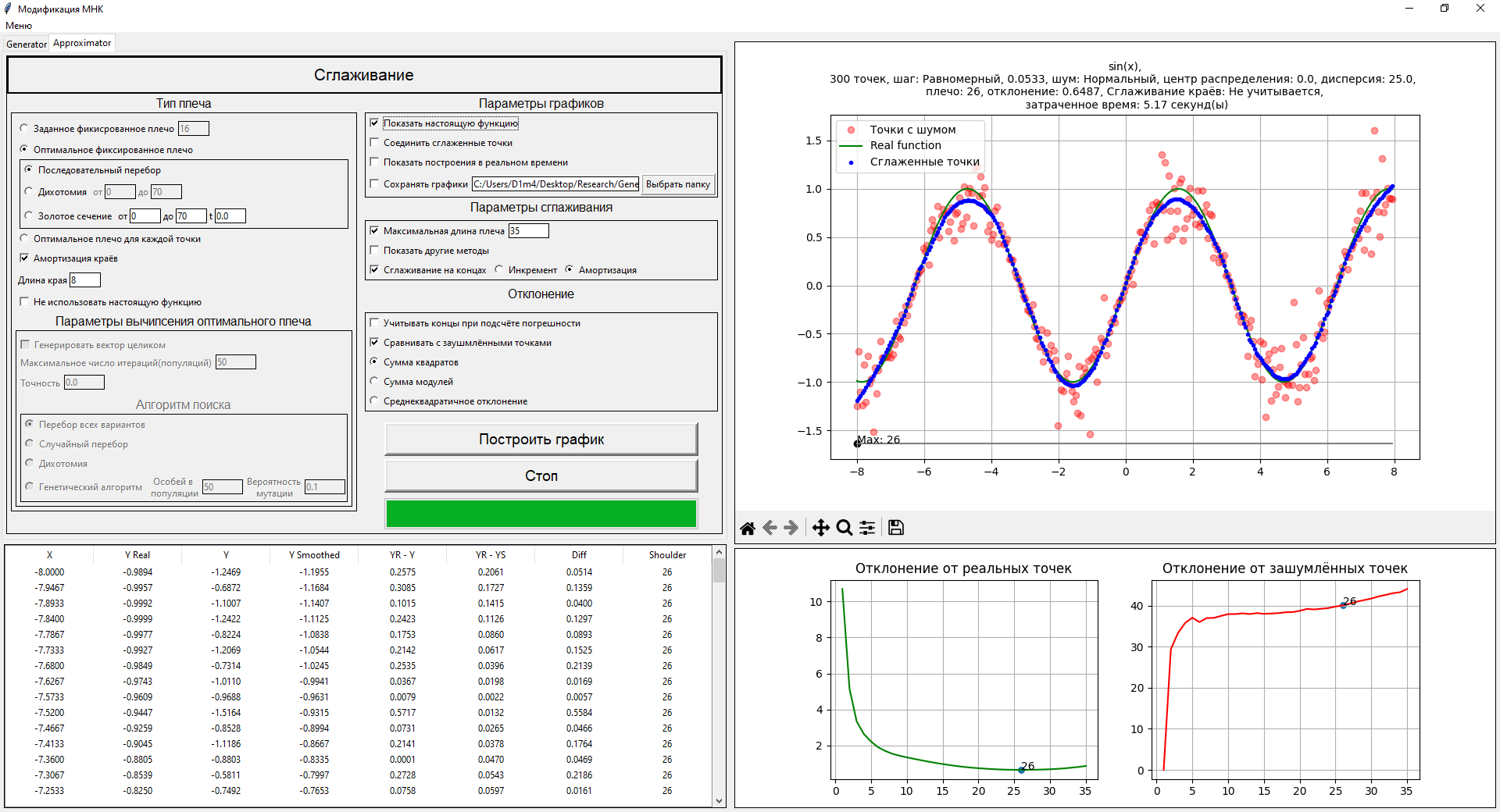
****

Рисунок 4 - Интерфейс блока "Аппроксиматор" программы

# Исследовательская часть

После реализации основной части необходимых инструментов мы приступили к исследованиям.

**Исследование влияния длины плеча параболы на точность аппроксимации. Автоматический подбор оптимального плеча по известным значениям реальной функции.**

Длина плеча оказывает прямое влияние на результаты апроксимации, потому что от неё зависит количество учтённых точек и, соответственно, коэффициенты аппроксимирующей параболы. Так, если плечо слишком мало, то точки смещаются недостаточно, и в данных остаётся значительная часть шумов. При слишком большой длине плеча, алгоритм пытается аппроксимировать большую часть точек одной параболой, из-за чего малые относительно плеча изгибы не могут быть полноценно учтены и происходит “пересглаживаение”, не давая хорошего результата. Разницу точности аппроксимации при различных значениях плеч можно увидеть на графиках, приведённых ниже.

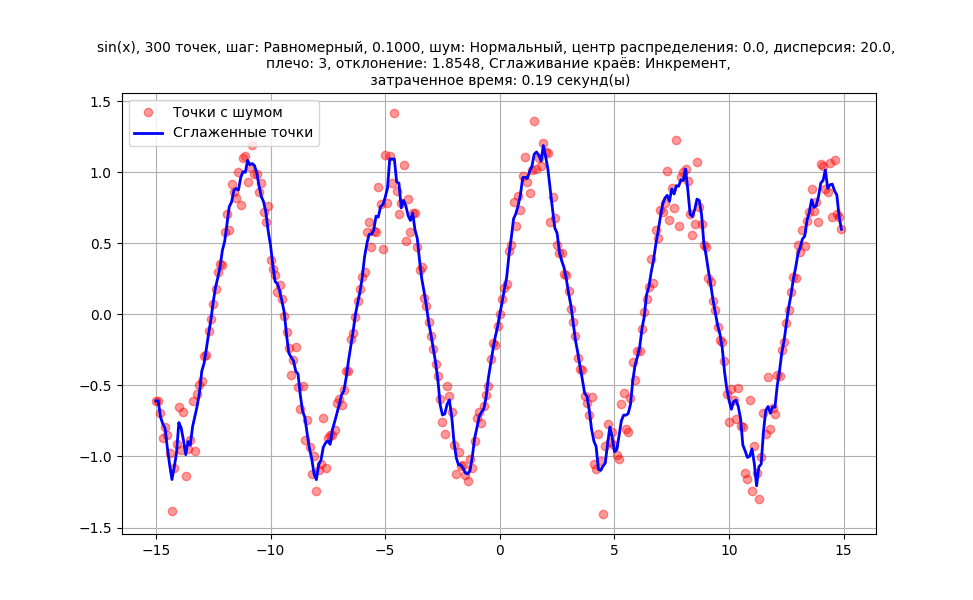


Рисунок 5 – Эффективность аппроксимации при плече меньше оптимального

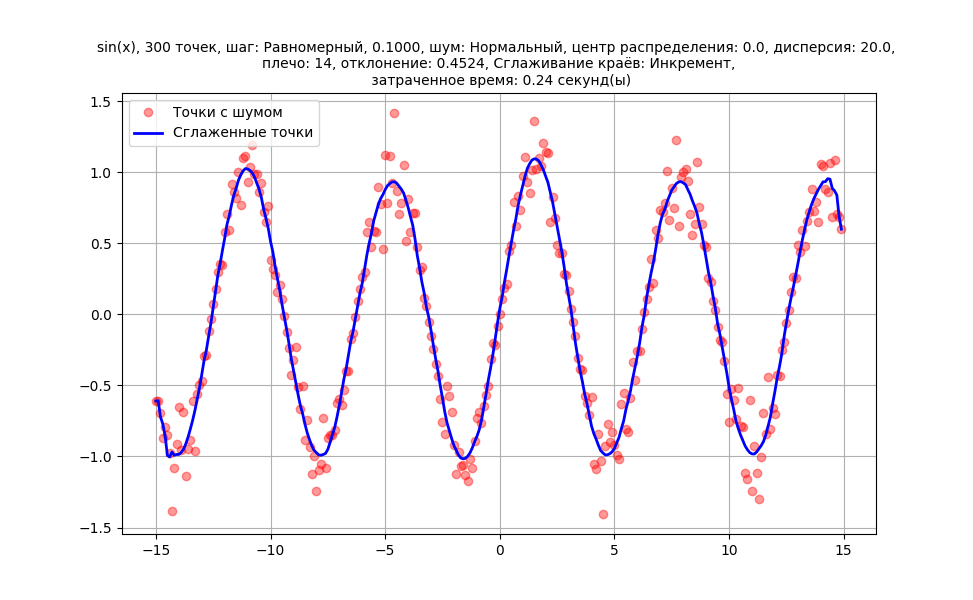


Рисунок 6 – Эффективность аппроксимации при оптимальном плече

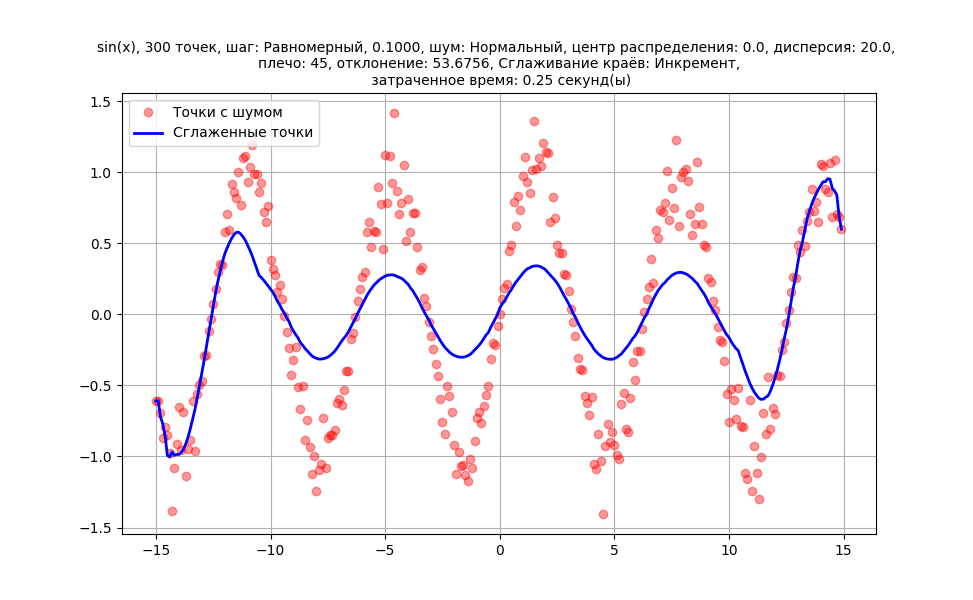


Рисунок 7 - Эффективность аппроксимации при плече больше оптимального

После ряда экспериментов с различными длинами плеч используемых для сглаживания одних и тех же наборов данных у нас появилась гипотеза о том, что для каждого набора данных существует одно единственное плечо при котором отклонение минимально. Для проверки нашей гипотезы в ПО была добавлена возможность постройки графика зависимости отклонения оценки от реальных значений функции (по вертикали) от длины фиксированного плеча (по горизонтали). Пример такого графика приведён ниже.

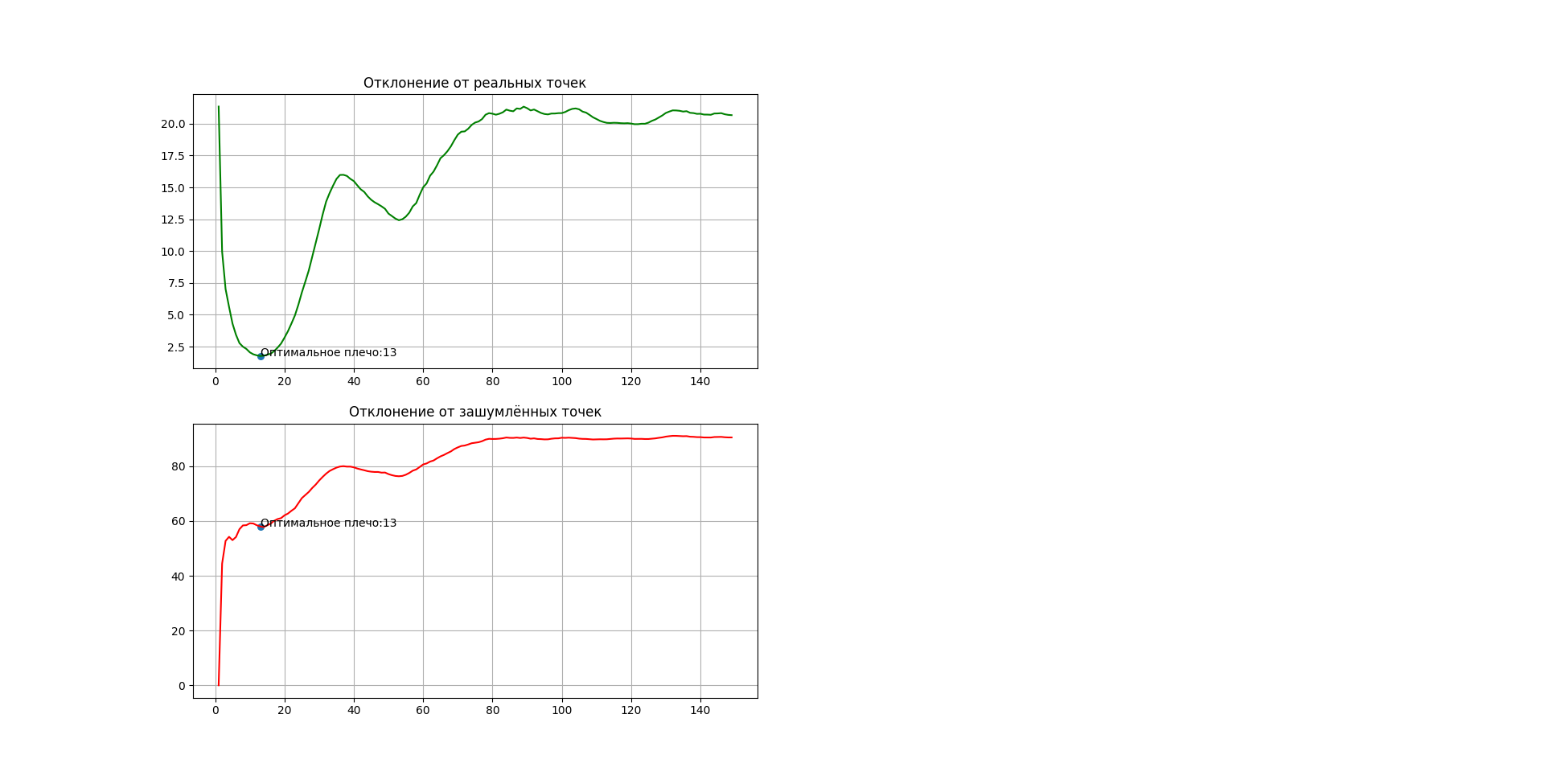


Рисунок 8 – Пример графика отклонения оценки от реальных значений функции

**После сравнения графиков отклонений различных наборов данных, мы пришли к выводу, что минимум функции отклонения всегда единственен и является её первым слева локальным минимумом. Расстояние от точки минимума до начала графика (плечо=1) при этом прямо пропорционально величине наложенных на данные шумов, что логично, ведь чем больший шум применён к точкам, тем большего сглаживания, предоставляемого большим плечом, они требуют. После этого минимума функция отклонения может вести себя по-разному в зависимости от исходной функции с помощью которой были сгенерированы данные, не достигая при этом значения первого минимума. Для быстрой минимизации функции отклонения было реализовано несколько различных методов, включающих дихотомию и метод золотого сечения (выбор методов был обусловлен текущей учебной программой, позволив применить полученные теоретические знания на практике).**

**Автоматический подбор оптимального плеча без реальных значений функции**

**Первой попыткой автоматизировать алгоритм стала реализация поиска оптимального фиксированного плеча без знаний о реальных значениях или виде исходной функции. В качестве анализируемого критерия было выбрано отклонение оценки от зашумлённых точек, как наиболее интуитивный и лёгкий в вычислении вариант.**

**Была программно реализована возможность вывода графика зависимости этого критерия от длины плеча. Затем были проанализированы графики отклонения от зашумлённых точек в совокупности с графиками отклонений от реальных точек для различных наборов данных, что позволило установить следующее наблюдение: график отклонения от зашумлённых точек для всех наборов данных имеет пологий интервал в левой части графика, причём середина этого интервала примерно соответствует минимуму графика отклонений оценки от реальных значений функции. Пример совокупности этих зависимостей приведён на графике ниже.**

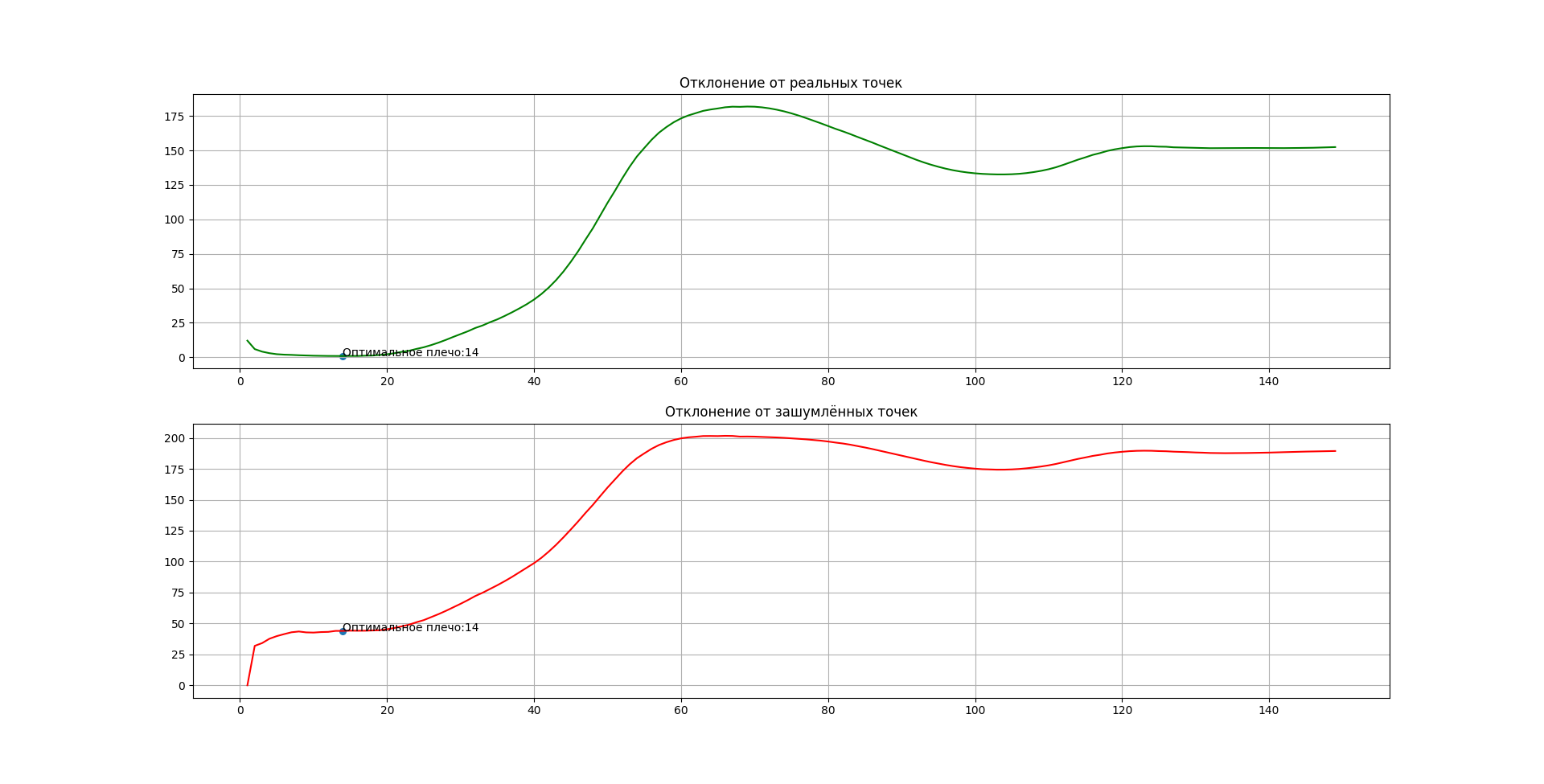
****

Рисунок 9 – сравнение графиков отклонения оценки от реальных и зашумлённых точек

**На основании этого наблюдения нами были предприняты попытки реализации алгоритма полностью автоматического подбора оптимального плеча, путём вычисления приближённых значений первой и второй производных функции отклонения от зашумлённых точек, однако попытки прийти к чёткому аналитическому решению на настоящий момент успехом не увенчались, ввиду немонотонности функции отклонения зашумлённых точек в окрестности искомой точки. По этой причине в качестве временного прототипа алгоритма был разработан следующий метод: для идентификации пологого интервала эмпирически было выбрано константное значение производной при значениях ниже которого интервал принимается пологим, а в качестве решения выбиралось середина этого интервала. Этот метод примитивен и неприменим в случае малого количества точек или малых значений шумов, но в остальных случаях он даёт пусть не точное, но приближённое к оптимальному значение плеча. В будущих исследованиях методологию планируется развивать, прибегая к использованию более обоснованных и эффективных методов. Пример работы примитивного алгоритма приведён на графике ниже.**

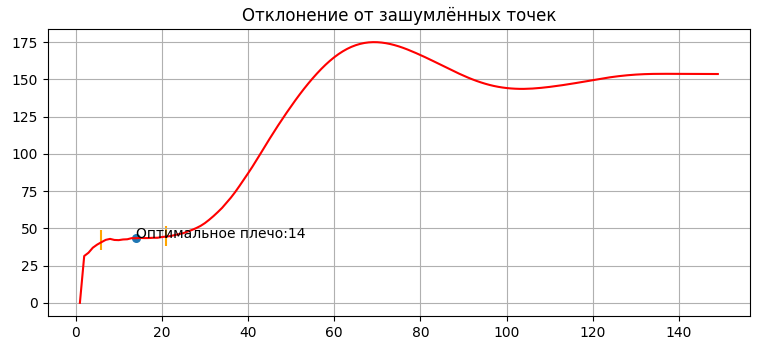
****

Рисунок 10 – Пример работы примитивного алгоритма поиска оптимального фиксированного плеча

**Исследование эффективности аппроксимации с помощью динамического плеча**

**В ходе исследований мы пришли к выводу, что существуют функции, которые не могут быть восстановлены корректно даже при оптимальном значении фиксированного плеча (по крайней мере не на всех промежутках). Для решения данной проблемы было предложено подбирать плечо для восстановления каждой точки отдельно.**

**Сглаживанием с помощью динамического плеча в данном контексте мы будем называть сглаживание с помощью вектора плеч , где -й точке соответствует одно значение плеча . Оптимальный вектор будет состоять из оптимальных плеч , которое можно определить как**

**Тогда**

**Критерий минимизации при этом примет вид**

**Для реализации подбора вектора оптимальных плеч были использованы те же методы что и для подбора фиксированного плеча, модифицированные для поиска лучшего плеча в каждой точке. Кроме того, был реализован дополнительный метод подбора с помощью генетического алгоритма. С помощью генетического алгоритма в целях эксперимента также был реализован метод подбора вектора плеч целиком, а не отдельно в каждой точке, однако он проявил себя значительно хуже, чем аналогичные методы, подбирающие точки отдельно.**

**Несмотря на то, что при работе с простыми функциями по типу синуса, увеличение точности при использовании динамического плеча в сравнении с фиксированным не так значительно, при аппроксимации нетривиальных зависимостей, динамическое плечо имеет значительное преимущество, которое увеличивается с ростом скорости осцилляции, и неоднородности восстанавливаемой функции. Графики со сравнением результатов, полученных с помощью фиксированного и динамического плеча приведены ниже.**

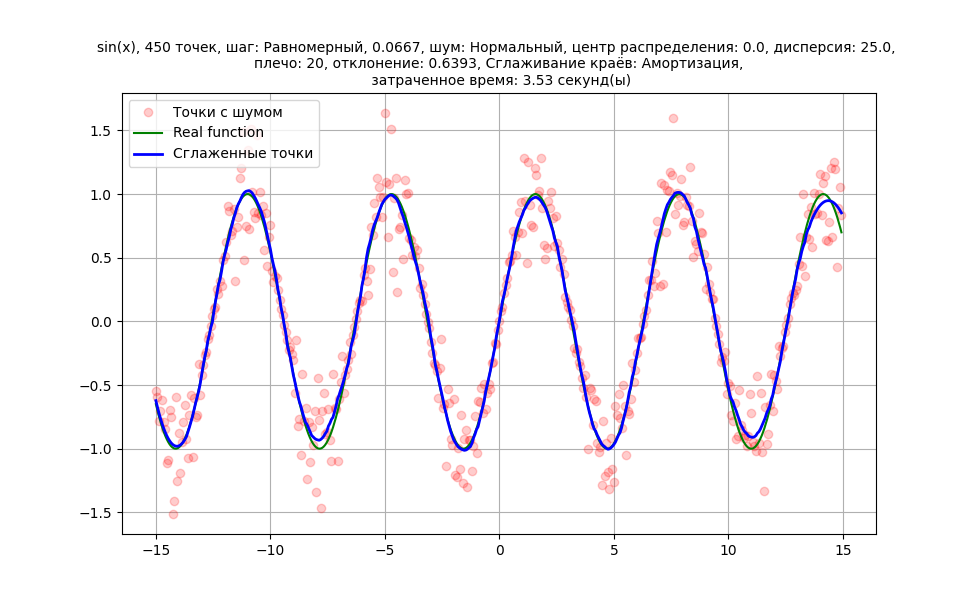


Рисунок 11 – Аппроксимация простой функции оптимальным фиксированным плечом



Рисунок 12 – Аппроксимация простой функции с помощью вектора оптимальных плеч

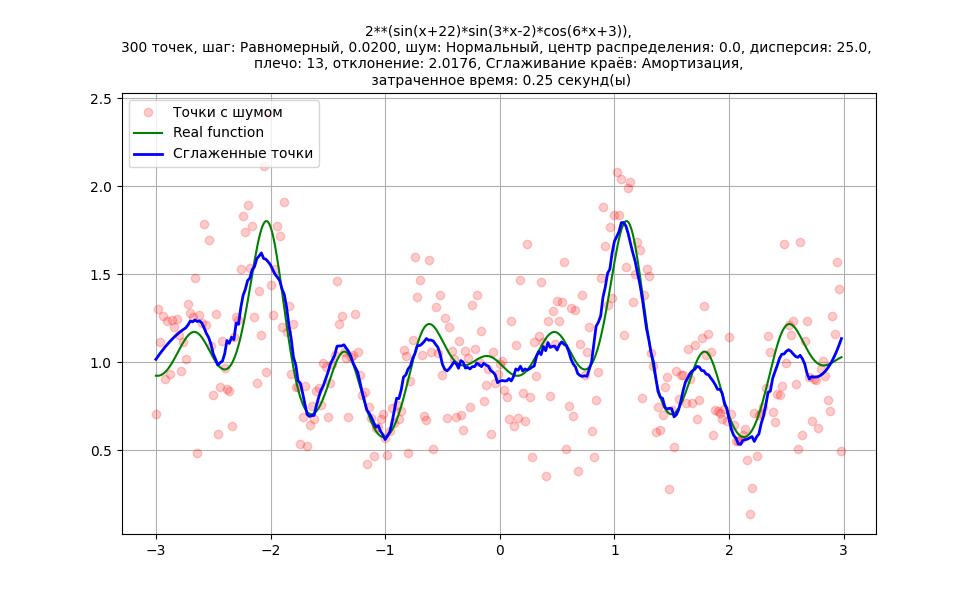


Рисунок 13 - Аппроксимация сложной функции оптимальным фиксированным плечом

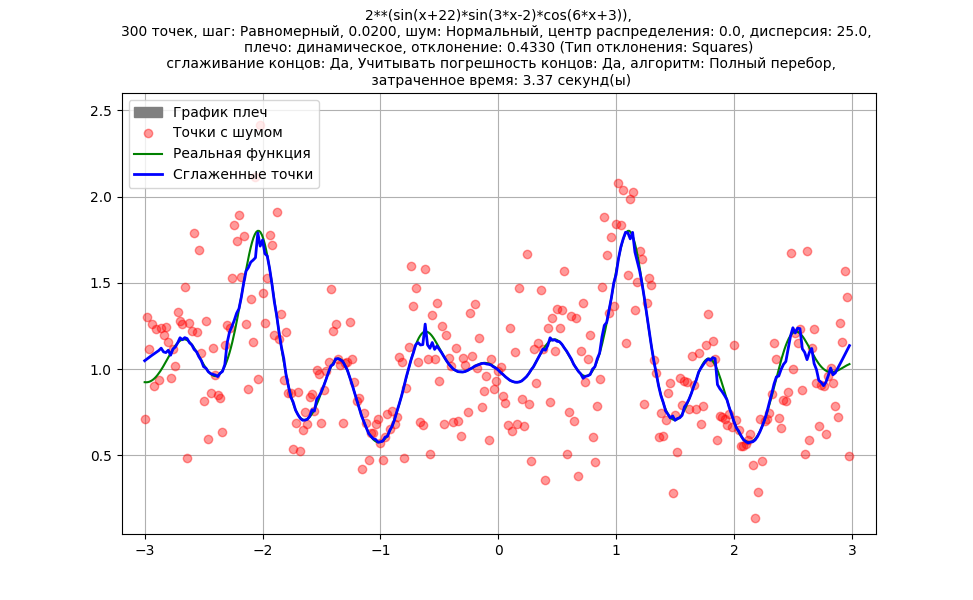


Рисунок 14 – Аппроксимация сложной функции с помощью вектора оптимальных плеч

**Исследование влияния различных методов подсчёта отклонения оценки от реальной функции на результат аппроксимации**

**Сначала нами была исследована эффективность сглаживания фиксированным произвольным плечом при различных методах подсчёта отклонения оценки от реального значения функции, а конкретно – сумма модулей отклонений и среднеквадратичное отклонение. Проведя исследования, мы пришли к выводам что разные методы подсчёта погрешности дают разные итоговые результаты, однако к однозначному выводу о их влиянии на конечный результат мы не пришли, ведутся исследования.**

**Предстоящие исследования**

* Реализовать возможность оптимального подбора точки амортизации
* Реализовать возможность автоматического подбора наилучшего фиксированного плеча параболы основываясь на реальных значениях функции
* Реализовать возможность подбора оптимальных значений плеча параболы для каждой точки основываясь на реальных значениях функции
* Реализовать возможность автоматического подбора наилучшего фиксированного плеча без реальных значений функции
* Реализовать возможность разбиения функции на интервалы и последующего сглаживания каждого интервала фиксированным оптимальным для него плечом
* Реализовать возможность подбора оптимальных значений плеча для каждой точки без реальных значений функции
* Реализовать систему весов для точек плеча параболы
* Реализовать возможность автоматического подбора оптимальной степени полинома для сглаживания конкретного фрагмента
* Расширить метод на многомерное пространство
* Расширить метод на комплексную плоскость

# Какие были проблемы

**Теоретическая база**

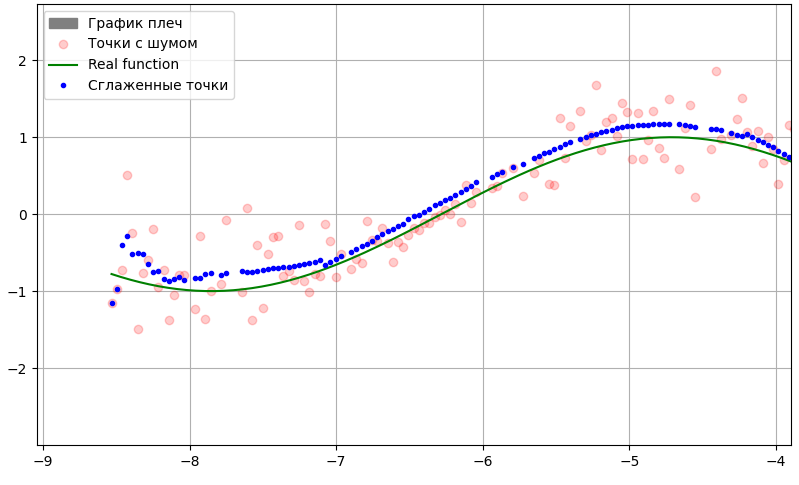
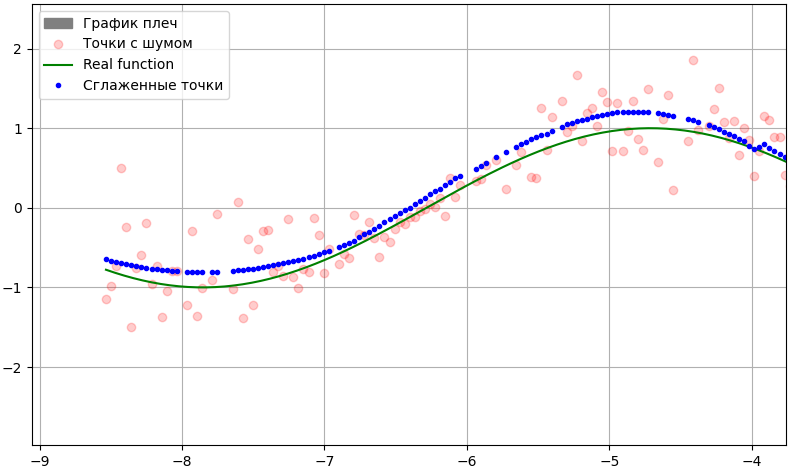
**В ходе работы мы столкнулись со значительным количеством проблем. Главная проблема заключалась в математической базе необходимой для понимания и корректного решения поставленных задач. Чтобы заполнить пробелы в знаниях, активно изучается литература по данной тематике.**

**Недостаточное количество опций генератора точек**

**В ходе работы мы пришли к выводу, что простого наложения шума на сгенерированные точки недостаточно для получения объективных результатов, поэтому возможности генератора точек были расширены опциями генерации точек с различными видами шага – равномерным, случайным и промежутками, возможность применять шум, распределённый нормально, а не равномерно, а также была добавлена возможность задавать величину шумов и выбросов в процентном значении, а не в абсолютном.**

**Сглаживание точек по краям**

**Точки, находящиеся на краях интервала, не всегда могли быть сглажены заданным плечом, поскольку сглаживаемая точка в реализованном нами методе находится в середине фрагмента сглаживающей параболы, некоторые точки просто не имели достаточного количества соседних точек с одной из сторон. Для решения этой проблемы было разработано 2 подхода. Первым было добавление опции игнорирования точек на концах. Они не сглаживались и не учитывались при расчётах отклонения. Однако такой подход показал себя не очень эффективно, так как при большой длине плеча относительно количества точек сглаживалась лишь малая часть от всех данных. Следующим подходом к решению данной проблемы стал выбор максимального из возможных плеч для точки, если заданное плечо больше соответствующего значения. Благодаря этому подходу удалось достигнуть сглаживания точек на краях графика хоть и менее точного.**

**** ****

**Поиск оптимального плеча без данных исходной функции**

**Уже расписано в исследовательской части.**

**Генерация вектора оптимальных плеч целиком**

**Генерация вектора целиком занимает слишком много времени и не имеет достаточной точности.**

# **Сравнение с другими методами**

Сравнение реализованного алгоритма с альтернативными методами сглаживания выявило значительные преимущества, заключающиеся в отсутствии необходимости подбора оптимальных параметров для выполнения корректного сглаживания, высокой точности, а также в высокой устойчивости к выбросам, с которыми не справляется большинство других методов.

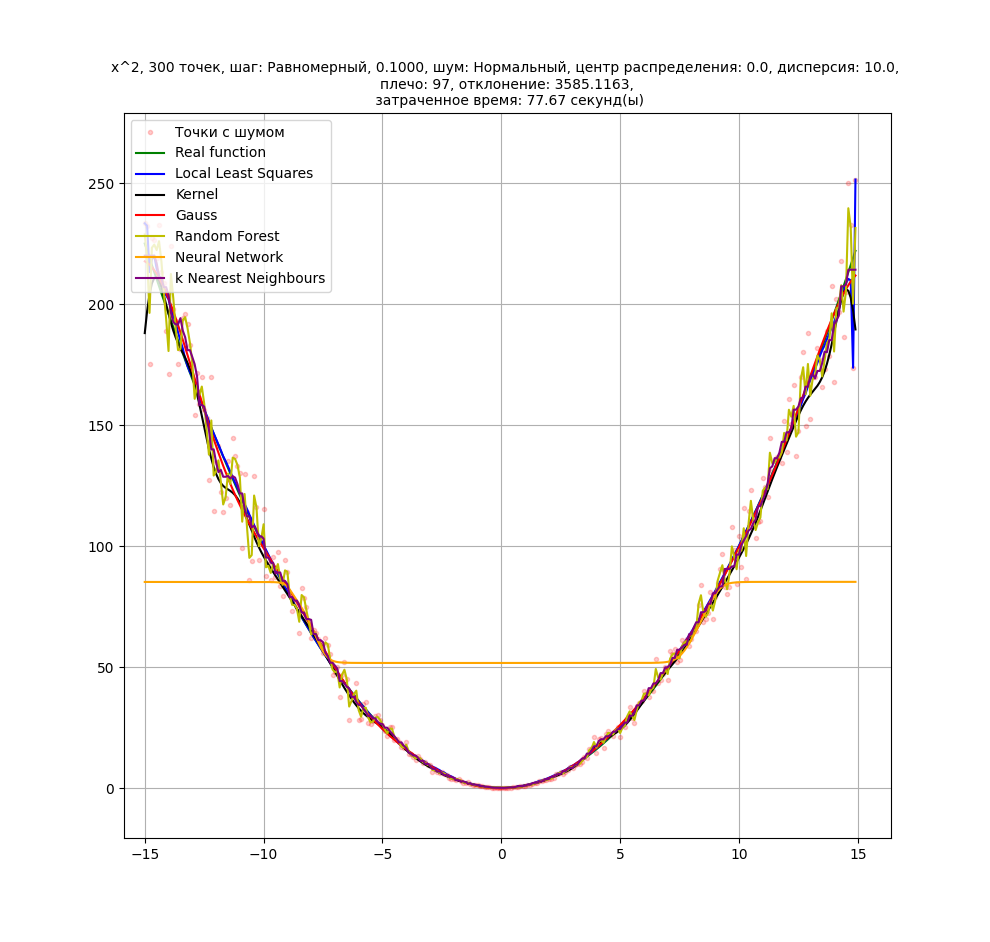


Рисунок 15 – Сглаживание параболы при нормальном шуме в 10% разными методами

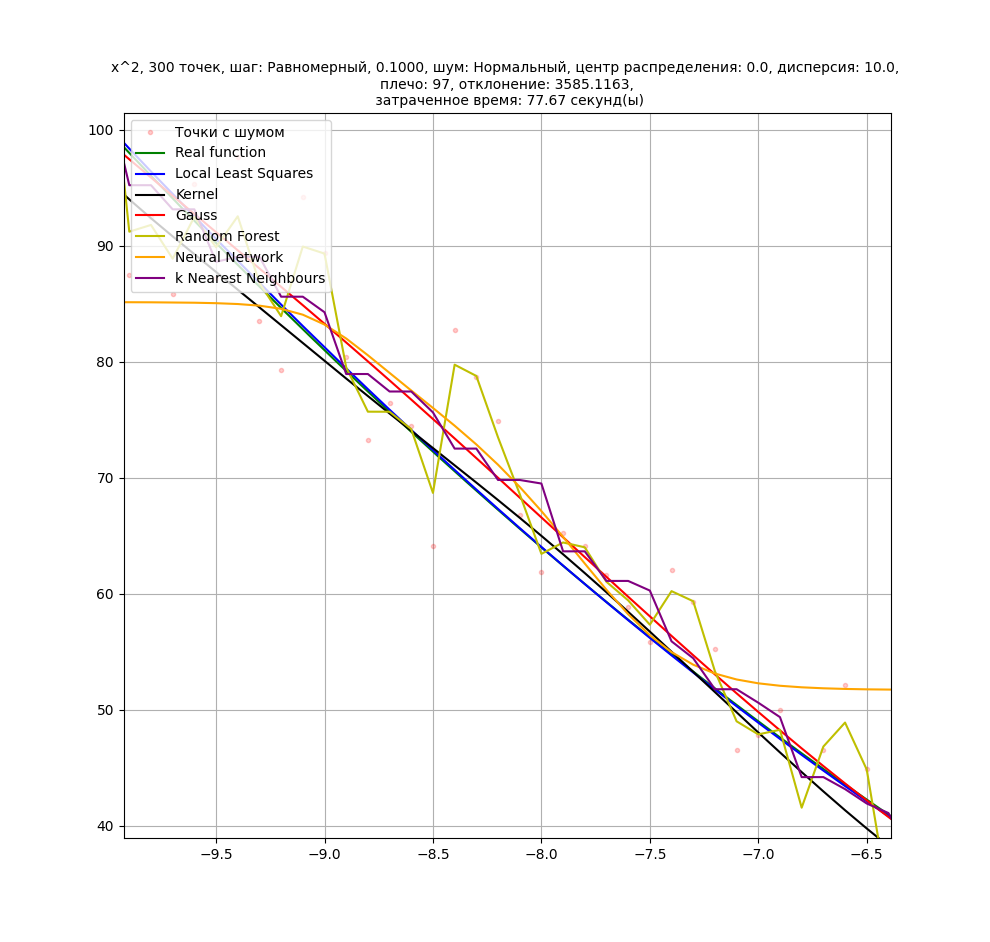


Рисунок 16 – Приближенное сглаживание параболы при нормальном шуме в 10% разными методами

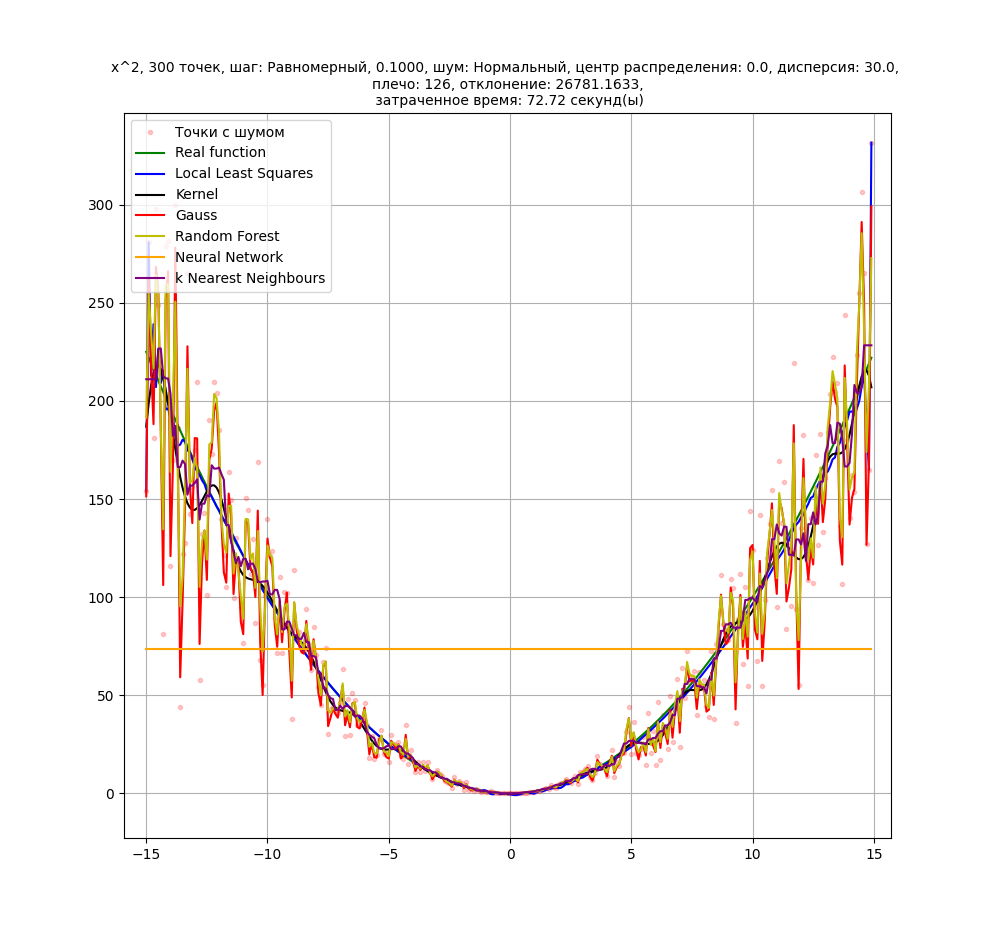


Рисунок 17 – Сглаживание параболы при нормальном шуме в 30% разными методами

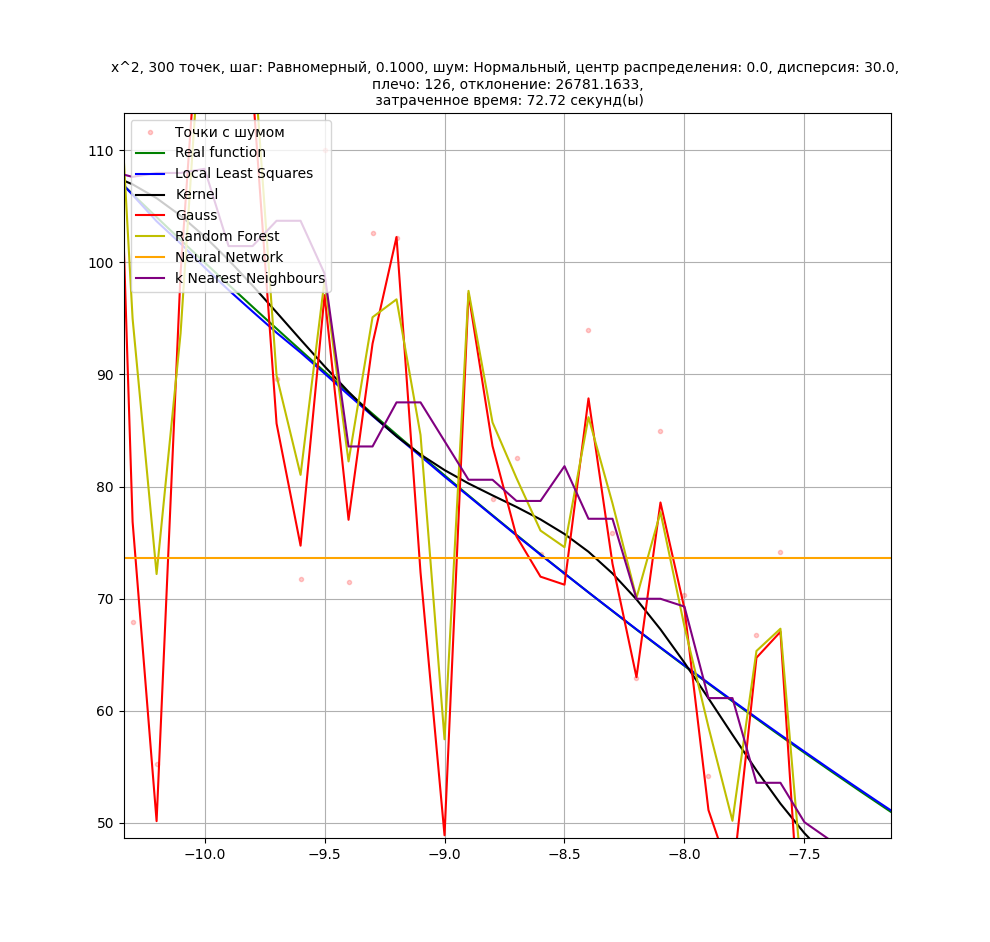


Рисунок 18 – Приближенное сглаживание параболы при нормальном шуме в 30% разными методами

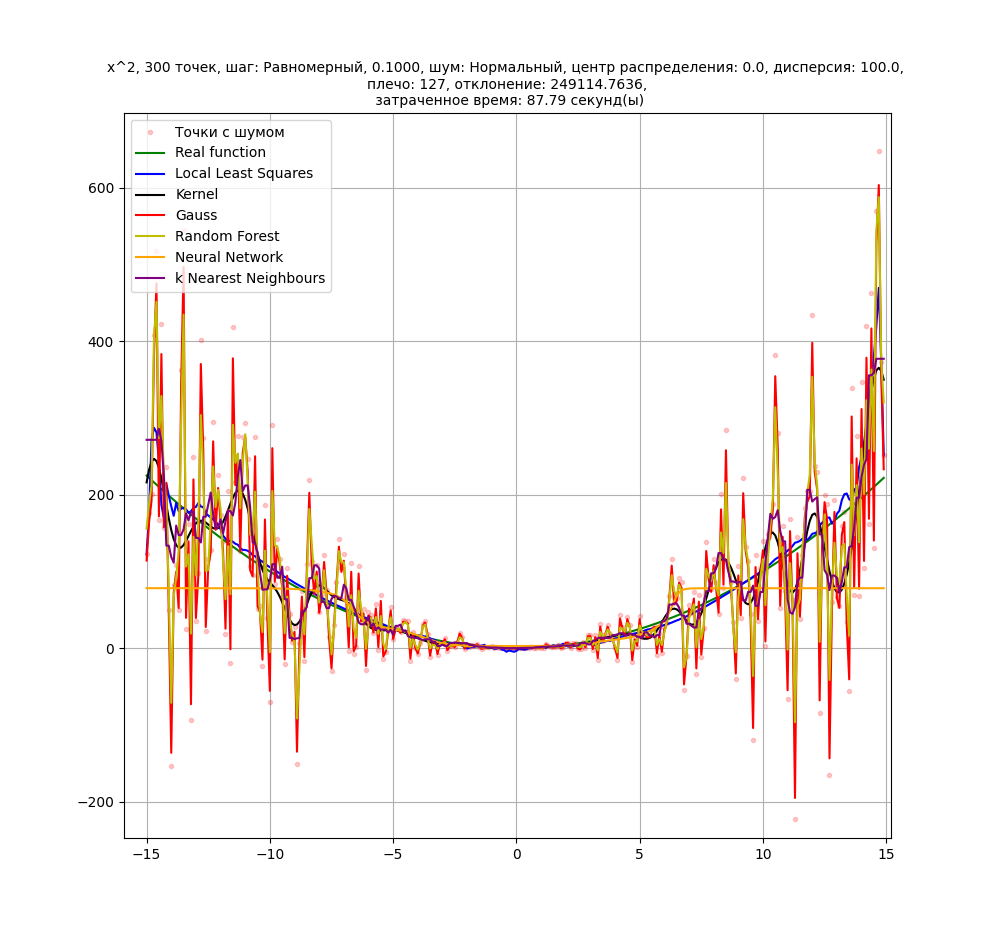


Рисунок 19 – Сглаживание параболы при нормальном шуме в 100% разными методами

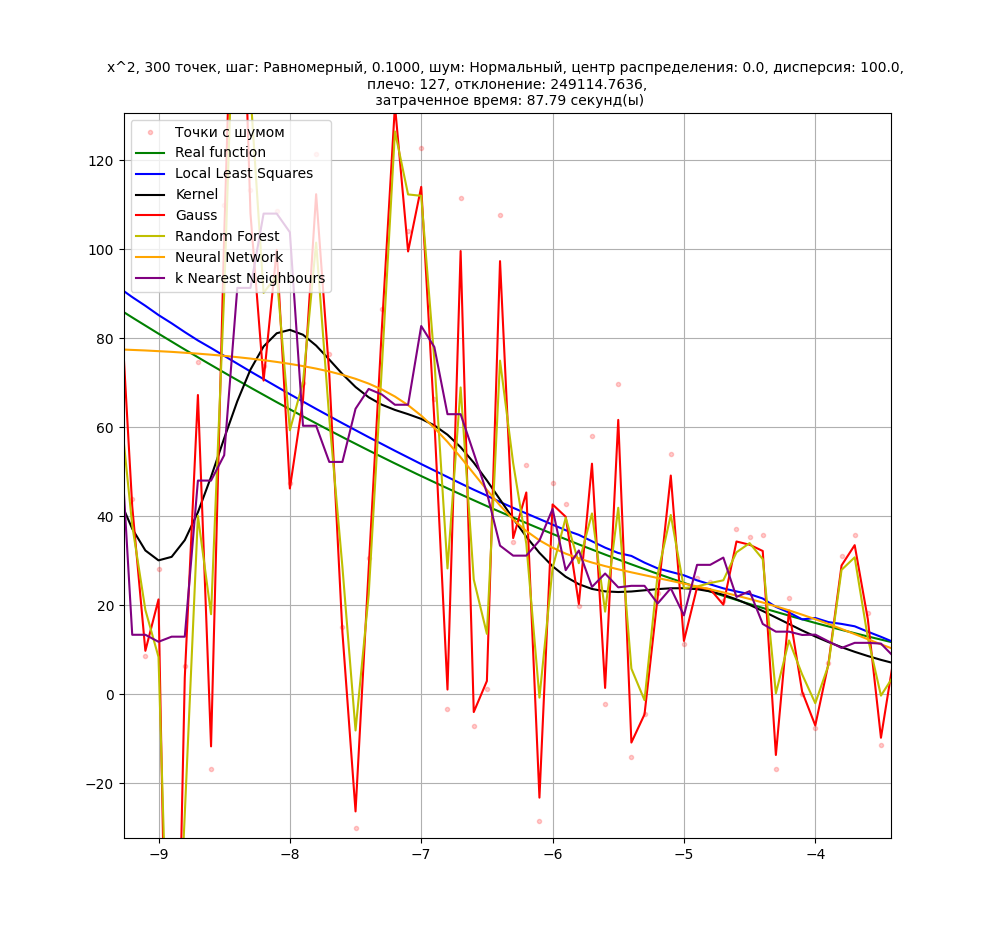


Рисунок 20 – Приближенное сглаживание параболы при нормальном шуме в 100% разными методами

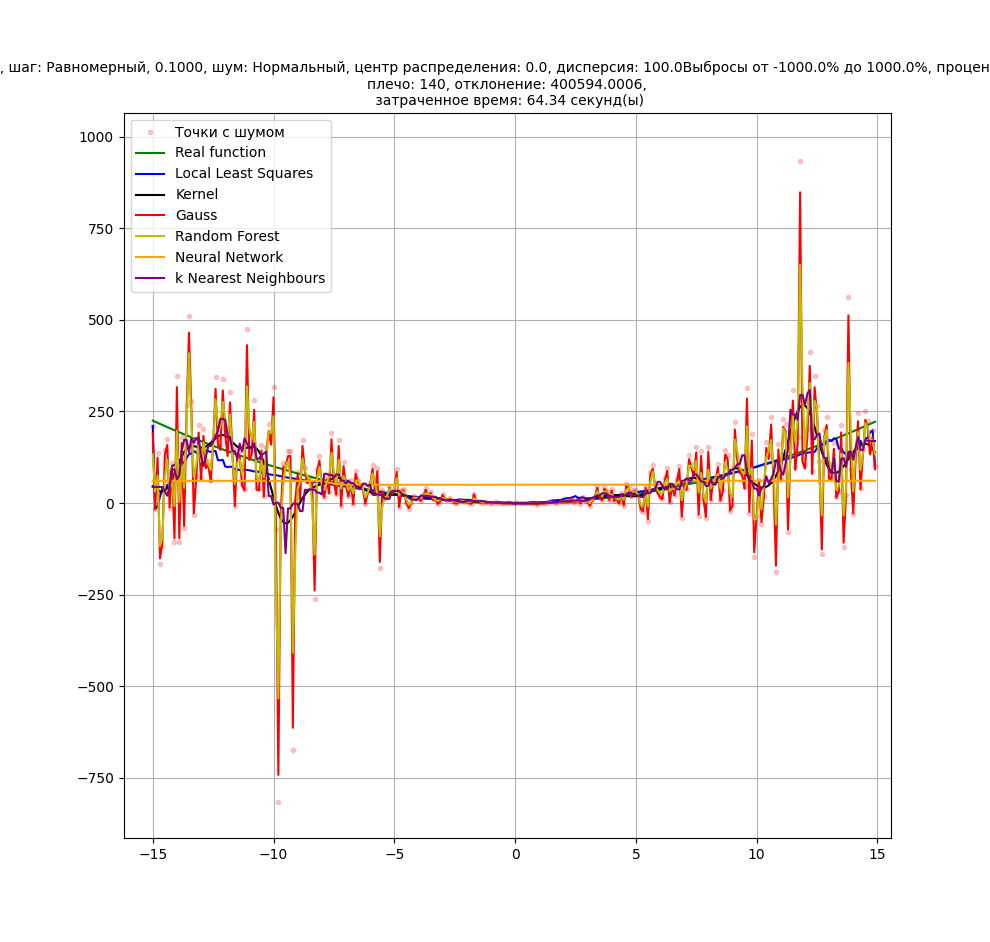


Рисунок 21 – Сглаживание параболы при нормальном шуме в 100% и выбросах в 1000% разными методами



Рисунок 22 – Приближенное сглаживание параболы при нормальном шуме в 100% и выбросах в 1000% разными методами

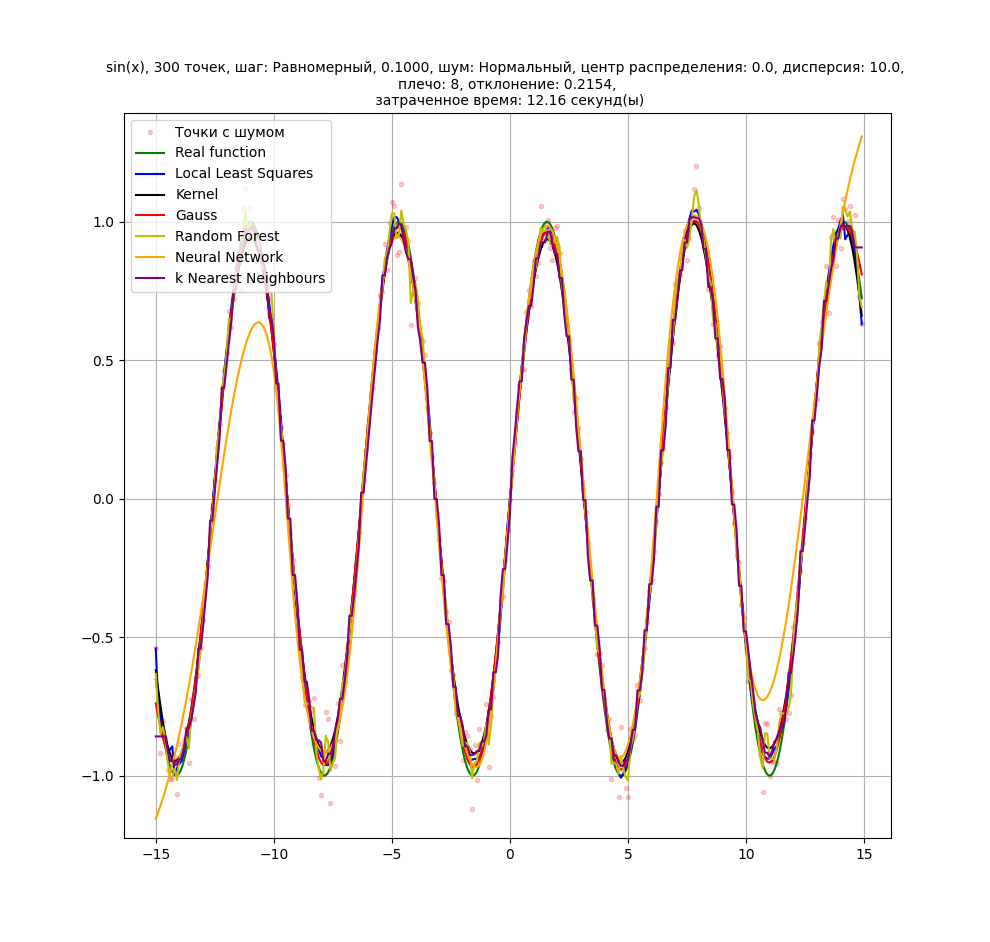


Рисунок 23 – Сглаживание синуса при нормальном шуме в 10% разными методами

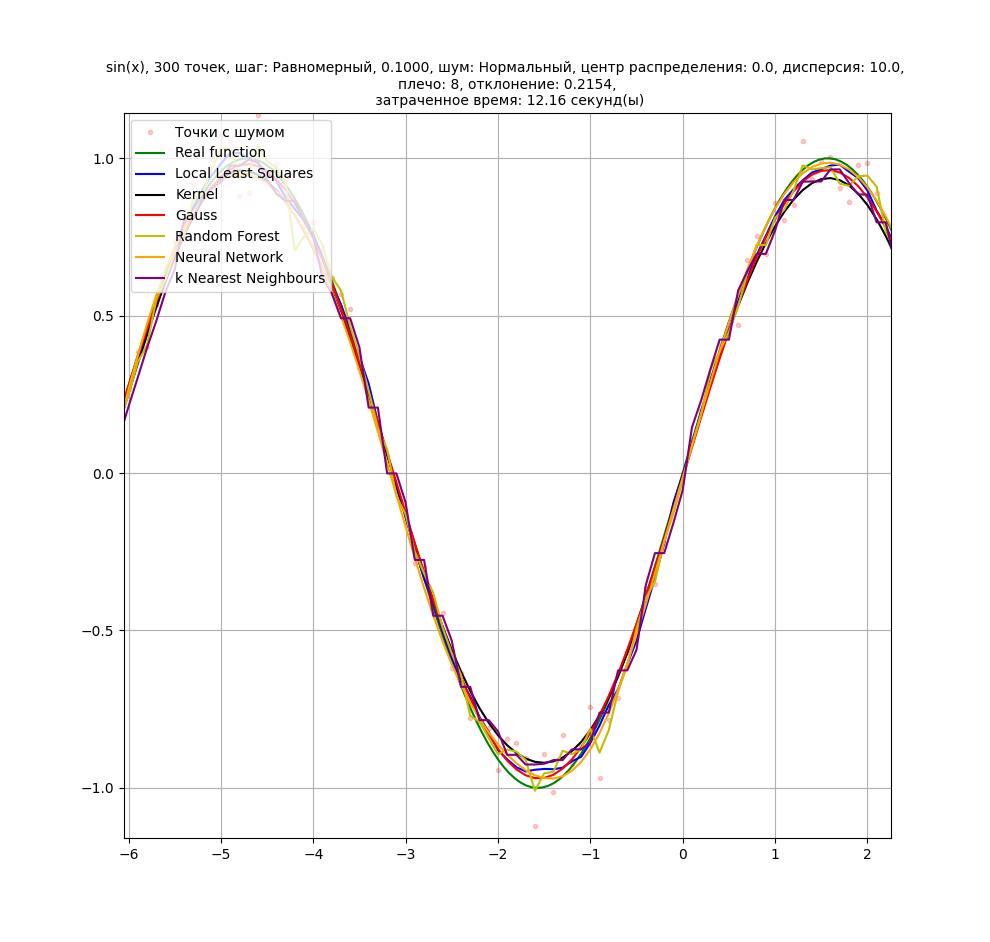


Рисунок 24 – Приближенное сглаживание синуса при нормальном шуме в 10% разными методами

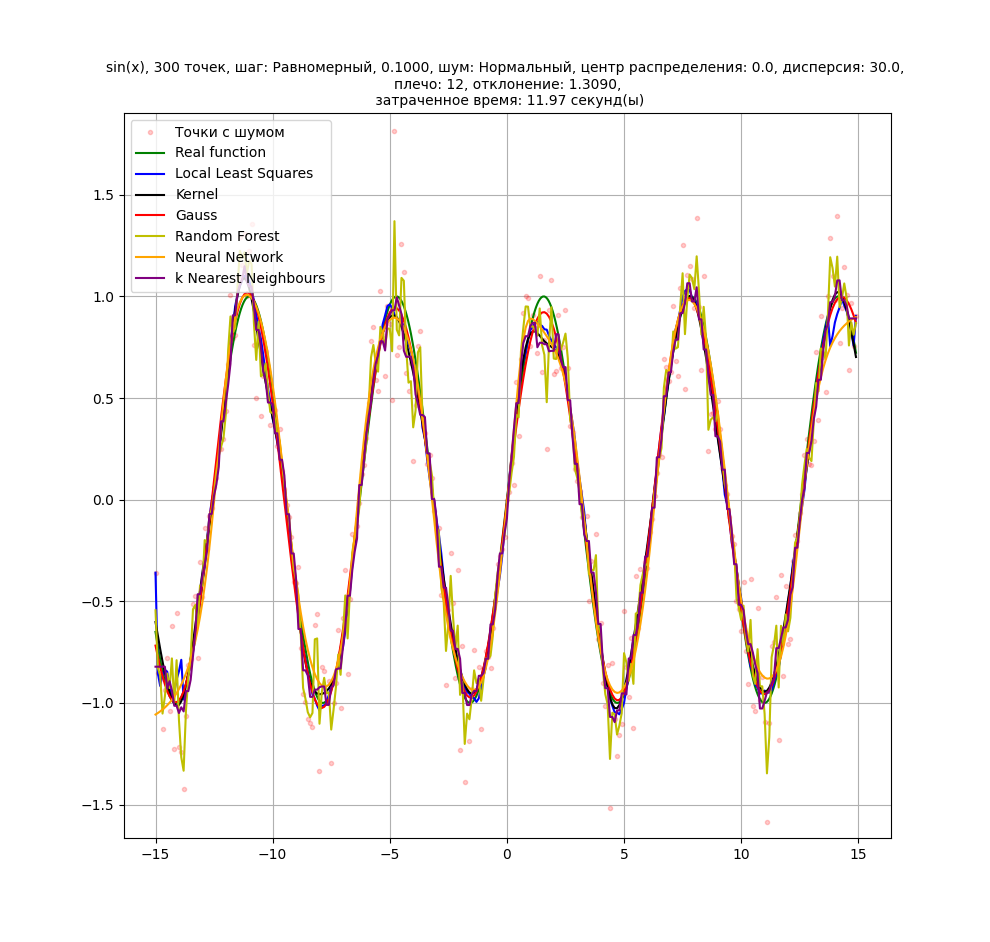


Рисунок 25 – Сглаживание синуса при нормальном шуме в 30% разными методами

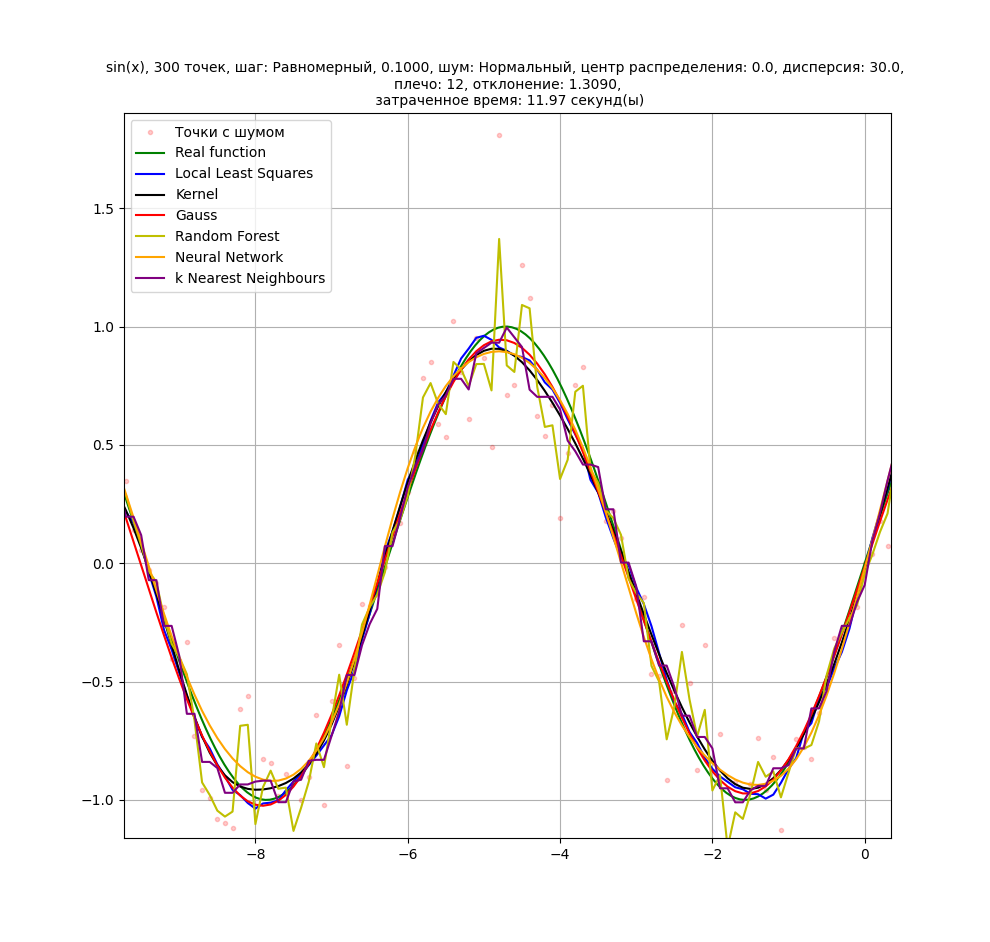


Рисунок 26 - Приближенное сглаживание синуса при нормальном шуме в 30% разными методами

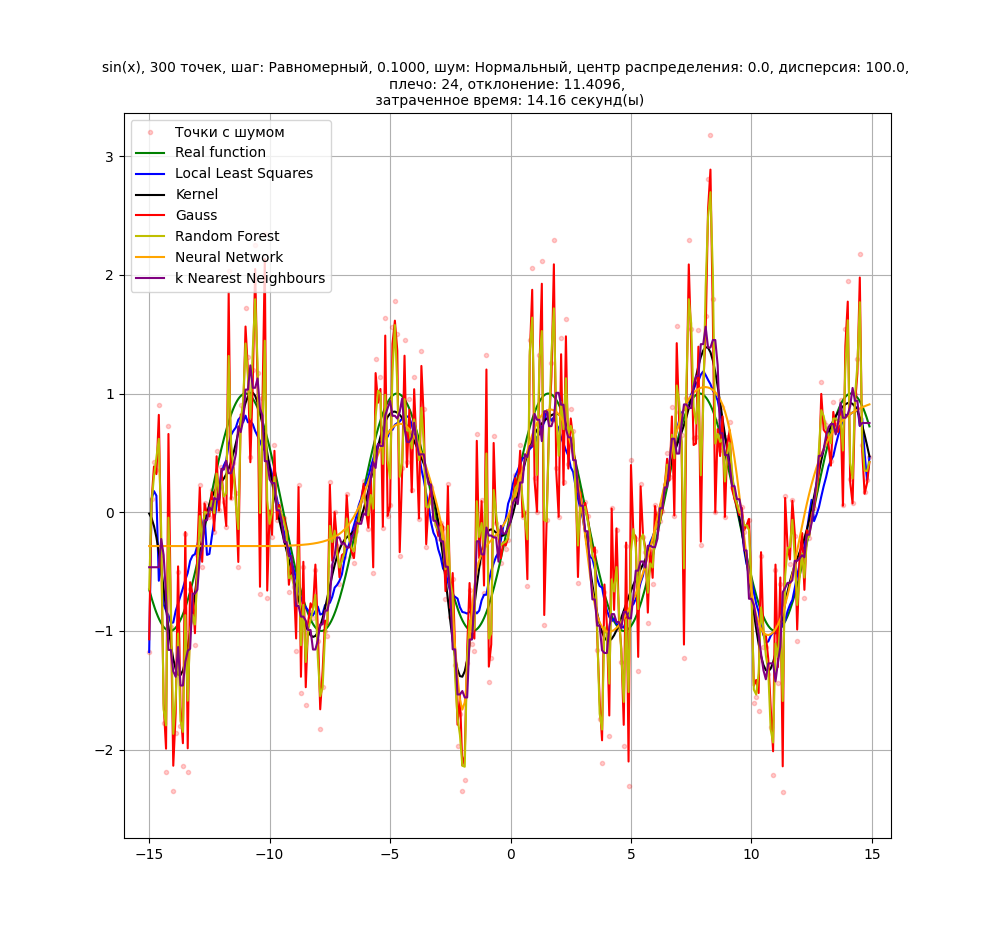


Рисунок 27 – Сглаживание синуса при нормальном шуме в 100% разными методами

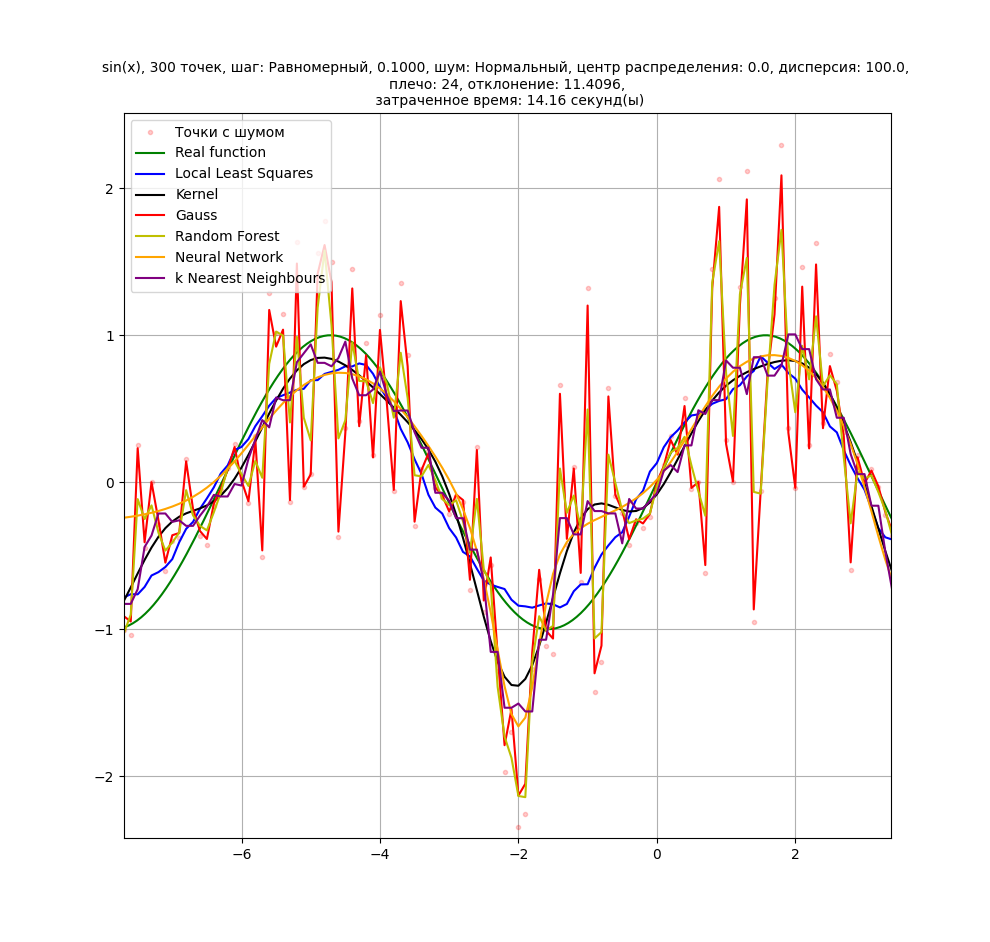


Рисунок 28 - Приближенное сглаживание синуса при нормальном шуме в 100% разными методами

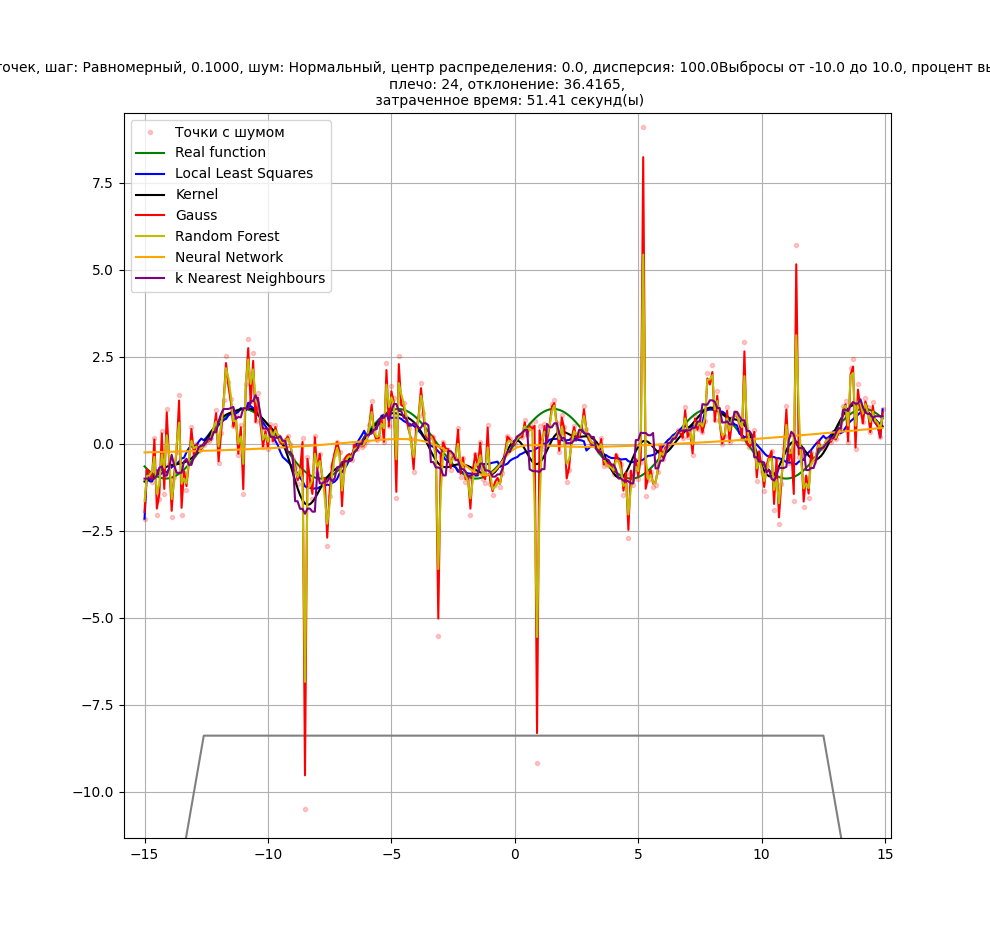
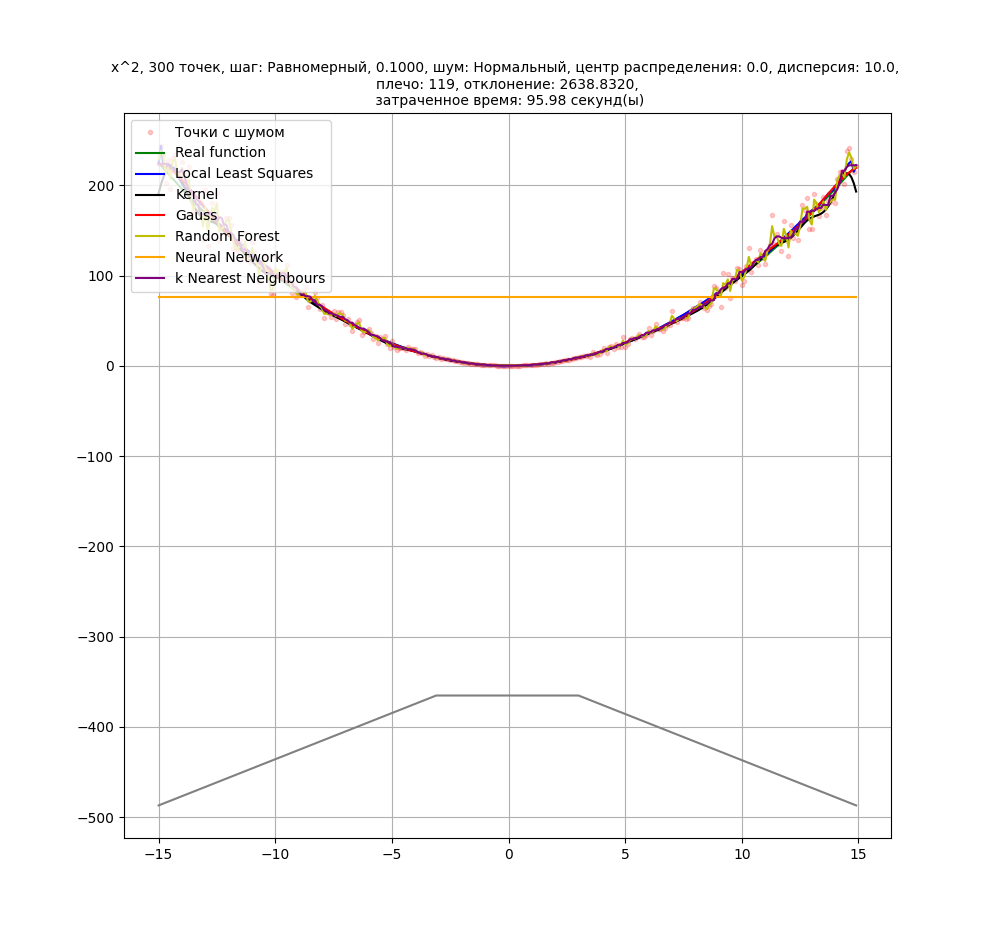
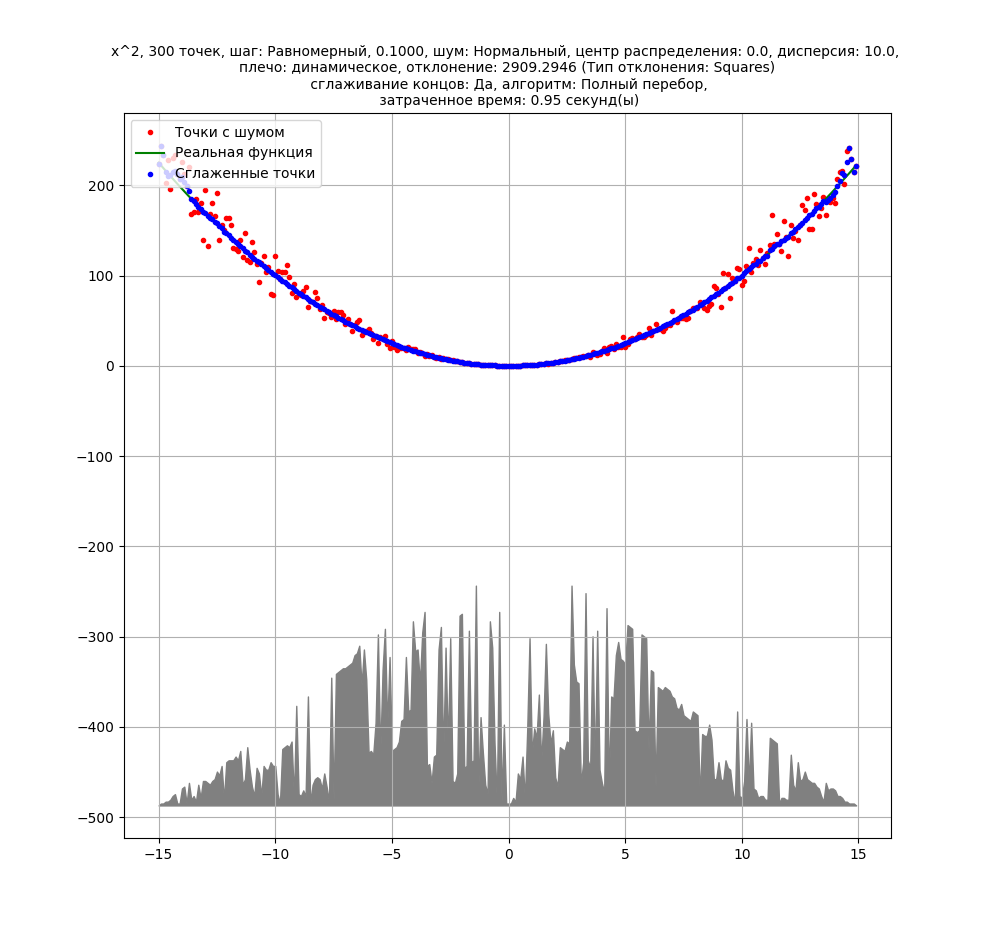
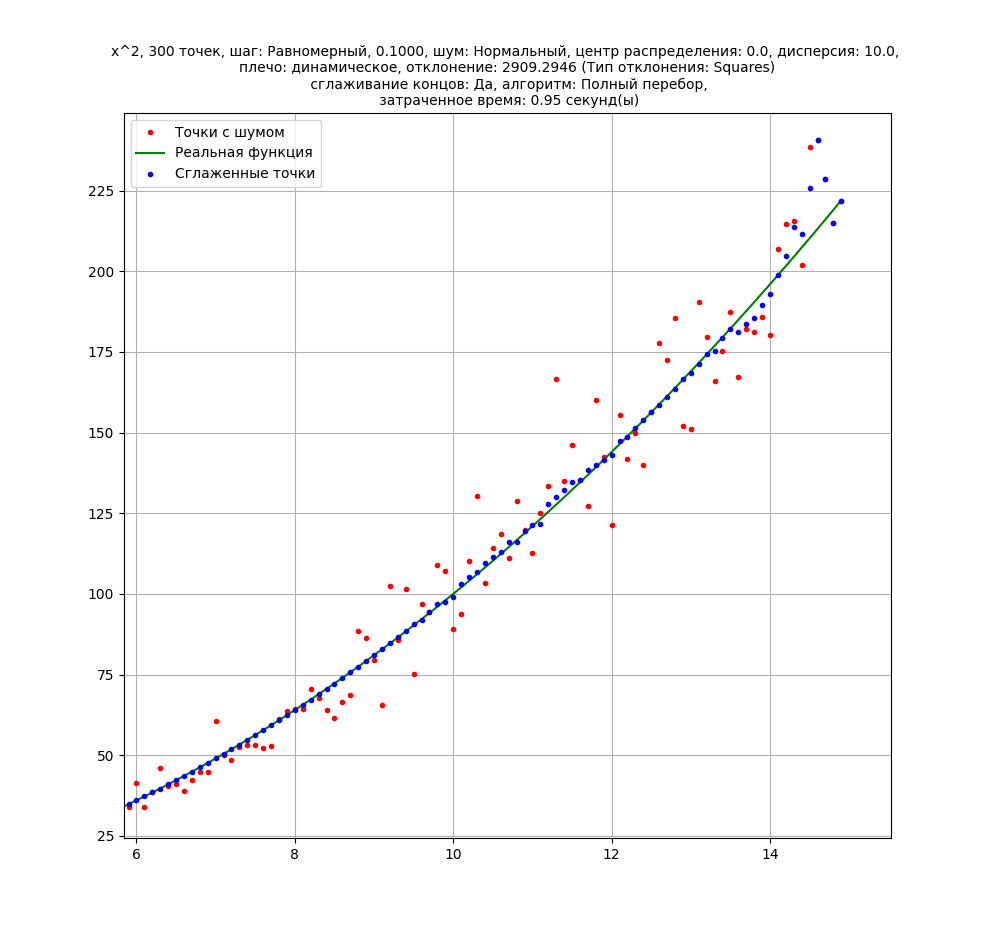
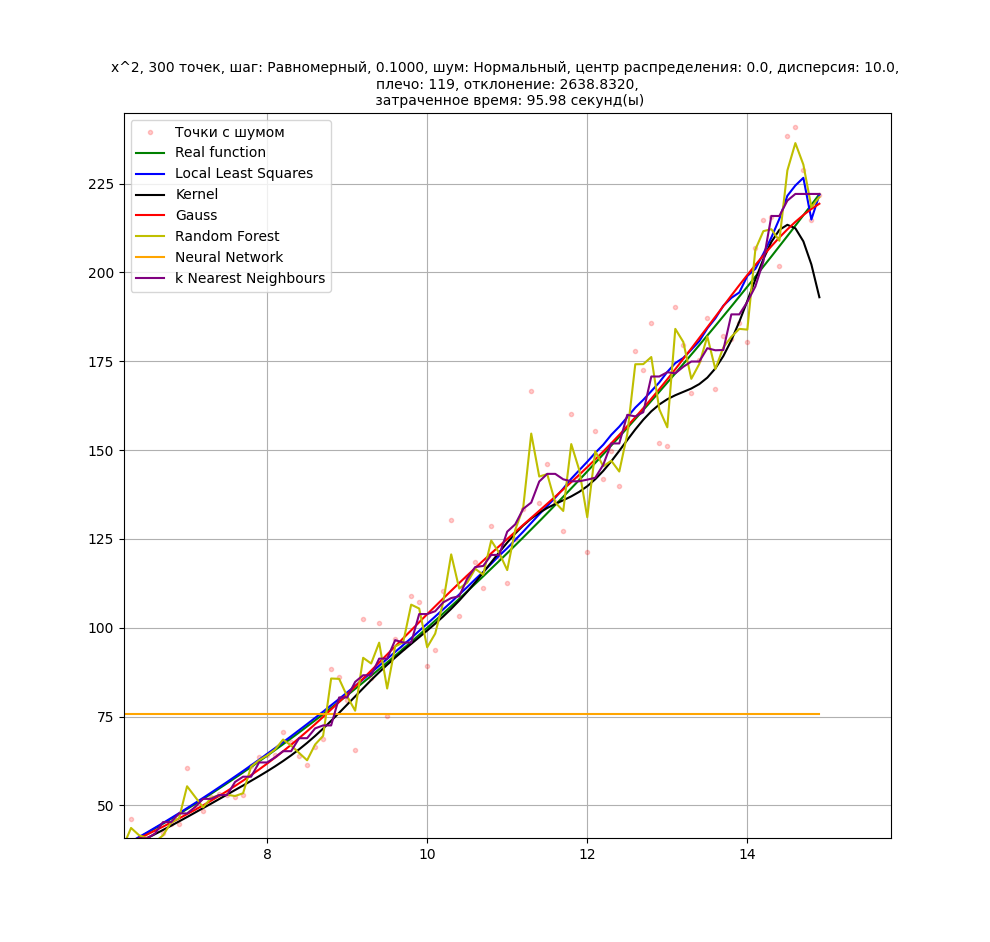
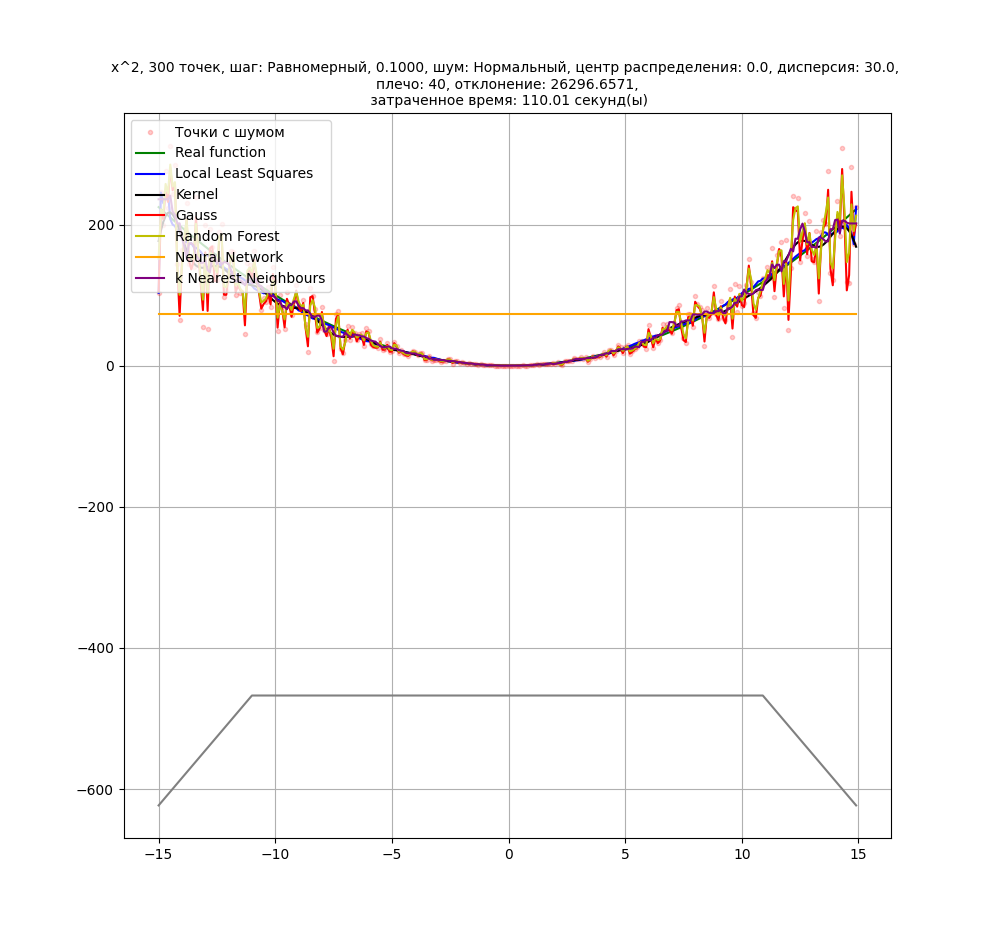
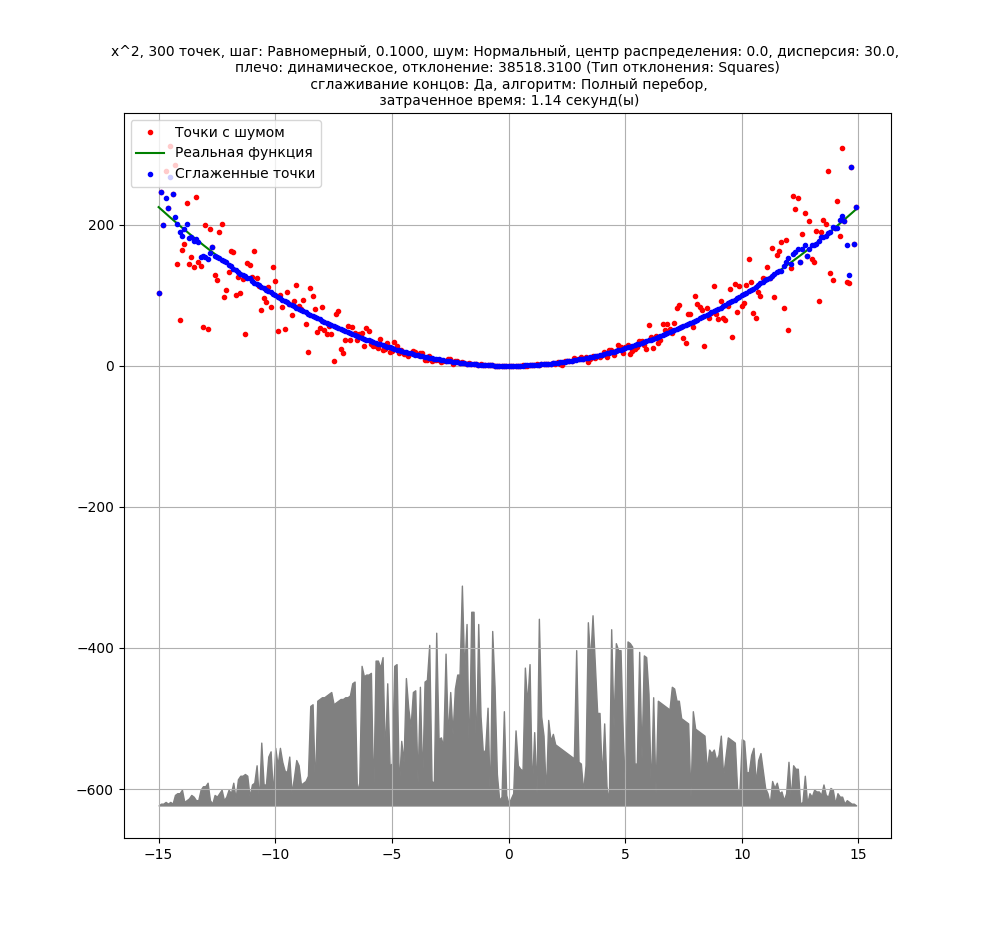
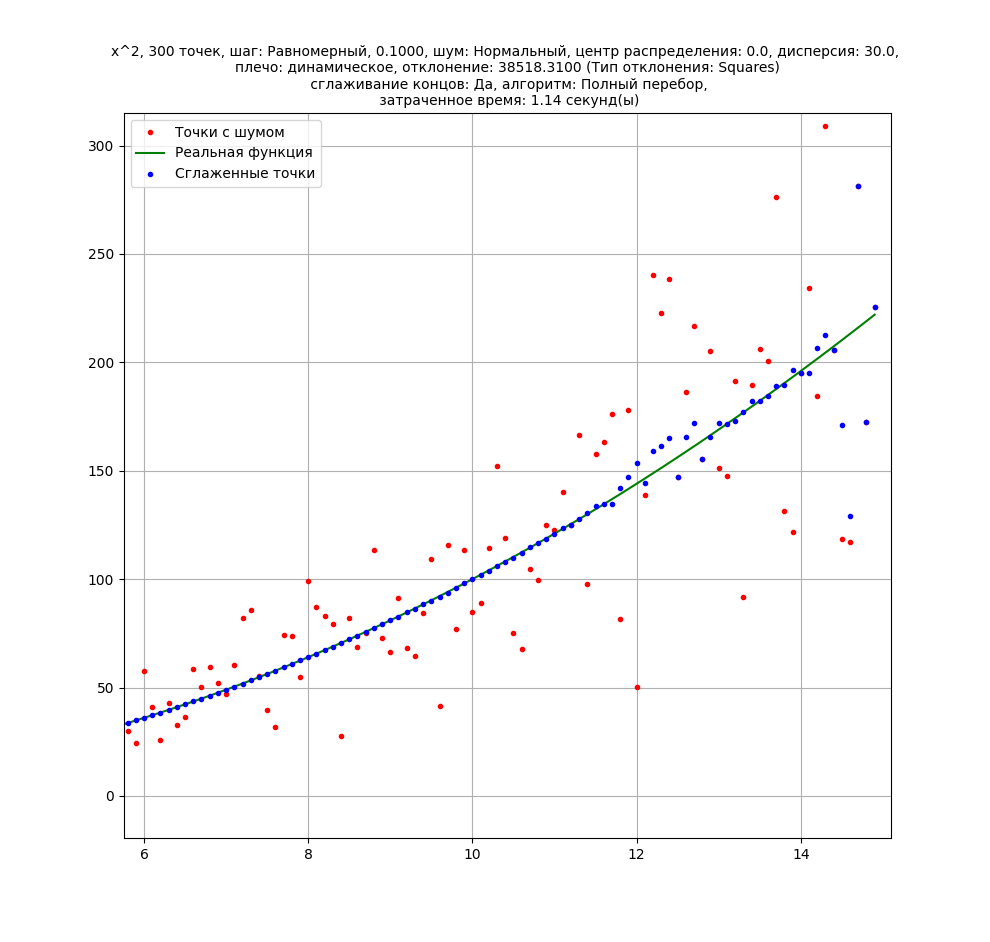
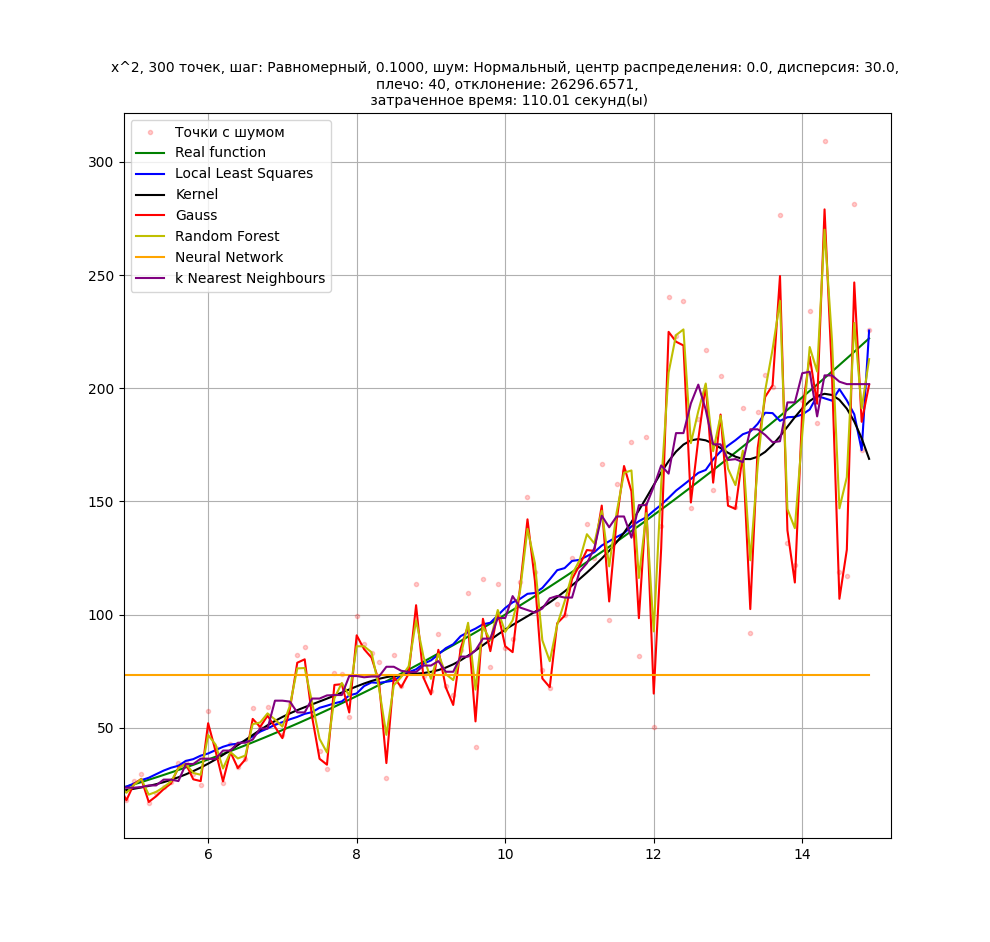
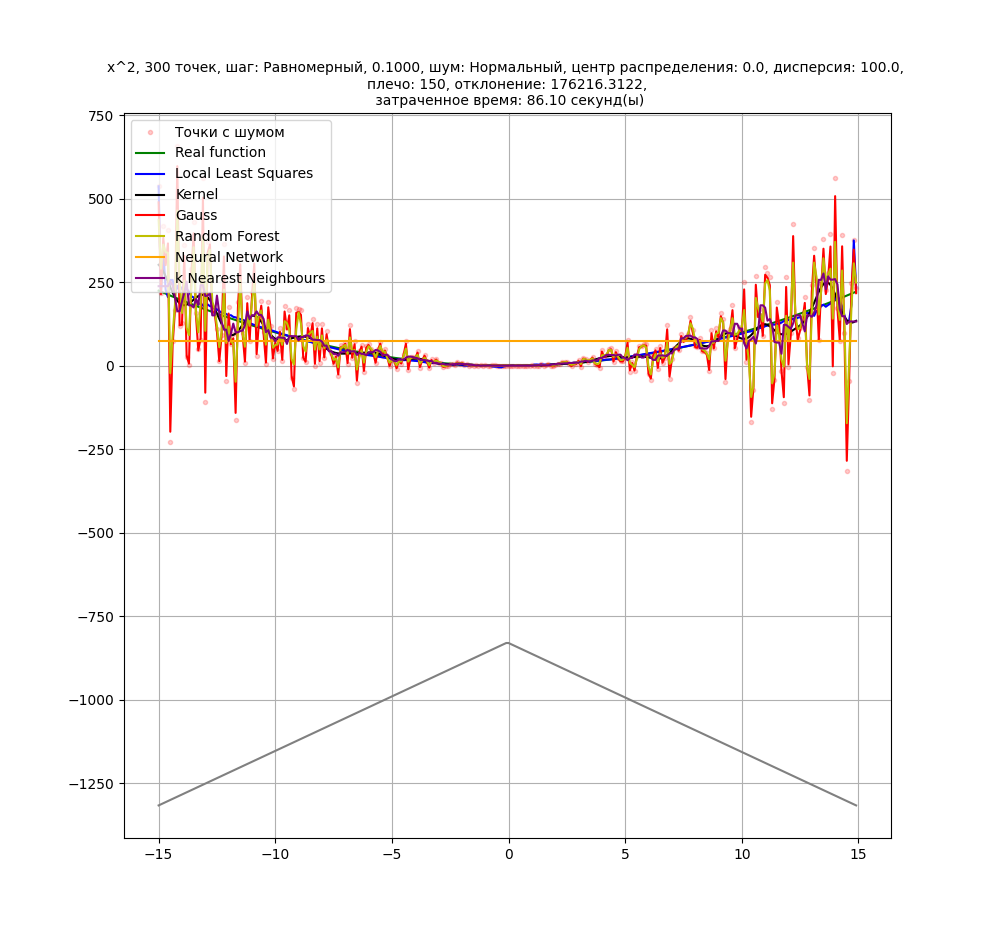
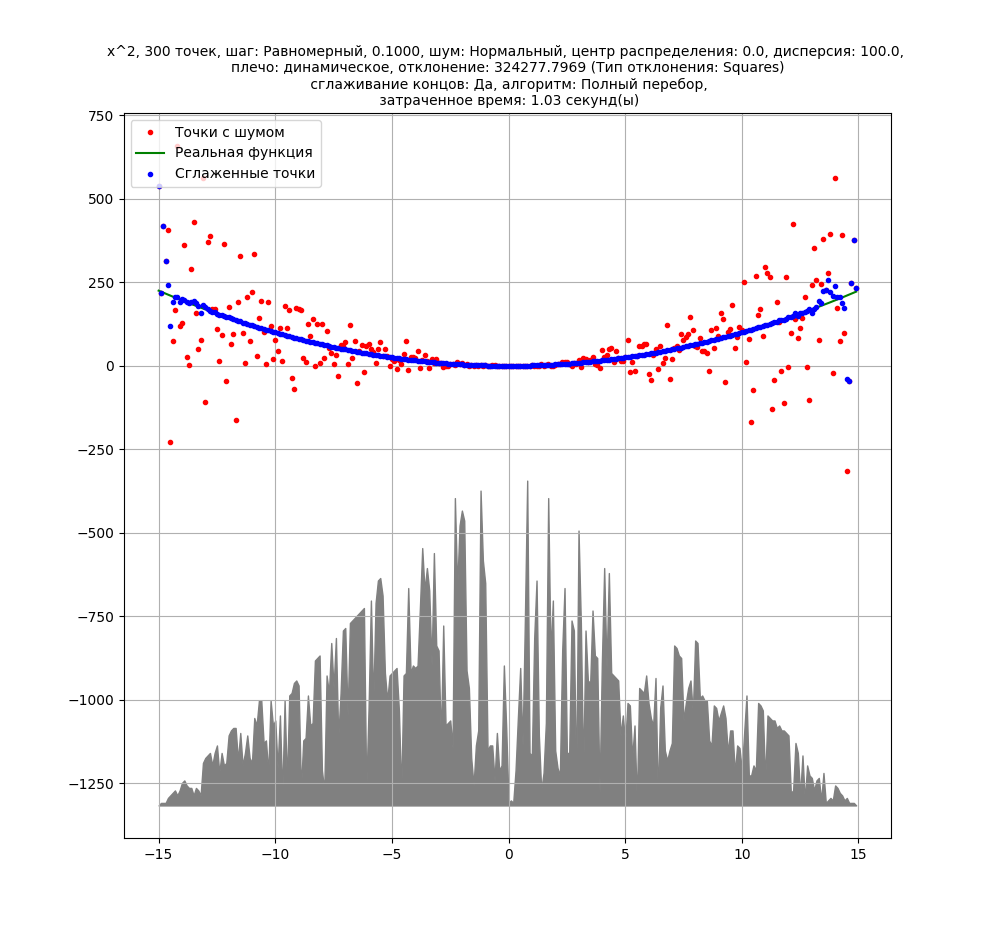
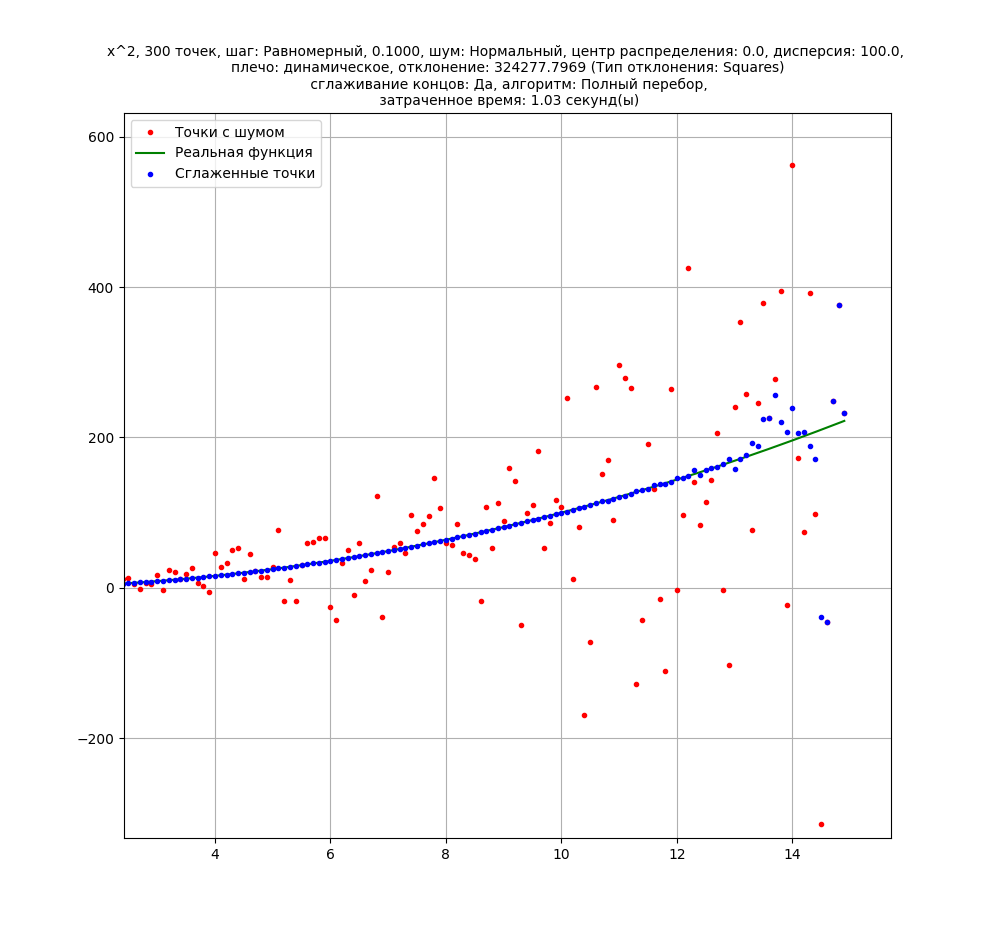
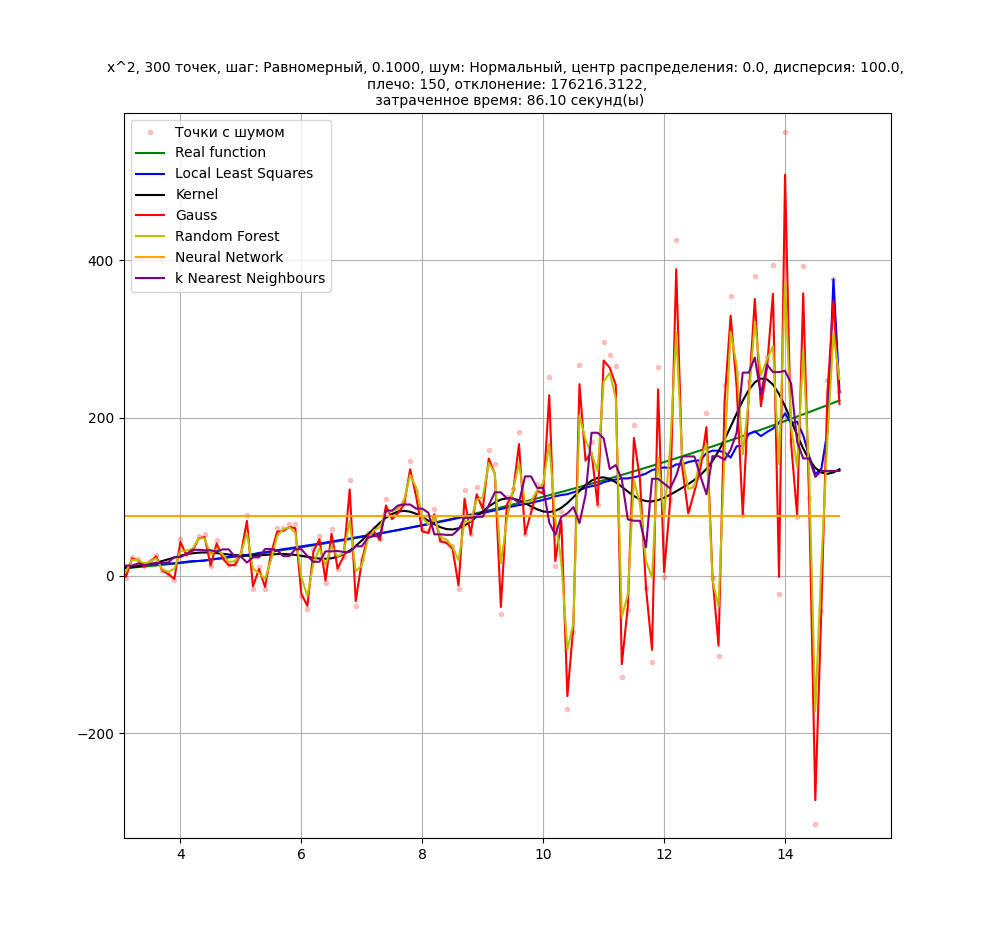
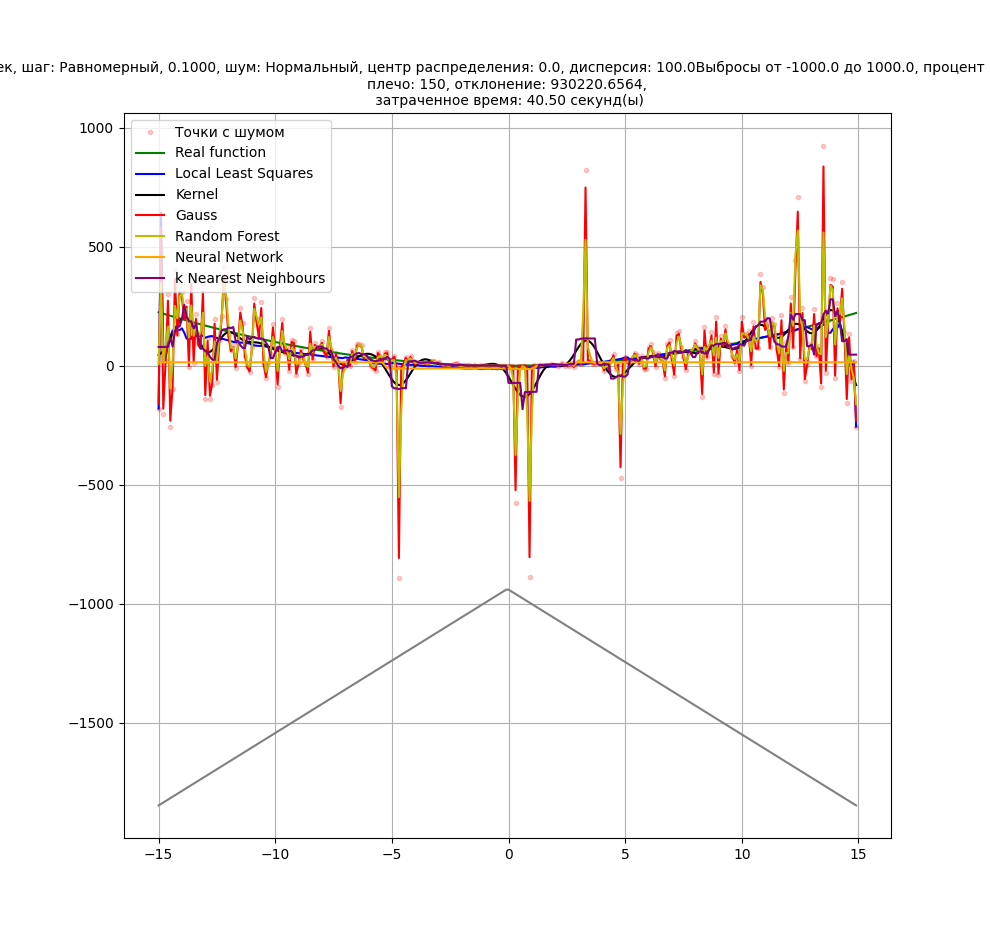
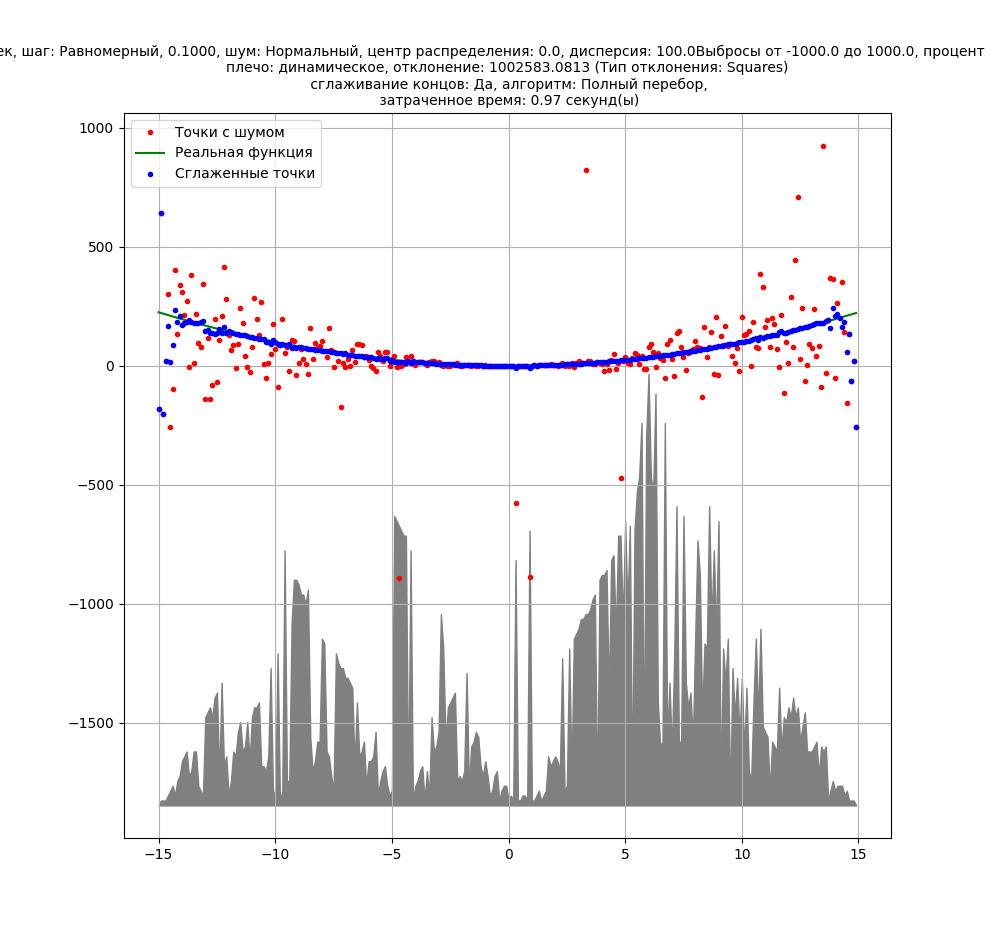
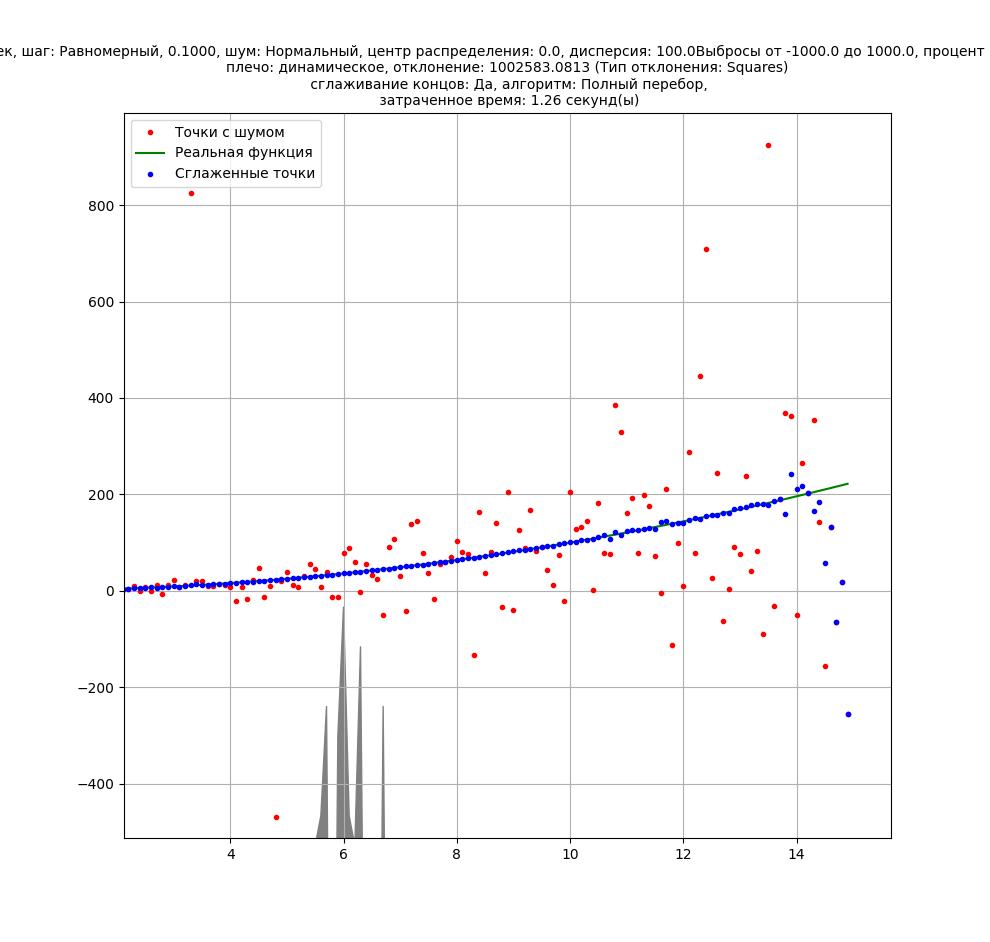
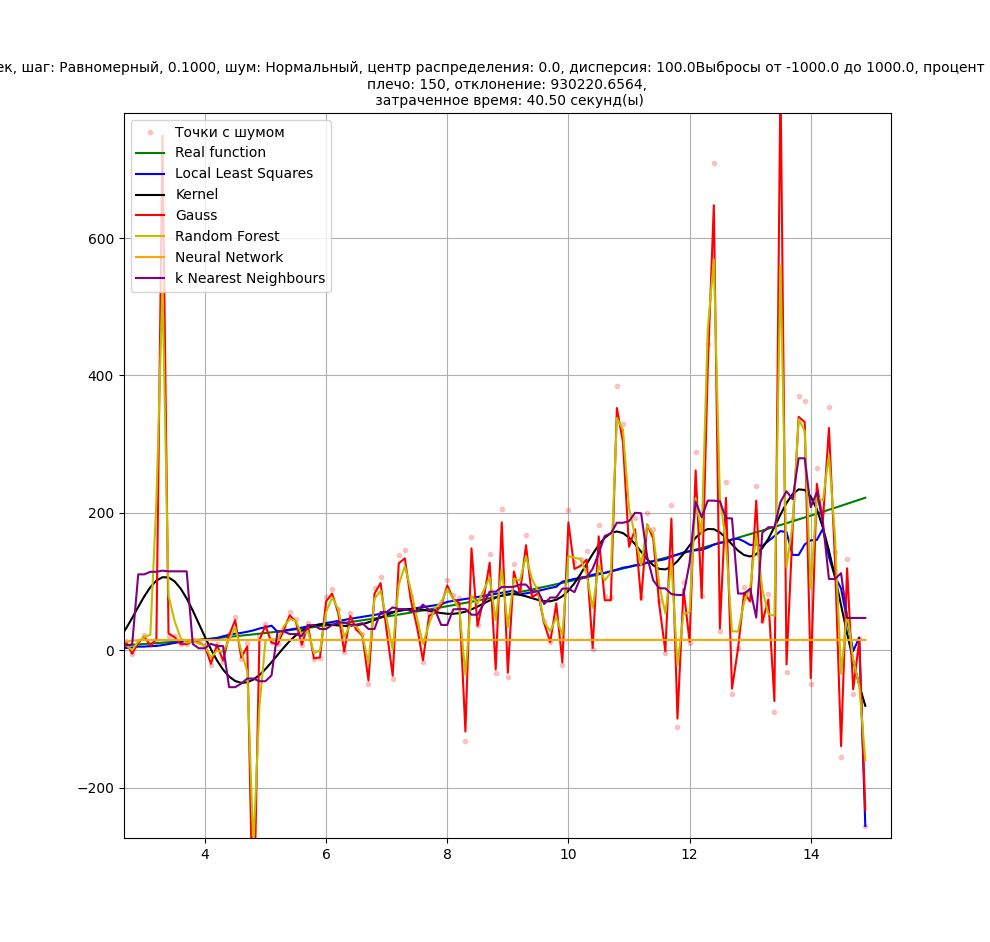
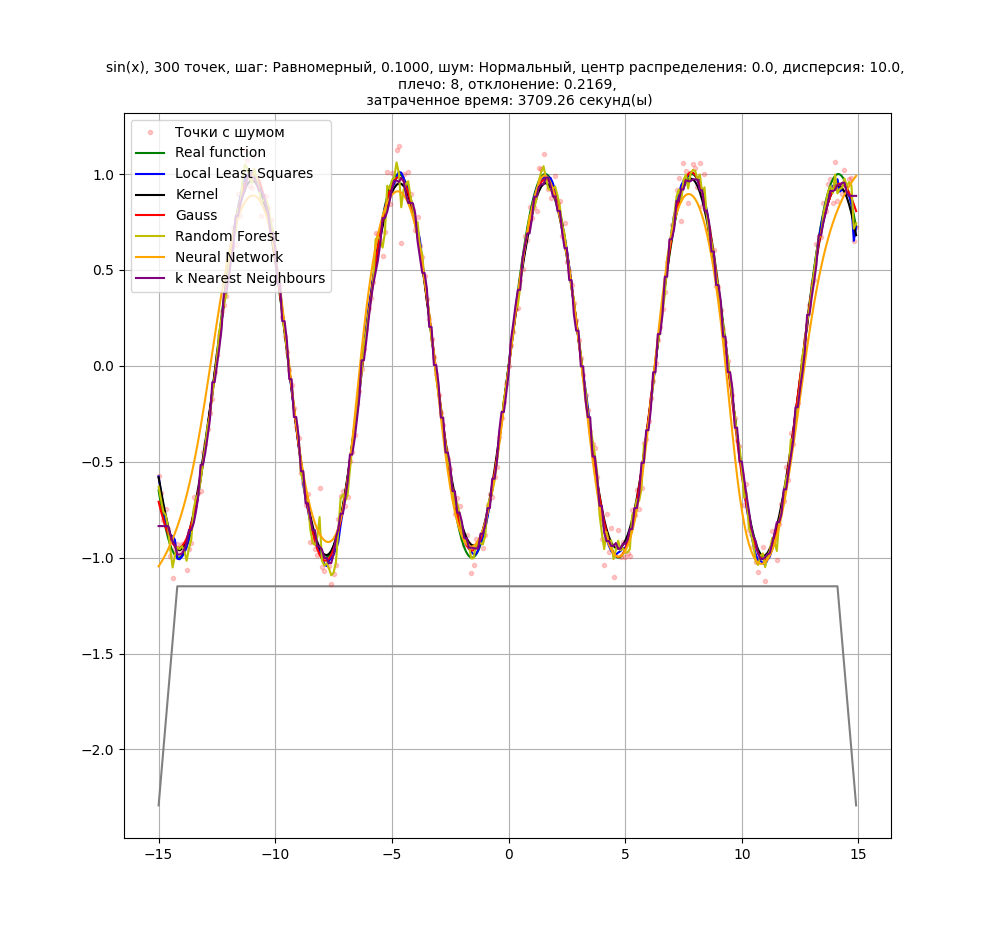
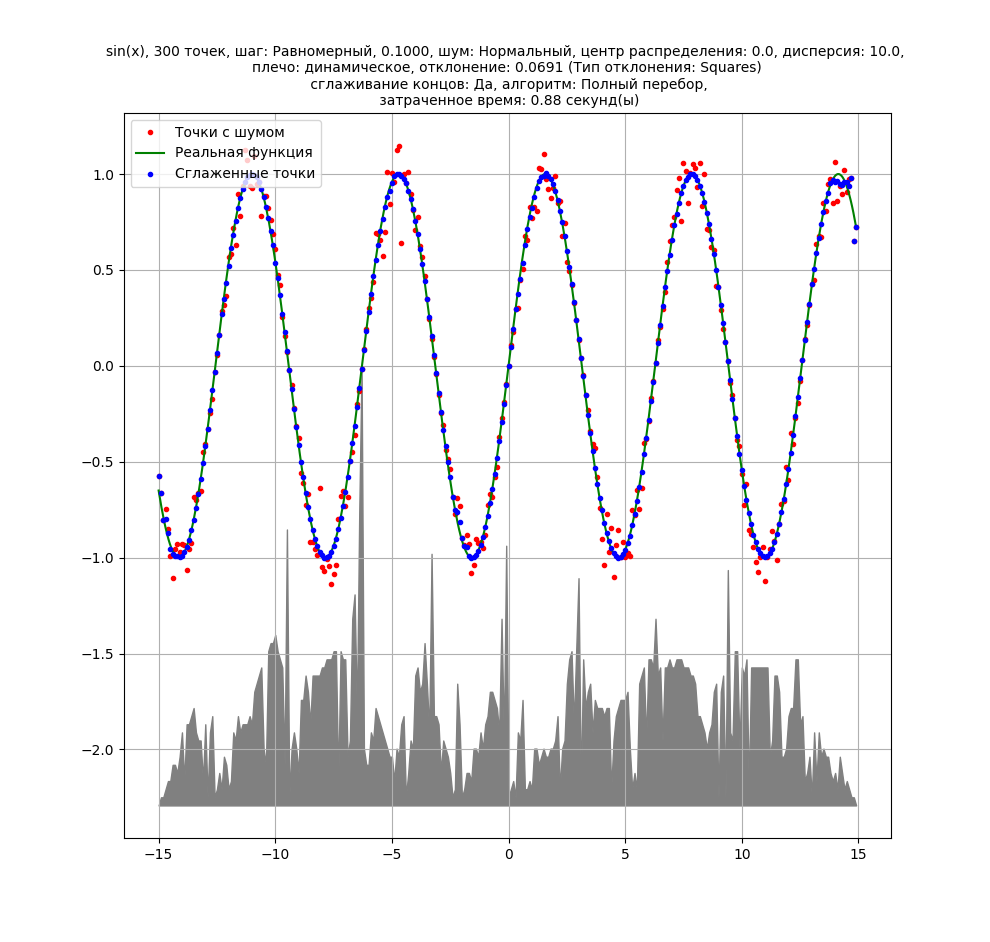
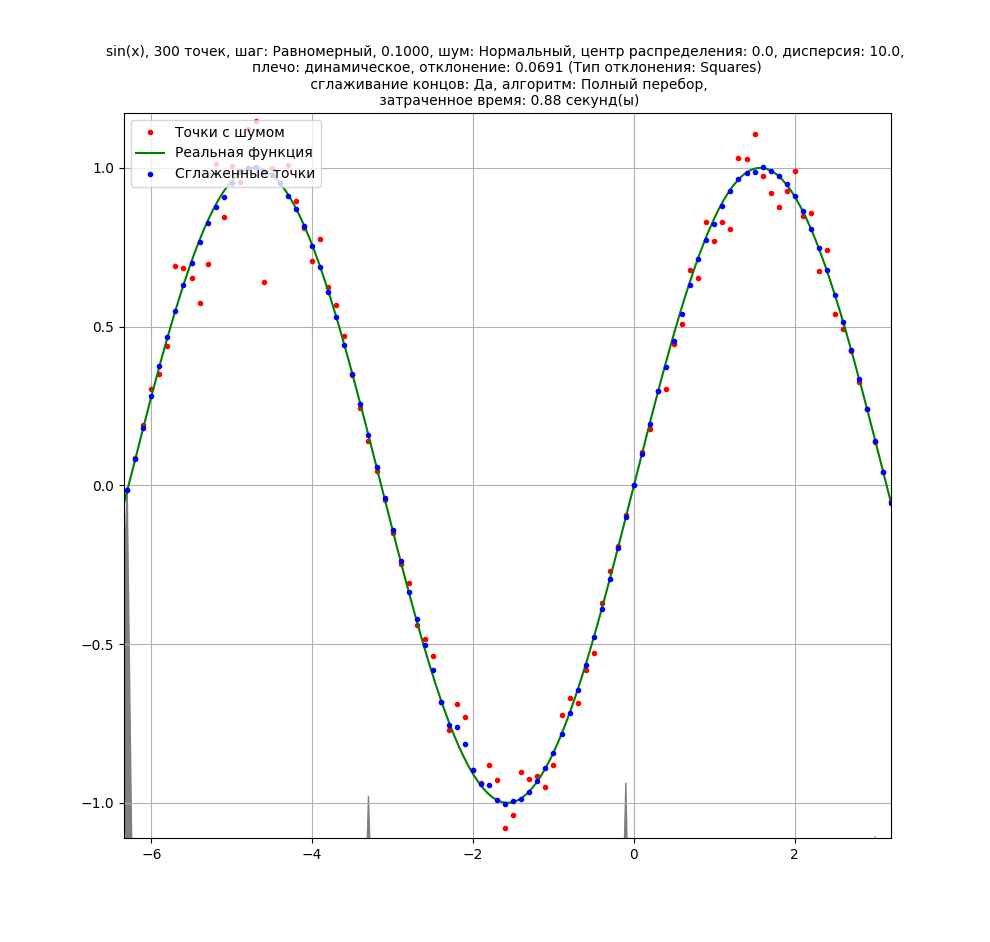
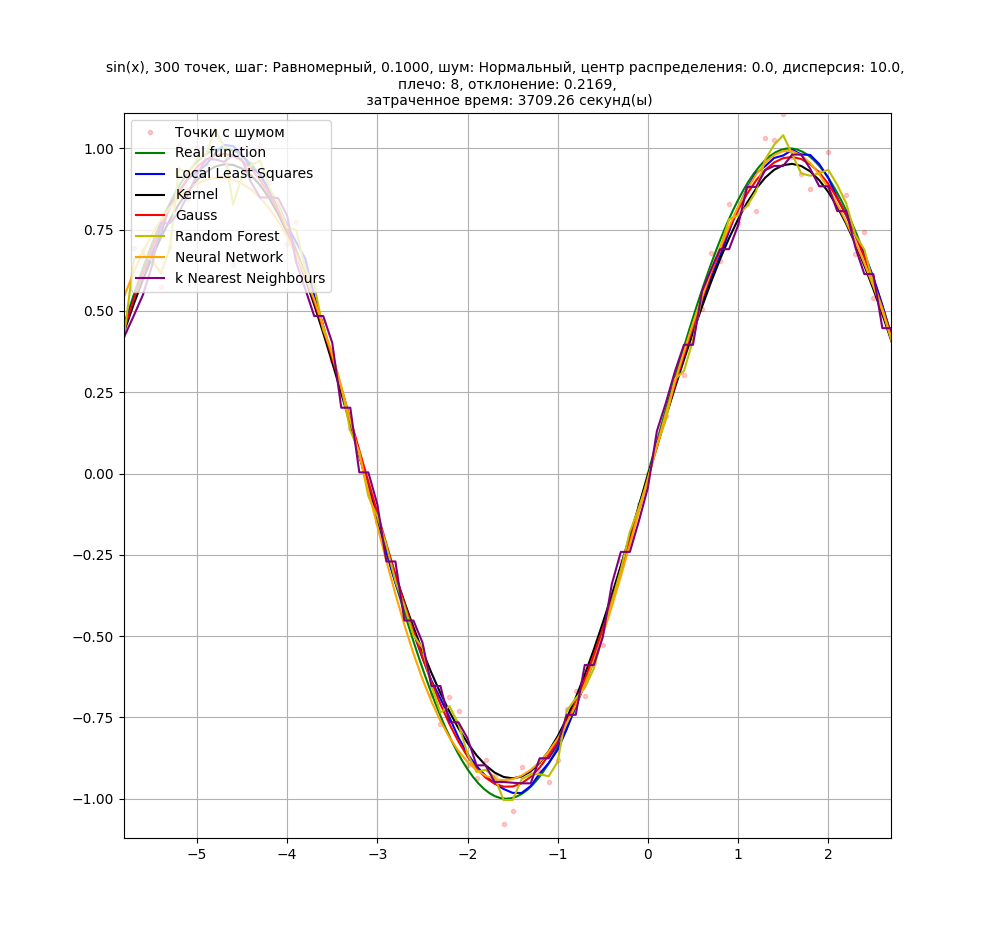
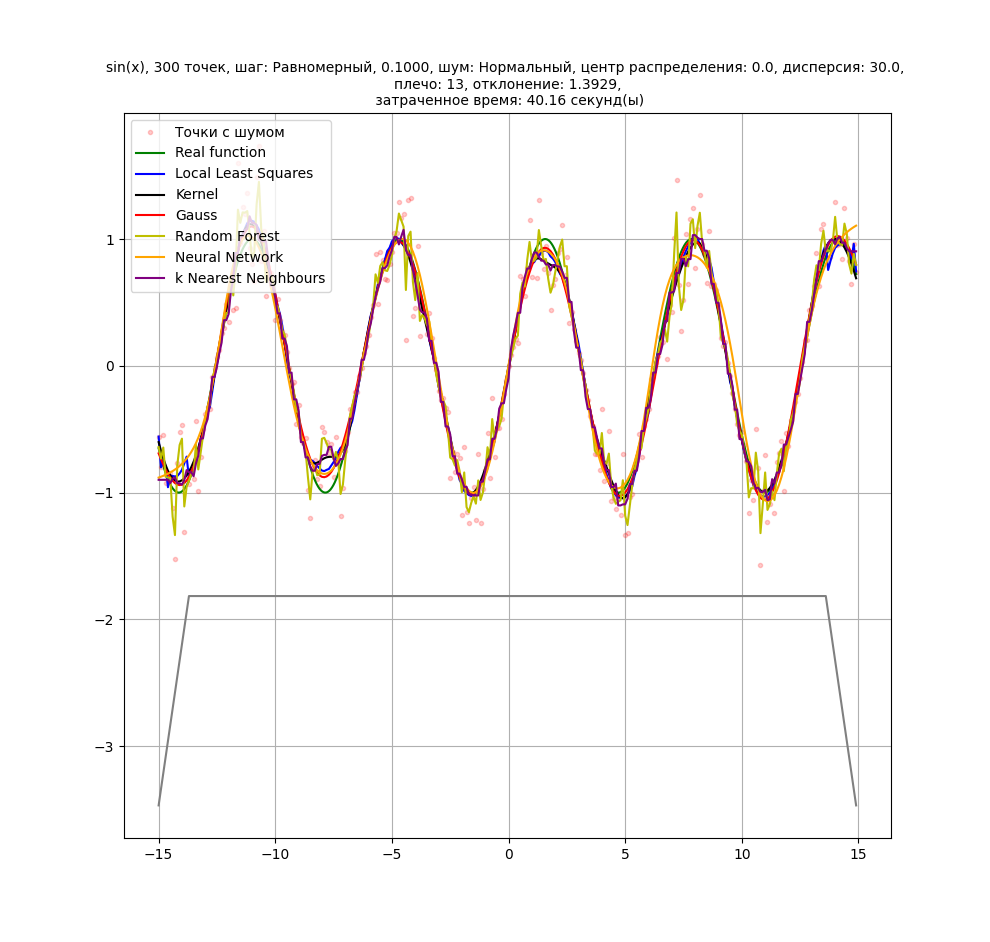
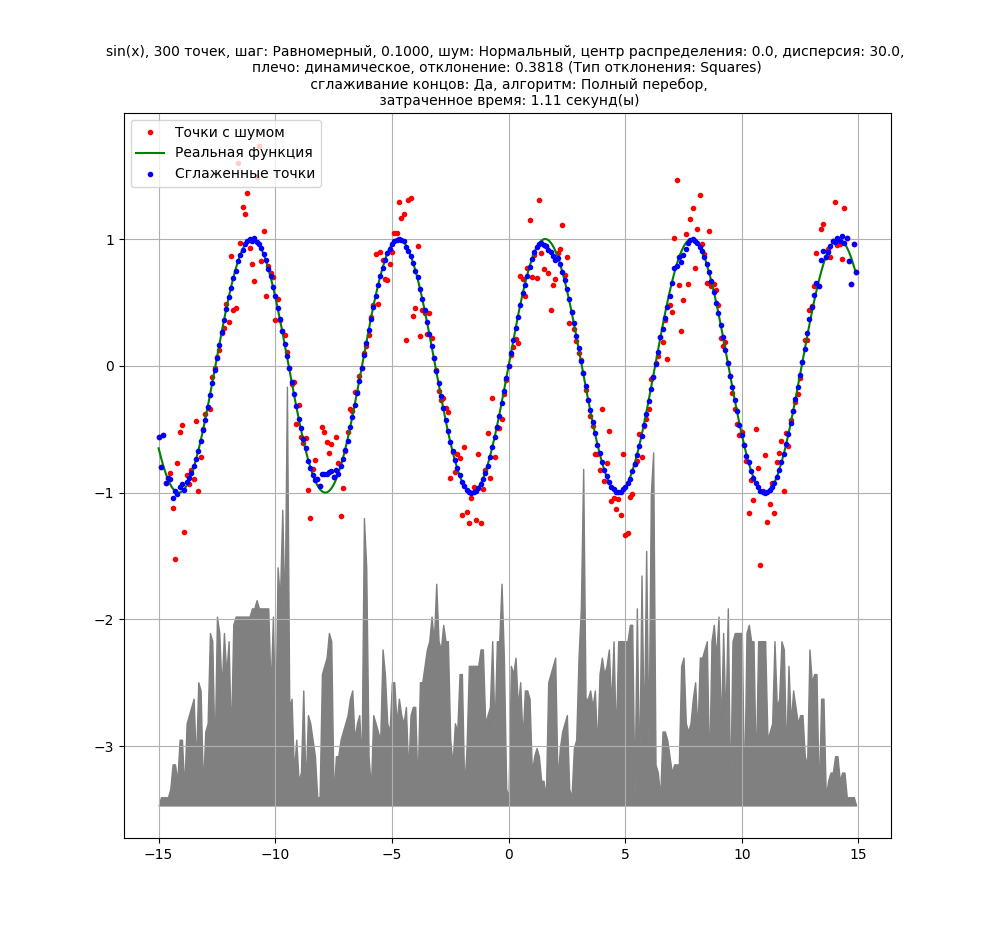
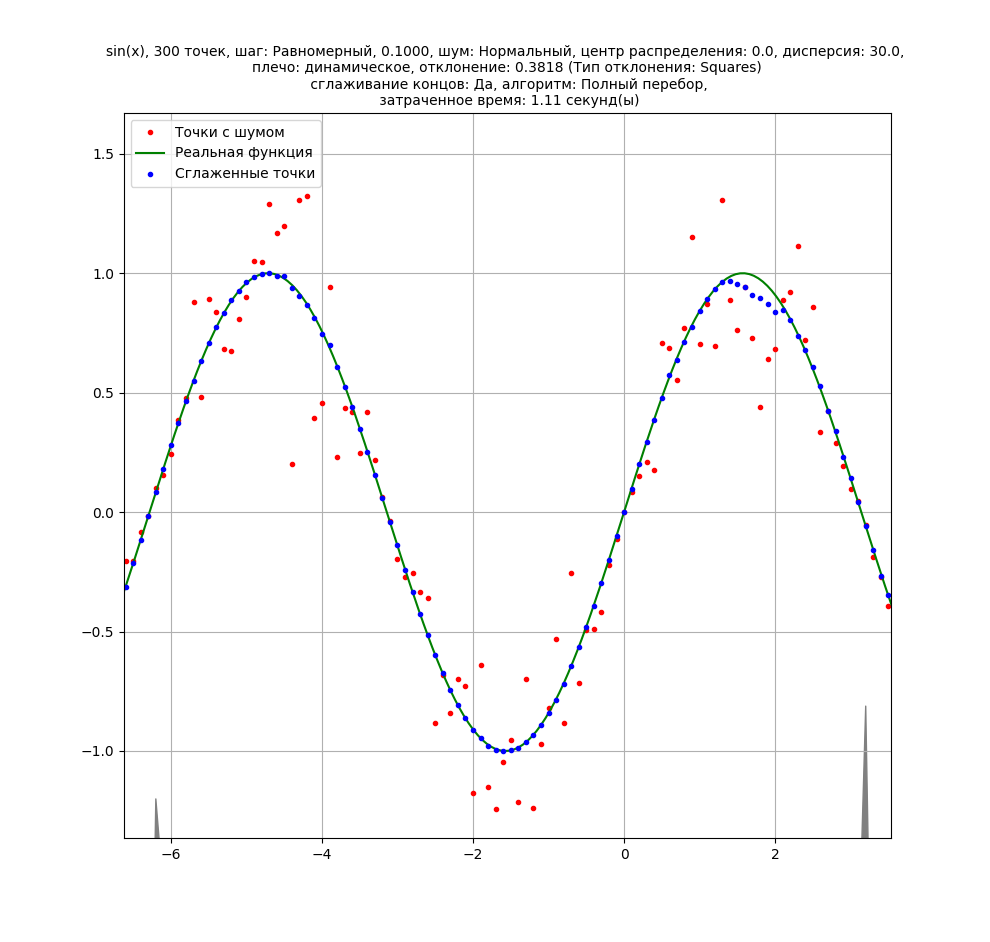
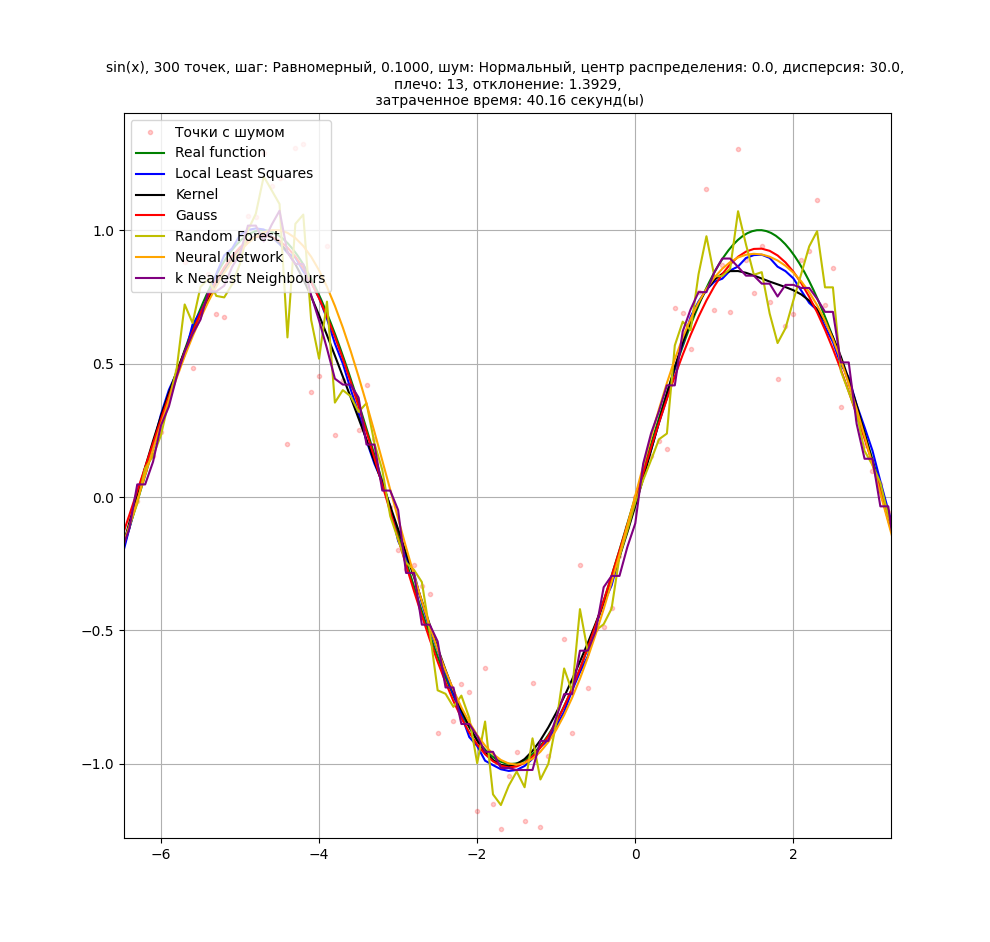
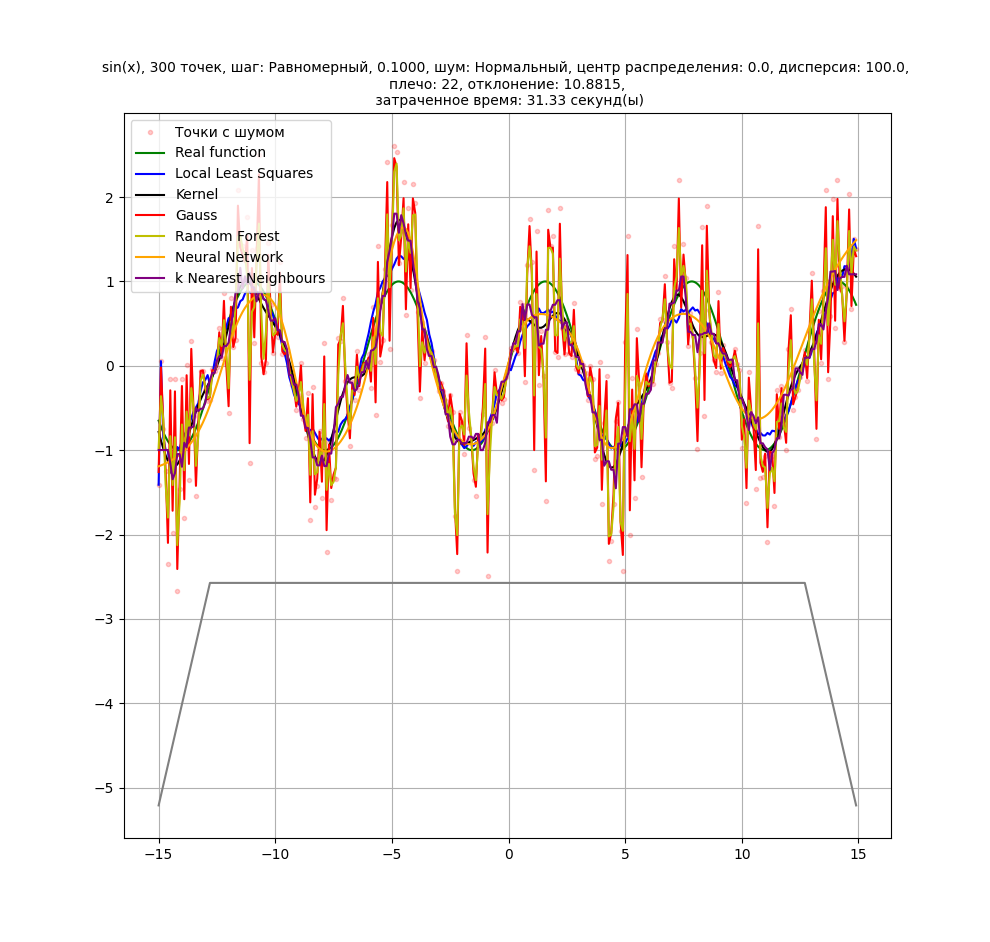
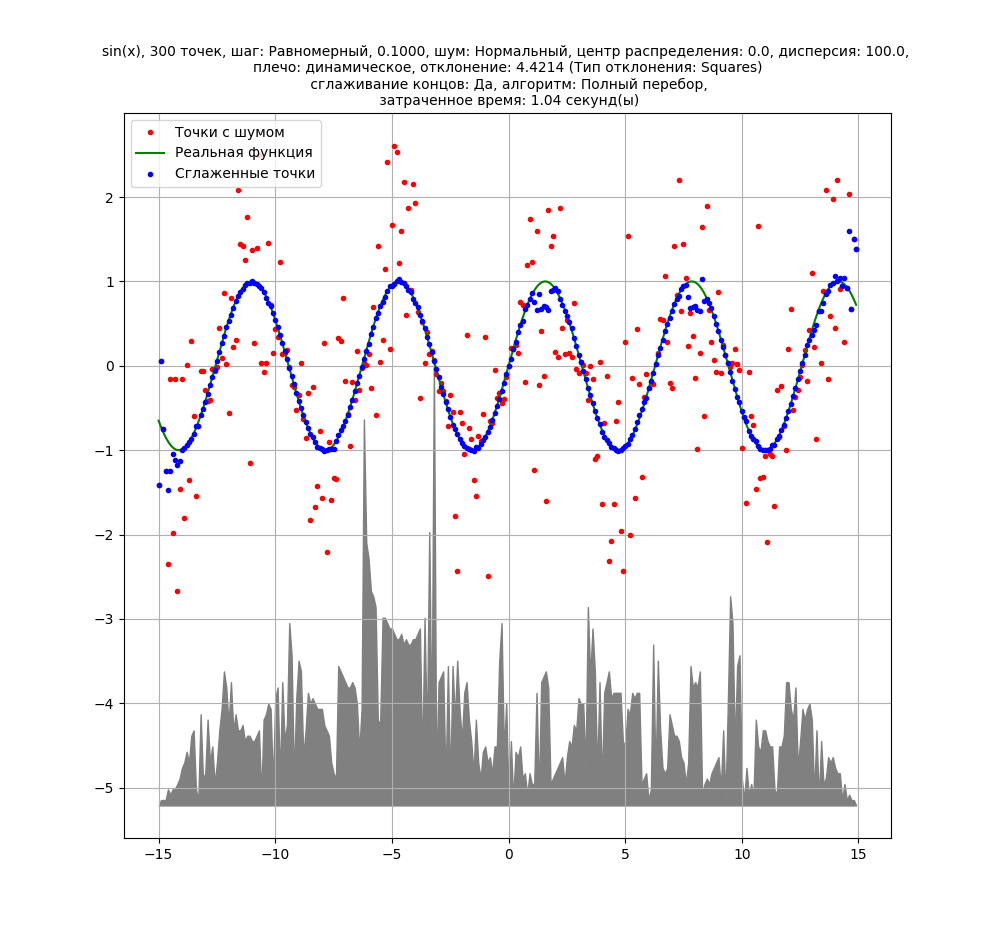
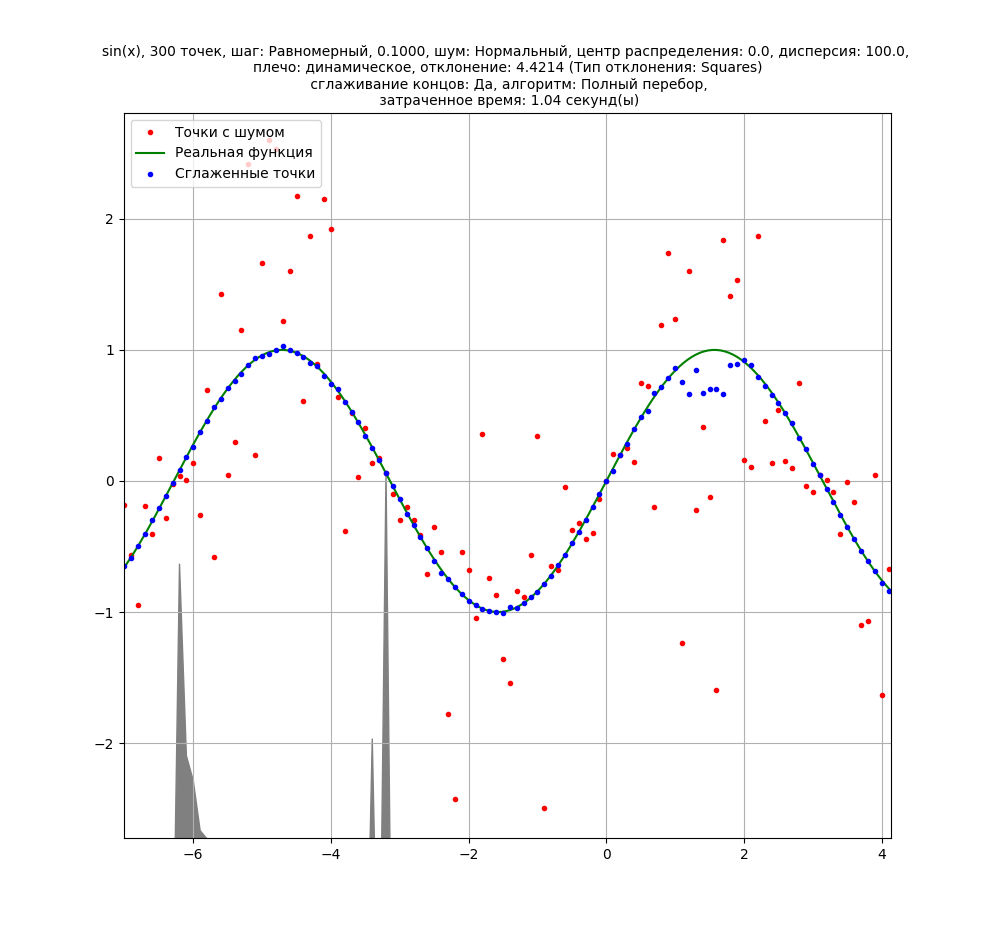
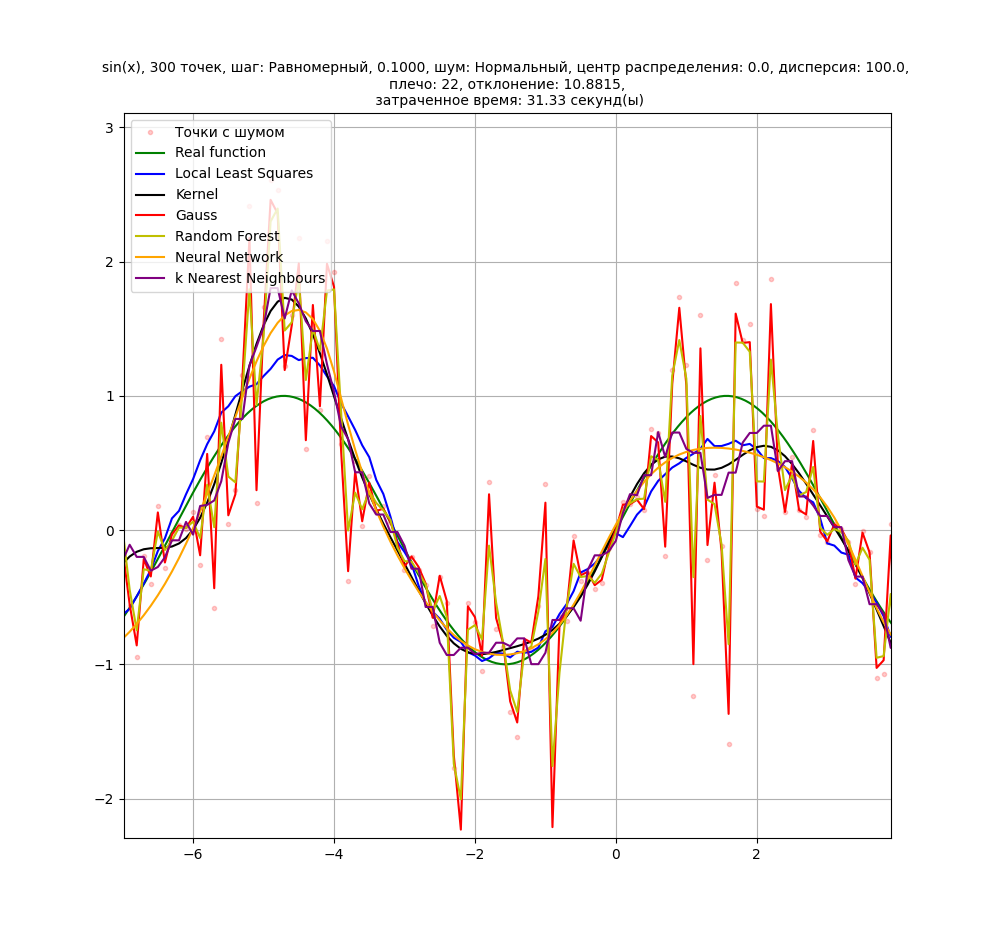
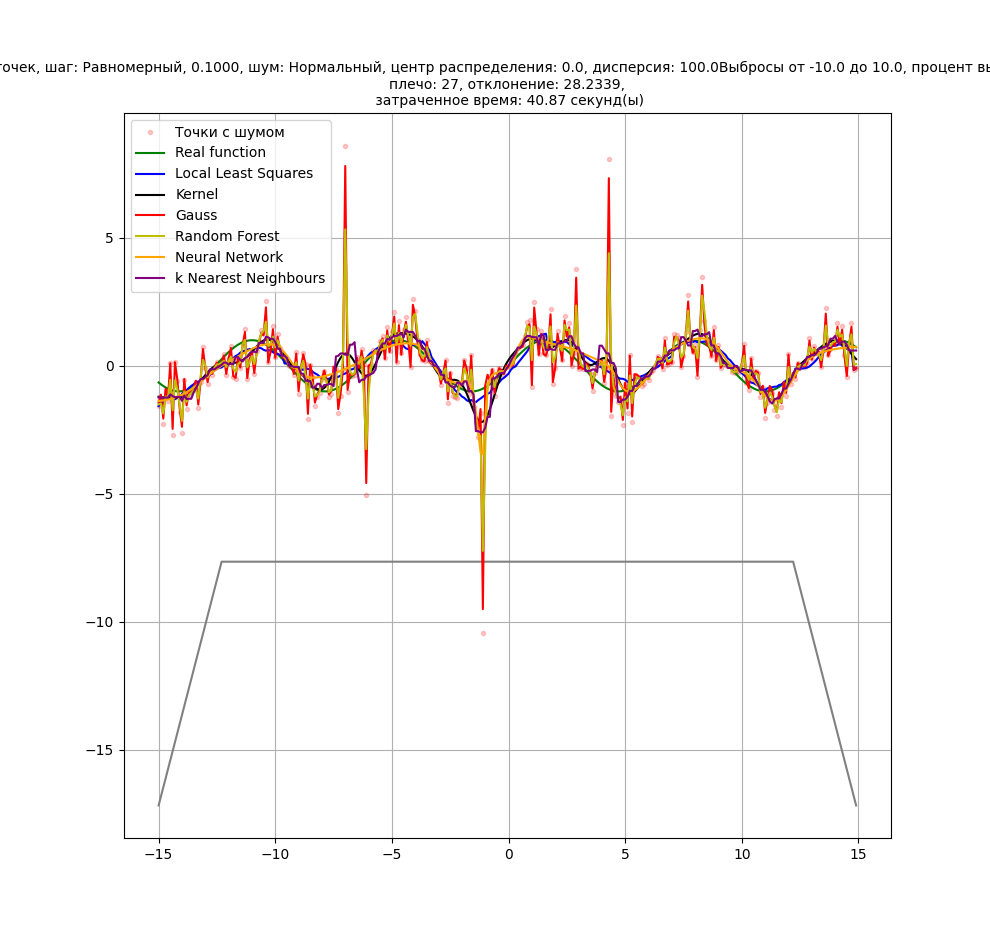
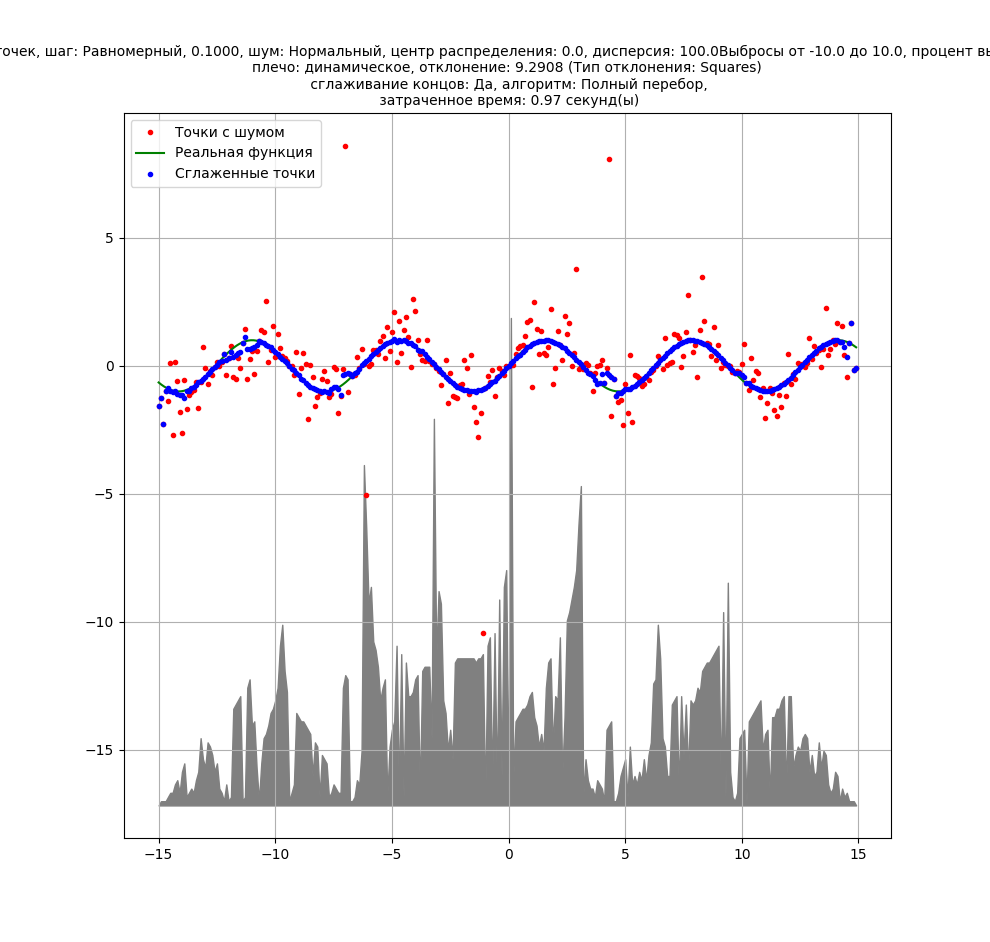
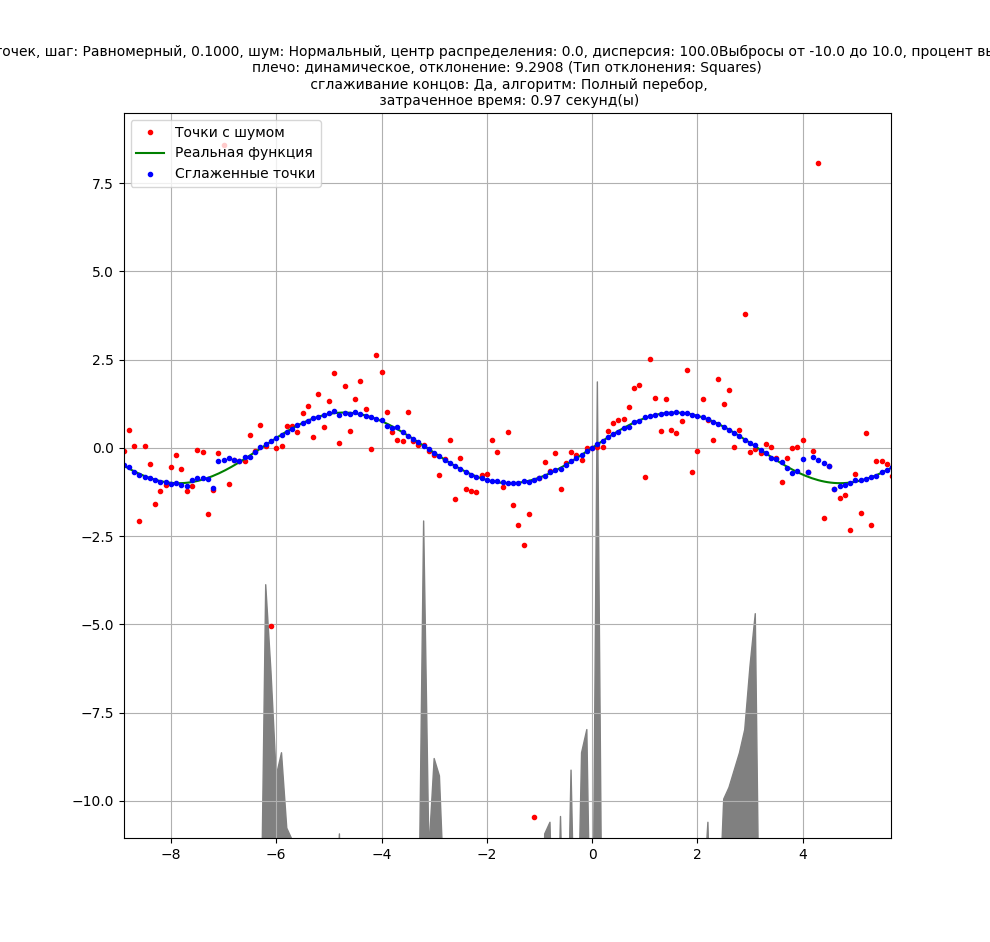
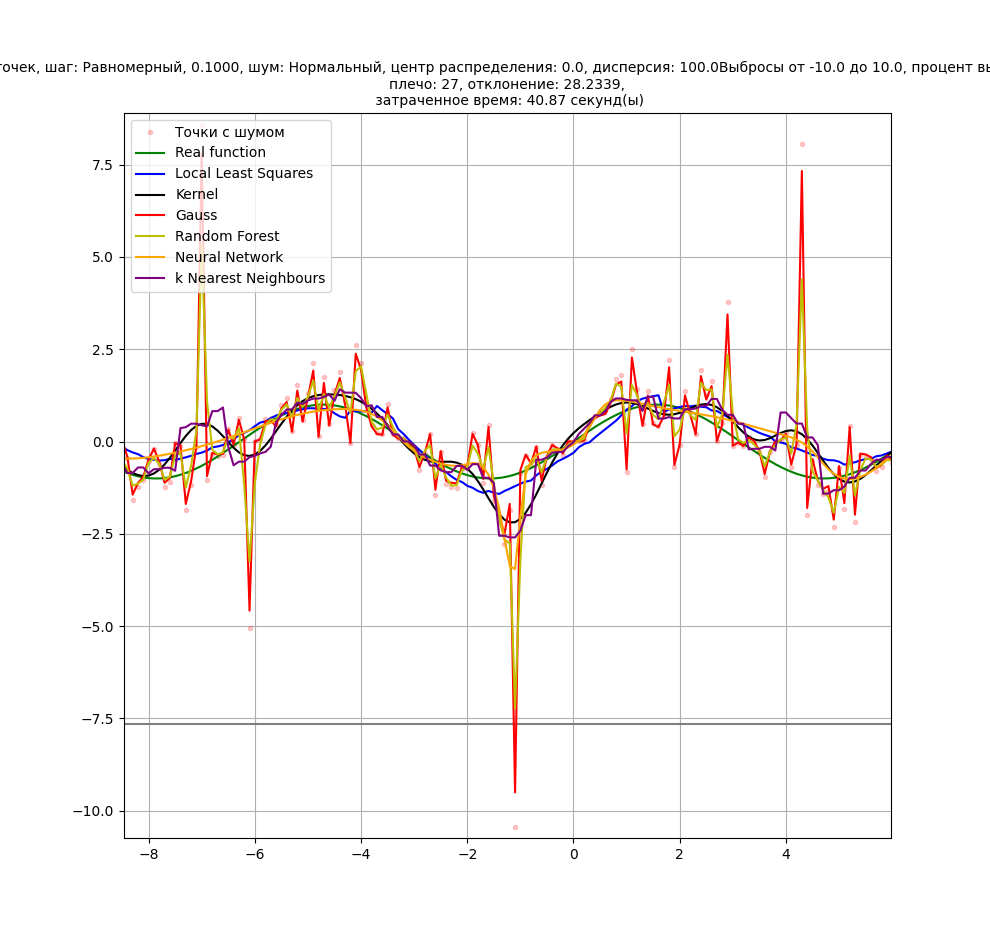


Рисунок 29 - Сглаживание синуса при нормальном шуме в 100% и выбросах в 1000% разными методами



Рисунок 30 – Приближенное сглаживание синуса при нормальном шуме в 100% и выбросах в 1000% разными методами



# Дальнейшее развитие

* Реализовать возможность оптимального подбора точки амортизации
* Реализовать возможность автоматического подбора наилучшего фиксированного плеча параболы основываясь на реальных значениях функции
* Реализовать возможность подбора оптимальных значений плеча параболы для каждой точки основываясь на реальных значениях функции
* Реализовать возможность автоматического подбора наилучшего фиксированного плеча без реальных значений функции
* Реализовать возможность разбиения функции на интервалы и последующего сглаживания каждого интервала фиксированным оптимальным для него плечом
* Реализовать возможность подбора оптимальных значений плеча для каждой точки без реальных значений функции
* Реализовать систему весов для точек плеча параболы
* Реализовать возможность автоматического подбора оптимальной степени полинома для сглаживания конкретного фрагмента
* Расширить метод на многомерное пространство
* Расширить метод на комплексную плоскость

# Выводы

В процессе выполнения данной работы был разработан программный комплекс, необходимый для качественного проведения исследований по восстановлению функциональной зависимости с одной переменной. Был программно реализован и усовершенствован приведённый в статье[1] алгоритм непараметрической аппроксимации. Были исследованы факторы, влияющие на точность регрессии, важнейшим из которых является длина интервала аппроксимации. Полученные результаты позволили предложить модификации для автоматизации и повышения эффективности существующего алгоритма и создать первый примитивный прототип усовершенствованного алгоритма, а также заложили фундамент для более глубоких исследований.

# Список литературы

1. Семёнов М. К. «Восстановление функциональной зависимости с минимальной вычислительной структурой» // Всероссийская научная конференция молодых учёных «Наука. Технологии. Инновации», НГТУ // Новосибирск, 2003.
2. Медведев А. В. «Основы теории адаптивных систем» // Монография // Красноярск, 2015.
3. Рубан А. И. «Методы анализа данных» // Монография // Красноярск, 2004.
4. Хардле В. «Прикладная непараметрическая регрессия» // Монография, перевод Назина А. В. // Москва, 1993.
5. Хампель Ф., Рончетти Э., Рауссеу П., Штаэль В. «Робастность в статистике» // Монография, перевод Золотарев В. М. // Москва, 1989.
6. Ли Р. «Оптимальные оценки, определение характеристик и управление» // Монография, перевод Кичатова Ю. Ф. и Сысоева Л. П. // Москва, 1966.