

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет  
имени К.И.Сатпаева

Институт автоматизации и информационных технологий

Кафедра «Программная инженерия»

**ДОПУЩЕН К ЗАЩИТЕ**  
Заведующая кафедрой ПИ  
канд. физ-мат. наук, профессор  
А.Н. Молдагулова  
« 20 » 05 2022 г.



**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к дипломному проекту

На тему: «Разработка программного инструмента для интерполяции скважинных данных на основе геостатистического метода кригинга»

По специальности 5В070400 –Вычислительная техника и программное обеспечение

Выполнила

Ли А.С.



Рецензент  
доктор PhD  
К. Шиббаева

Научный руководитель  
сеньор лектор, доктор PhD  
Д. Айжулов  
" 18 " 05 2022 г.

Алматы 2022

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет  
имени К.И.Сатпаева

Институт автоматки и информационных технологий

Кафедра "Программная инженерия"

5B070400 – Вычислительная техника и программное обеспечение

**УТВЕРЖДАЮ**

Заведующая кафедрой ПИ

канд. физ-мат. наук, профессор

А.Н. Молдагулова

« 20 » 05 2022 г.

**ЗАДАНИЕ**

**на выполнение дипломного проекта**

Обучающемуся: *Ли Ангелине Сергеевне*

Тема: *Разработка программного инструмента для интерполяции скважинных данных на основе геостатистического метода кригинга*

Утверждена приказом проректора по академической работе: *№489-П/О от 24.12.21*

Срок сдачи законченного проекта *23.05.2022*

Исходные данные к дипломному проекту: *описание необходимых функций*

Краткое содержание дипломной работы:

*А) обзор метода кригинга;*

*Б) визуализация результата метода кригинга;*

*В) проектирование и реализация программного инструмента;*

*Г) верификация метода.*

Перечень подлежащих разработке в дипломном проекте вопросов: (с точным указанием обязательных чертежей): *представлены 24 слайда презентации.*

Рекомендуемая основная литература: *из 20 наименований.*

**ГРАФИК**  
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
1. Анализ предметной области и актуальности работы	20.01.2021	Выполнено
2. Разработка технического задания	15.02.2021	Выполнено
3. Анализ современных технологий разработки и реализация метода предметной области	21.03.2021	Выполнено
4. Разработка программного инструмента	29.03.2021	Выполнено
5. Тестирование приложения	17.04.2021	Выполнено
6. Написание пояснительной записки к дипломному проекту	05.05.2021	Выполнено

**Подписи**

консультантов и нормоконтролера на законченный дипломный проект с указанием относящихся к ним разделов проекта

Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Нормоконтролер	Жекамбаева М.Н Доктор PhD, ассоциированный профессор	14.05.22	
Програмное обеспечение	Марғұлан Қ Магистр технических наук, лектор.	17.05.22	

Научный руководитель \_\_\_\_\_  Д. Айжулов

Задание принял к исполнению обучающийся \_\_\_\_\_  А. Ли

Дата

«17» 11 2021 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет  
имени К.И.Сатпаева

Институт автоматизации и информационных технологий

Кафедра «Программная инженерия»

Ли Ангелина Сергеевна

Разработка программного инструмента для интерполяции скважинных данных  
на основе геостатистического метода кригинга.

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к дипломному проекту

5B070400 – Вычислительная техника и программное  
обеспечение

Алматы 2022

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет  
имени К.И.Сатпаева

Институт автоматизации и информационных технологий

Кафедра «Программная инженерия»

**ДОПУЩЕН К ЗАЩИТЕ**  
Заведующая кафедрой ПИ  
канд. физ-мат. наук, профессор  
\_\_\_\_\_ А.Н. Молдагулова  
«\_\_\_\_\_» \_\_\_\_\_ 2022 г.

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к дипломному проекту

На тему: «Разработка программного инструмента для интерполяции скважинных данных на основе геостатистического метода кригинга»

По специальности 5В070400 –Вычислительная техника и программное обеспечение

Выполнила

Ли А.С.

Рецензент  
доктор PhD

Научный руководитель  
сеньор лектор, доктор PhD

\_\_\_\_\_ Қ. Алибаева

\_\_\_\_\_ Д. Айжулов

"\_\_\_\_\_" \_\_\_\_\_ 2022 г.

"\_\_\_\_\_" \_\_\_\_\_ 2022 г.

Алматы 2022

# МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет  
имени К.И.Сатпаева

Институт автоматизации и информационных технологий

Кафедра "Программная инженерия"

5B070400 – Вычислительная техника и программное обеспечение

## УТВЕРЖДАЮ

Заведующая кафедрой ПИ  
канд. физ-мат. наук, профессор  
\_\_\_\_\_ А.Н. Молдагулова  
«\_\_\_\_\_» \_\_\_\_\_ 2022 г.

## ЗАДАНИЕ

### на выполнение дипломного проекта

Обучающемуся: *Ли Ангелине Сергеевне*

Тема: *Разработка программного инструмента для интерполяции скважинных данных на основе геостатистического метода кригинга*

Утверждена приказом проректора по академической работе: \_\_\_\_\_

Срок сдачи законченного проекта \_\_\_\_\_

Исходные данные к дипломному проекту: *описание необходимых функций*

Краткое содержание дипломной работы:

*А) обзор метода кригинга;*

*Б) визуализация результата метода кригинга;*

*В) проектирование и реализация программного инструмента;*

*Г) верификация метода.*

Перечень подлежащих разработке в дипломном проекте вопросов: (с точным указанием обязательных чертежей): *представлены \_\_\_ слайда презентации.*

Рекомендуемая основная литература: *из \_\_\_ наименований.*

## ГРАФИК

подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
1. Анализ предметной области и актуальности работы	20.01.2021	Выполнено
2. Разработка технического задания	15.02.2021	Выполнено
3. Анализ современных технологий разработки и реализация метода предметной области	21.03.2021	Выполнено
4. Разработка программного инструмента	29.03.2021	Выполнено
5. Тестирование приложения	17.04.2021	Выполнено
6. Написание пояснительной записки к дипломному проекту	05.05.2021	Выполнено

### Подписи

консультантов и нормоконтролера на законченный дипломный проект с указанием относящихся к ним разделов проекта

Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Нормоконтролер	Жекамбаева М.Н Доктор PhD, ассоциированный профессор		
Програмное обеспечение	Марғұлан Қ Магистр технических наук, лектор.		

Научный руководитель \_\_\_\_\_ Д. Айжулов

Задание принял к исполнению обучающийся \_\_\_\_\_ А. Ли

Дата

«\_\_» \_\_\_\_\_ 2021 г.

## АННОТАЦИЯ

Целью данного дипломного проекта является упрощения георазведочных работ, с помощью программного инструмента (ПИ) для интерполяции межскважинных данных на основе геостатического метода исследования, который менее трудозатратный и является более простым решением.

Дипломная работа состоит из введения, основной части (исследовательский раздел, технологический раздел, проектный раздел), заключения, списка использованной литературы и необходимых приложений.

В исследовательском разделе присутствует теоретический материал исследуемой предметной области, в нем рассмотрена актуальность метода, область применения, а также особенности метода, его преимущества в отличие от других методов исследования, в том числе и его различные виды.

Во втором разделе рассмотрена технологическая часть с описанием языка разработки, описаны причины выбора языка, его преимущества, используемые библиотеки, а также в разделе проанализирован выбор среды разработки.

В проектном разделе рассмотрена более подробная проектная часть, где присутствует разъяснение работы программного инструмента и присутствующих в нем функций.

Вся дипломная работа состоит из 47 страниц, 26 рисунков, 2 приложений. В работе использовались 20 источников.

## АНДАТПА

Дипломдық жобаның мақсаты геостатикалық зерттеу әдісі негізінде ұңғымааралық мәліметтерді интерполяциялау үшін арналған бағдарламаны қолдана отырып, геологиялық барлау жұмыстарын жеңілдету болып табылады. Бұл бағдарлама көп қажет етпейді және жеңіл шешім болып табылады.

Дипломдық жұмыс кіріспеден, негізгі бөлімнен (зерттеу бөлімі, технологиялық бөлім, жобалау бөлімі), қорытындыдан, пайдаланылған әдебиеттер тізімінен және қажетті қосымшалардан тұрады.

Зерттеу бөлімінде зерттелетін пәндік аймақтың теориялық материалы бар, ол әдістің өзектілігін, қолдану аясын, сонымен қатар әдістің ерекшеліктерін, басқа зерттеу әдістеріне қарағанда оның артықшылықтарын, соның ішінде оның әр түрлі түрлерін қарастырады.

Екінші бөлімде технологиялық бөлім бағдарламау тілінің сипаттамасымен қатар қарастырылған. Тілді таңдау себептері, оның артықшылықтары, пайдаланылған кітапханалар сипатталған, сонымен қатар бұл бөлімде әзірleme ортасын таңдауға талдау жасалған.

Жобалық бөлімде бағдарламаның жұмысы мен ондағы функциялардың түсіндірмесі қарастырылған.

Барлық дипломдық жұмыс 47 беттен, 26 суреттен, 2 қосымшадан тұрады. Жұмыста 20 дереккөз қолданылды.

## **ABSTRACT**

The purpose of this diploma project is to simplify geological exploration, using a software tool for interpolating inter-well data based on the geostatic research method, which is less labor-intensive and is a simpler solution.

The diploma work consists of an introduction, the main part (research section, technology section, project section), conclusion, list of references and necessary appendices.

The research section contains theoretical material of the subject area under study, it examines the relevance of the method, the scope of application, as well as the features of the method, its advantages in contrast to other research methods, including its various types.

The second section discusses the technological part with a description of the development language, describes the reasons for choosing the language, its advantages, the libraries used, and also analyzes the choice of the development environment.

In the project section, a more detailed project part is considered, where there is an explanation of the operation of the software tool and the functions present in it.

The entire diploma work consists of 47 pages, 26 drawings, 2 appendices. 20 sources were used in the work.

## СОДЕРЖАНИЕ

	Введение	9
1	Исследовательский раздел	10
1.1	Цель разработки системы	10
1.2	Определения, термины и сокращения	10
1.3	Актуальность и область применения	12
1.3.1	Актуальность	12
1.3.2	Область применения	13
1.4	Предметная область	13
1.4.1	Методы интерполяции	13
1.4.2	Геостатистика	15
1.4.3	Вариограмма	16
1.4.4	Метод кригинг	17
2	Технологический раздел	21
2.1	Среда разработки Jupyter	21
2.2	Язык разработки Python	21
2.2.1	Python	21
2.2.2	Tkinter	22
3	Проектная часть	25
3.1	Архитектура системы взаимодействия	25
3.2	Описание диаграммы деятельности	25
4	Экспериментальный раздел	27
4.1	Разработка алгоритма кригинга	27
4.2	Создание программного инструмента	28
4.3	Верификация метода	35
	ЗАКЛЮЧЕНИЕ	36
	СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	37
	ПРИЛОЖЕНИЕ А. Техническое задание	39
	ПРИЛОЖЕНИЕ Б. Текст программы	41

## ВВЕДЕНИЕ

Добыча полезных ископаемых требует проводить полномасштабную подготовку и выполнение множества сложных производственных работ с предварительным исследованием месторождения. В том числе и выполнение такого процесса, как разведка месторождения, включающая в себя выполнение различных геологоразведочных работ.

Важнейшим конечным результатом в предварительном исследовании месторождения является определение рентабельности, перспективности дальнейшей добычи месторождения.

Конечной задачей геологоразведочных работ на месторождении является получение специальных геологических карт, с помощью которых можно точно определить перспективность изученной территории, а также определить точное расположение областей, содержащих руду и уже по ним оценить объемы содержания ресурсов для дальнейшего освоения в промышленных масштабах.

Некоторые геологоразведочные работы месторождения включают в себя этап, при котором используются так называемые геостатистические методы исследования. Эти методы исследования способствуют получению наиболее полной информации о геологическом строении месторождения на основе данных, который получают при предварительной разведке месторождения.

## 1 Исследовательский раздел

### 1.1 Цель разработки системы

При исследовании месторождений на начальных этапах необходимо определить примерные объемы содержания и запасы полезного ископаемого на месторождении.

Используя геостатистический метод интерполяции, который является на сегодня одним из мощных средств для получения достоверной информации о распределении полезных ископаемых в межскважинном пространстве, можно получить данные о объеме руд на месторождении.

При разведке месторождения количество созданных скважин на определенной области ограничено, тем самым ограничивается и количество получаемых данных о содержании руды. Для той области, где отсутствует информация и по которой нужно получить данные применяется статистический интерполяционный метод исследования. В основном требуется определить значение величины на основе имеющихся данных, т. е. решить задачу интерполяции.

Главной целью является разработка программного инструмента для вычисления и визуализации межскважинных данных с помощью интерполяционного геостатистического метода кригинга, который решит проблему недостатка данных о содержании полезного ископаемого в межскважинных пространствах.

### 1.2 Определения, термины и сокращения

В таблице 1.2.1 сформулированы все термины и сокращения, которые используются в предметной области разрабатываемого проекта, а также специфические термины, связанные с программной реализацией проекта и используемыми технологиями при разработке.

**Таблица 1.2.1 – Сокращения, термины и их определения**

<b>Сокращение или термин</b>	<b>Определение</b>
ПСВ	Подземное скважинное выщелачивание
ПО	Программное обеспечение
ПИ	Программный инструмент
Jupyter Notebook	Интерактивная веб-среда для разработки

Продолжение таблицы 1.2.1

Сокращение или термин	Определение
TIOBE	Расшифровывается как The Importance Of Being Earnest, это рейтинг языков программирования. В настоящее время в рейтинге учитываются 150 языков программирования. Измерения проводятся не менее 10 раз в месяц
Big Data	Большие данные — обозначение структурированных и неструктурированных данных огромных объёмов и значительного многообразия
Python	Высокоуровневый и универсальный язык программирования
Кригинг	Геостатический метод интерполяции
Вариограмма	Функция для работы метода кригинга
UML	(Unified Modeling Language) - унифицированный язык моделирования, то есть это язык графического описания для объектного моделирования в области разработки программного обеспечения
Tcl	Tcl (Tool Command Language) - это динамический язык программирования.
Tk	Tk - это инструментальный графического пользовательского интерфейса. Tk является стандартным графическим интерфейсом не только для Tcl, но и для многих других динамических языков и может создавать богатые собственные приложения, которые работают без изменений в Windows, Mac OS X, Linux и других.
GUI	GUI (Graphical User Interface) или ГИП (графический интерфейс пользователя) — система средств для взаимодействия пользователя с компьютером, основанная на представлении всех доступных пользователю системных объектов и функций в виде графических компонентов экрана.
JSON	JavaScript Object Notation

## 1.3 Актуальность и область применения

### 1.3.1 Актуальность

Одной из ведущих отраслей горнодобывающей промышленности Казахстана является добыча урана. Казахстан занимает первое место в мире по добыче урана и второе по его запасам. По данным Всемирной ядерной ассоциации, добыча урана в Казахстане в 2019 году составила 22 808 тонн что составляет 41,66 % от мировой добычи (см Рисунок 1.1) [1].

Uranium resources by country in 2019

	tonnes U	percentage of world
Australia	1,692,700	28%
Kazakhstan	906,800	15%
Canada	564,900	9%
Russia	486,000	8%
Namibia	448,300	7%
South Africa	320,900	5%
Brazil	276,800	5%
Niger	276,400	4%
China	248,900	4%
Mongolia	143,500	2%
Uzbekistan	132,300	2%
Ukraine	108,700	2%
Botswana	87,200	1%
Tanzania	58,200	1%
Jordan	52,500	1%
USA	47,900	1%
Other	295,800	5%
World total	6,147,800	

Production from mines (tonnes U)

Country	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
Kazakhstan	19,451	21,317	22,451	23,127	23,607	24,689	23,321	21,705	22,808	19,477
Australia	5983	6991	6350	5001	5654	6315	5882	6517	6613	6203
Namibia	3258	4495	4323	3255	2993	3654	4224	5525	5476	5413
Canada	9145	8999	9331	9134	13,325	14,039	13,116	7001	6938	3885
Uzbekistan (est.)	2500	2400	2400	2400	2385	3325	3400	3450	3500	3500
Niger	4351	4667	4518	4057	4116	3479	3449	2911	2983	2991
Russia	2993	2872	3135	2990	3055	3004	2917	2904	2911	2846
China (est.)	885	1500	1500	1500	1616	1616	1692	1885	1885	1885
Ukraine	890	960	922	926	1200	808	707	790	800	744
India (est.)	400	385	385	385	385	385	421	423	308	400
South Africa (est.)	582	465	531	573	393	490	308	346	346	250
Iran (est.)	0	0	0	0	38	0	40	71	71	71
Pakistan (est.)	45	45	45	45	45	45	45	45	45	45
Brazil	265	326	192	55	40	44	0	0	0	15
USA	1537	1596	1792	1919	1256	1125	940	582	58	6

Рисунок-1.3.1 – Статистика по мировой добычи урана 2019 и 2020 гг

Подземное скважинное выщелачивание (ПСВ) является основным методом добычи урана во многих странах, включая США и Казахстан. Примерно 51% мирового производства урана добывается методом ПСВ. ПСВ требует закачки различных растворов в подземные пористые среды с целью растворения минералов, содержащих уран, с последующей добычей с помощью эксплуатационных скважин.

Суть проблемы заключается в дороговизне процедур разведки. До сих пор, в отличие от других подземных ресурсов, разведка урана опирается исключительно на бурение скважин и использование различных зондов для определения концентрации урана в пласте. Из-за дороговизны процесса бурения.

Решением этой проблемы может стать использование статистического подхода к освоению месторождения урана. Такие методы имеют более перспективное использование затрачиваемых ресурсов. И метод кригинг основанный на надежности прогнозирования технологических характеристик руд имеет более низкую стоимость и трудоемкость выполнения работ. Тем самым оптимизирую процесс разведки месторождения [5].

### 1.3.2 Область применения

Метод кригинга широко используется в области пространственного анализа. Такой анализ много, где применяется в человеческой деятельности. Например, для моделирования карт изменения температуры (см Рисунок 1.2), скорости ветра, осадков. В медицине, где анализируются факторы, влияющие на заболевания, а также этот метод используется в томографии. Кригинг применяется и в графических редакторах.

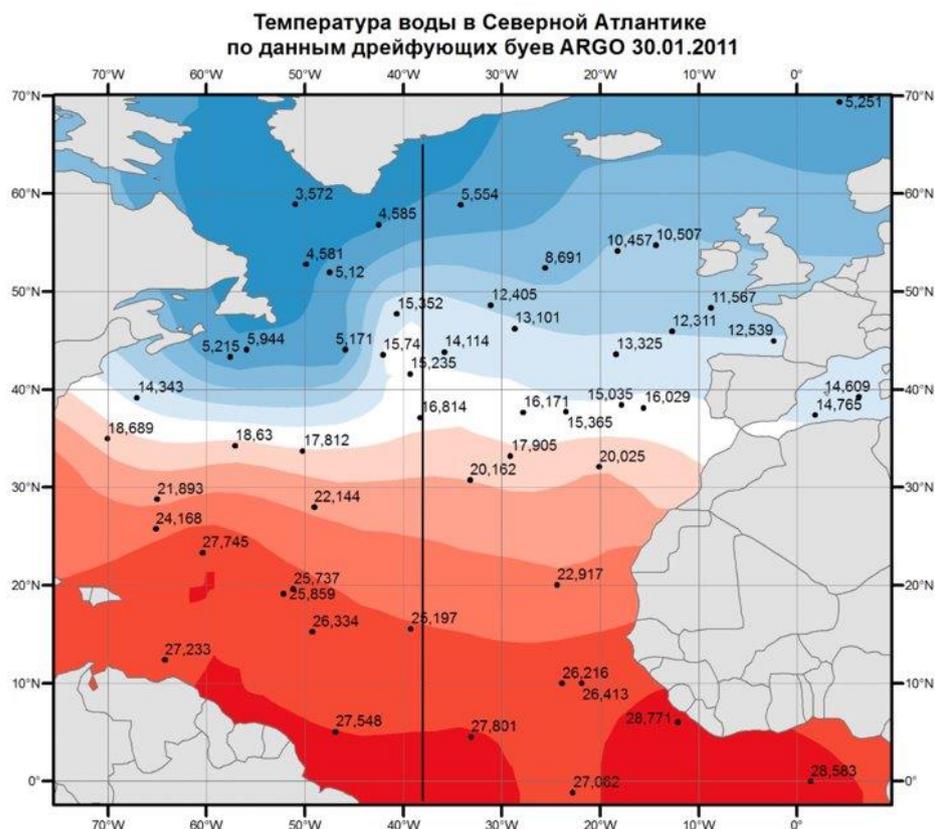


Рисунок-1.3.2 – Смоделированная карта температур Атлантического океана

### 1.4 Предметная область

#### 1.4.1 Методы интерполяции

Чтобы более подробно перейти к методу кригинга, сначала нужно определить понятие интерполяции. Интерполяция — это метод управления простой функцией из заданного набора дискретных данных таким образом, чтобы функция проходила через предоставленные точки данных. Это помогает

определить точки данных между заданными точками данных. Этот метод всегда необходим для вычисления значения функции для промежуточного значения независимой функции. Короче говоря, интерполяция — это процесс определения неизвестных значений, которые лежат между известными точками данных. В основном он используется для прогнозирования неизвестных значений для любых связанных с географией точек данных, таких как уровень шума, количество осадков, высота над уровнем моря и так далее.

Методы интерполяции делятся в основном на две большие группы (см Рисунок 1.4.1): детерминированные и статистические (геостатистические).



**Рисунок-1.4.1 – Методы интерполяции**

Детерминированные методы интерполяции проектируют поверхности на основе степени схожести или на уровне сглаживания.

В детерминированных методах предполагается, что все данные описываются какой-то детерминистической функцией  $Z(x, \lambda)$ , где  $x \in S$  — координаты точки;  $S$  - область исследования;  $\lambda$  — набор внутренних параметров модели. Задача состоит в том, чтобы подобрать параметры  $\lambda$  и на основании их построить функцию  $Z(x, \lambda)$  для всей области  $S$ . И уже после этого можно будет найти значение в любой точке по формуле.

В основном детерминированные методы интерполяции делятся на два вида: глобальные и локальные. Глобальные методы производят интерполяцию основываясь на всех данных о точках. А локальные методы используют для интерполяции значения в точках, которые расположены ближе к оцениваемой области.

В глобальных интерполяторах искомая функция подвергается преобразованиям для того, чтобы она стала сглаженной. В локальных методах сама окрестность, которая используется для оценки, определяется различными способами. Например, окрестность зависит от числа ближайших точек. В такой случае размер области, влияющей на оценку, начинает зависеть от плотности точек в исследуемой области. Возможен вариант, когда окрестности задается конкретными размерами, в пределах которой все точки используются для оценивания.

К детерминированным методам интерполяции относятся такие методы как, метод обратных взвешенных расстояний, интерполяции ядра, диффузии, радиальные базисные функции.

Методы этого вида имеют популярность из-за простоты использования

формулы для вычисления оценки. Их главными недостатками является то, что они не дают возможности как-то охарактеризовать качество оценки, таким способом многие методы просто пренебрегают пространственной корреляцией.

Статические или геостатистические методы интерполяции основываются на статистических свойствах точек. Этот метод основан на изменениях автокорреляции. Геостатистические методы включают в себя группу методом кригинга.

### 1.4.2 Геостатистика

Основателями геостатистики считаются французский математик Г. Матерона и южноафриканский горный инженер Д. Г. Криге, которые в своих работах разрабатывали оптимальную процедуру интерполяции, которую возможно было применять в горнодобывающей промышленности.

Геостатистика повсеместно используется во многих сферах человеческой деятельности и также используется в пространственной статистике. Изначально геостатистикой назывался метод кригинга, который представляет собой статистическую версию интерполяции. Но сейчас геостатистика включает в себя не только метод кригинга, но и детерминированные методы.

Все методы кригинга основываются на общей формуле  $Z(x)$ :

$$Z^*(x) - m(x) = \sum_{i=1}^{n(x)} \lambda_i(x) [Z(x_i) - m(x_i)],$$

где  $\lambda_i(x)$  — весовые коэффициенты, относящиеся к данным  $z(x_i)$ .

$m(x)$  и  $m(x_i)$  являются математическими ожиданиями случайных переменных  $Z(x)$  и  $Z(x_i)$ .

Рассмотрим различные методы кригинга для моделирования одной переменной.

Простой кригинг (Simple Kriging) - этот метод основывается на том, что его веса не зависят от значений исходного набора данных, а зависят только от корреляции, который строится на основе данных. То есть, если есть несколько наборов исходных данных в одних и тех же точках и которые описываются одинаковыми функциями ковариации, то для моделирования методом простого кригинга в точке, систему уравнения кригинга достаточно решить один раз, тем самым найти веса и просто в дальнейшем их и использовать для остальных точек. Простой кригинг имеет лучшую оценку точности в линейных методах.

Основным недостатком простого кригинга является то, что этот метод предполагает постоянное значение среднего - математического ожидания. Тем самым используя этот метод, все веса для интерполяции данных становятся зависимыми от значений изначальных данных. Помимо этого, оценка математического ожидания может оказаться искажённой, смещённой и т. п.

Поэтому простой кригинг обычно используется, когда данные созданы с какими-то искусственными комбинациями, где среднее находится вследствие предварительных изменений исходных данных.

Обычный кригинг (Ordinary kriging) — это один из наиболее часто используемых в геостатистике методов интерполяции. Этот метод отличается от простого кригинга тем, что не предполагает знание среднего значения. Тут среднее значение является постоянным, но его следует найти в процессе. Помимо этого, при использовании обычного кригинга в локальной оценке среднее значение не постоянно по всей области оценивания, а оно постоянно только в окрестности конкретной точки, для которой производится оценка. Это условие о постоянстве среднего в пределах небольшой окрестности делает метод более правдоподобным, чем условие о том, что данные обладают пространственной непрерывностью.

Таким образом, со всего вышесказанного следует, что на качество результата обычного кригинга во многом влияет правильность подобранной модели вариограммы, то есть точность пространственной корреляционной структуры, соответственно моделирование вариограммы является важнейшим этапом в этом методе интерполяции.

### 1.4.3 Вариограмма

Рассмотрим более подробно одну из основных тем в геостатистике построение и моделирование вариограмм. Вариография — это моделирование и анализирование пространственной корреляции данных.

Главным свойством распределения данных является пространственная непрерывность, которая означает то, что расположенные близко друг к другу значения будут иметь близкие значения. Эта пространственная непрерывность обычно описывается с помощью корреляционных или ковариационных функций, которые отображают меру непрерывности. Корреляция представляется статистическими моментами. Самой популярный статистический момент - это функция вариограммы, или по-другому статистический двуточечный момент второго порядка.

Задачей этапа моделирование вариографией является получение аналитической функции, которая описывает пространственную корреляцию структуры данных для дальнейшего использования в геостатистических методах интерполяции. Другими словами, конечной целью этапа вариографии является построение модели вариограммы. Качество этой модели определяет качество геостатистической оценки.

Ковариация (covariance) — это мера корреляции между значениями в точках, которые разделены вектором  $h$ . Она описывает степень схожести данных, то есть чем более похожи данные (ближе значения), тем больше

значение ковариации.

Полувариограмма (semivariogram) или вариограмма — это закономерность вариации разницы значений двух точках, то есть она описывает количественное различие данных в зависимости от расстояния между ними. Чем меньше разница между значениями данных, тем больше значение вариограммы.

#### 1.4.4 Метод кригинг

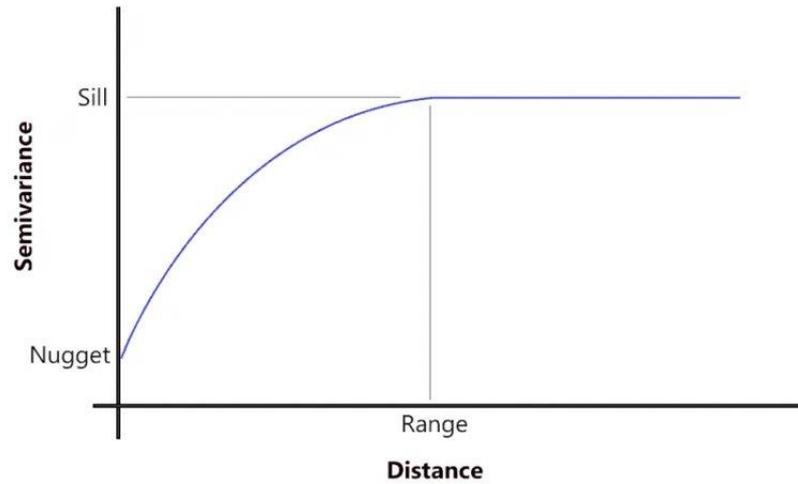
Кригинг - это один из нескольких методов геостатистики, который использует ограниченный набор выборочных точек данных для построения более точной поверхности.

Кригинг в отличие от более простых методов, использует пространственную корреляцию между выбранными точками для интерполяции значений в пространственном поле. В общем смысле, веса в точках кригинга рассчитываются таким образом, чтобы точки, расположенные поблизости от интересующего местоположения, имели большее влияние чем те, которые находятся дальше.

Кригинг состоит из двух этапов:

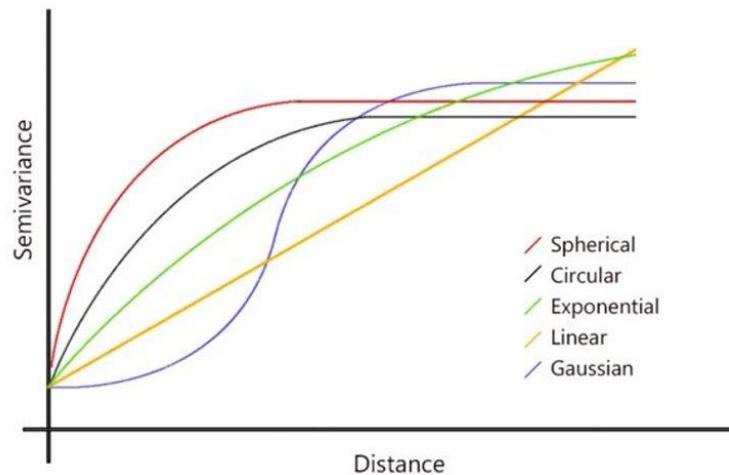
- первый – вычисление и построение вариограммы;
- второй – вычисления и моделирование поверхности с помощью соответствующего алгоритма;

В основе статистического метода кригинга лежит вычисление и построение так называемой вариограммы (см Рисунок 1.4.2). Вариограмма определяет автокорреляцию, поскольку она отображает дисперсию всех расстояний между точками в зависимости от самого расстояния. То есть, более близкие точки имеют большее влияние и имеют небольшую полу-дисперсию. В то время как далекие точки имеют меньшее влияние и имеют высокую полу-дисперсию. Но на определенном расстоянии (range) автокорреляция становится независимой. Значение, где эта вариация выравнивается, называется (sill). Это означает, что больше нет никакой пространственной автокорреляции (влияние) для больших расстояний чем range [3].



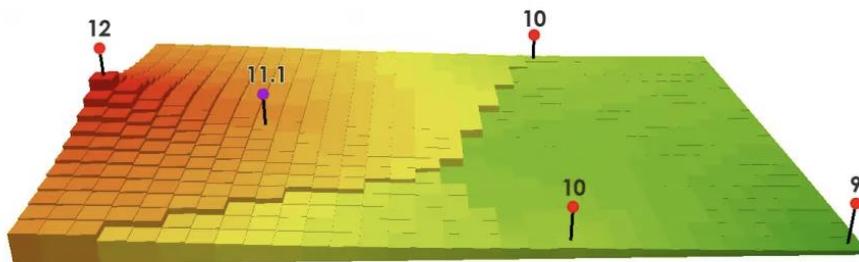
**Рисунок-1.4.2 – Пример вариограммы с отображением range и sill**

Вариограмма может строиться по разным методам, и в зависимости от метода вариограмма делится на виды, такие как сферическая, экспоненциальная, гауссовская, круговая, линейная в соответствии с рисунком 1.4.3.



**Рисунок-1.4.3 – Различные виды вариограммы**

Кригинг является «наилучшим» оценителем в статистическом смысле в классе линейных интерполяторов — его оценка обладает минимальной вариацией ошибки. Важное свойство кригинга — точное воспроизведение значений измерений в имеющихся точках (точный оценитель). В отличие от многочисленных детерминистических методов, оценка кригинга сопровождается оценкой ошибки интерполяции в каждой точке.



**Рисунок-1.4.4 – Пример прогнозирования одной точки**

Кригинг включает в себя следующие шаги:

- 1) Считывание данных, которые содержат координаты и значения в точках;
- 2) Вычисление и построение вариограммы  $V(h)$ ;
- 3) Вычисление матрицы  $K$  на основе всех известных точек;
- 4) Вычисление вектора  $M$  на основе искомой точки ( $x$ ) и всех известных точек ( $x_i$ );
- 5) Решить матричное уравнение, тем самым вычислить вектор весов ( $\lambda_i$ );
- 6) Найти значение в точке для каждой неопределённой точки, тем самым повторить алгоритм;
- 7) Отображение результатов.

Для того чтобы перейти к написанию алгоритма кригинга, сперва пройдемся по основным формулам. Самое главное, что нужно первоначально определить это основная формула кригинга:

$$Z^*(x) = \sum_{i=1}^n \lambda_i Z(x_i) \quad (1.1)$$

где  $Z(x)$  — это искомое нами значение в точке;

$Z(x_i)$  — это значение содержания руды в известной точке;

$\lambda_i$  — это коэффициент весов, который нужно определить на основе известных данных.

В зависимости от выбранной модели вариограммы, для расчета используется различные формулы. В каждой из них присутствуют 3 переменных: range, sill,  $h$ .

Формула сферической и экспоненциальной вариограммы [2]:

$$\gamma(h) = \begin{cases} c_0 + c \left[ \frac{3h}{2a} - \frac{1}{2} \left( \frac{h}{a} \right)^3 \right] & \text{для } h \leq a \\ c_0 + c & \text{для } h > a, \end{cases} \quad (1.2)$$

$$\gamma(h) = \begin{cases} 0, & h = 0, \\ c_0 + (c - c_0) \left[ 1 - \exp\left(\frac{-3h}{a}\right) \right], & h \neq 0 \end{cases}, \quad (1.3)$$

где  $h$  – расстояние между точками,  $m$ ;  
 $c$  – sill - дисперсия случайной величины;  
 $a$  – range - расстояние, при котором влияние на значение становится незначительным,  $m$ .

Для вычисления вариограммы нужно отдельно определить sill и range по формуле:

$$sill = \sigma_x = E(x - E(x))^2, \quad (1.4)$$

где  $E$  - это среднее значение;

$x$  - значение в известных точках.

Далее решаем матричное уравнение:

$$\begin{matrix} \begin{bmatrix} C(x_1 - x_1) & C(x_1 - x_2) & \dots & C(x_1 - x_n) & 1 \\ C(x_2 - x_1) & C(x_2 - x_2) & \dots & C(x_2 - x_n) & 1 \\ \dots & \dots & \dots & \dots & 1 \\ C(x_n - x_1) & C(x_n - x_2) & \dots & C(x_n - x_n) & 1 \\ 1 & 1 & \dots & 1 & 0 \end{bmatrix} & \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \dots \\ \lambda_n \\ \mu \end{bmatrix} & = & \begin{bmatrix} C(x - x_1) \\ C(x - x_2) \\ \dots \\ C(x - x_n) \\ 1 \end{bmatrix} \\ \text{М} & \lambda & & \text{К} \end{matrix}, \quad (1.5)$$

где  $C(x)$  - это функция вариограммы;

$\lambda$  - искомые нами значения весов для каждой точки.

Для того чтобы найти вектор  $\lambda$  нужно решить матричное уравнение в формуле (2.3) для этого потребуется транспонировать матрицу  $K$  и затем умножить на вектор  $M$ .

После того как найдем вектор  $\lambda$ , используя формулу кригинга можем найти значение для каждой неизвестной точки на сетке. И затем приступить к визуализации.

## **2 Технологический раздел**

### **2.1 Среда разработки Jupyter**

В нынешнее время существует большой выбор редакторов кода, предназначенных для разработки различных приложений. Каждые из них имеют свои преимущества и недостатки. Выбор того или иного программного средства разработки для конкретной задачи является одним из важнейших вопросов при разработке какого-либо приложения или системы, так как одни и те же функции для одной задачи могут быть как достоинством, так и недостатком для другой.

При реализации проекта в качестве средства для разработки для реализации инструмента для интерполяции скважинных данных с помощью метода кригинга использовалось средство разработки Jupyter Notebook, который имеет весь необходимый перечень функций и библиотек для упрощения проектируемой системы.

JupyterLab — это довольно новая интерактивная веб-среда разработки для ноутбуков, кода и данных с открытым исходным кодом. У него гибкий интерфейс, который позволяет пользователям настраивать и организовывать рабочие процессы в различных областях [4].

### **2.2 Язык разработки Python**

#### **2.2.1 Python**

Python — это многофункциональный высокоуровневый язык программирования. Он имеет простой дизайн, что упрощает написание хорошо читаемого кода, это играет более важную роль в программировании, чем кажется на первый взгляд. Этот язык программирования позволяет создавать более короткие программы, по сравнению с аналогичными программами на статистическом языке, которые будут намного длиннее, тем самым Python позволяет повысить продуктивность написания кода.

Python это универсальный язык программирования, эта универсальность следует из того, что, во-первых, его можно использовать для различных задач, во-вторых, он может запускаться почти везде. Главное его преимущество — это большое количество стандартных библиотек для реализации разнообразного ПО.

Все эти преимущества сделали Python одним из популярных языков программирования в наше время в соответствии с рисунком 2.2.1. Так, например, по рейтингу TIOBE (The Importance Of Being Earnest) Python является самым популярным языком программирования за апрель 2022 года.

### Рост основных языков программирования

Данные основаны на просмотрах вопросов со Stack Overflow из стран с высоким уровнем доходов

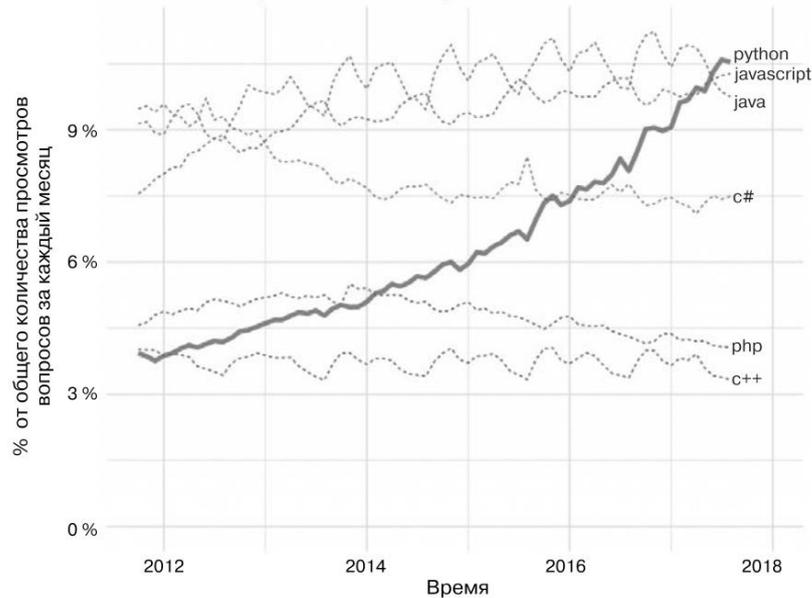


Рисунок-2.2.1 – Рост популярности Питона

Для работы с данными в Python существуют несколько самых популярных специализированных библиотек: NumPy — для вычислений, Pandas — для анализа табличных данных, Matplotlib — для визуализации.

Визуализация данных является важным пунктом в статистике для наглядного отображения распределения данных. Она помогает получить более качественное понимание данных. И в свою очередь Python может предоставить такую функцию, у него есть, так называемая стандартная библиотека для визуализации данных - Matplotlib. Она имеет достаточно обширный список функций для построения различных 2d и 3d графиков.

### 2.2.2 Tkinter

Библиотека виджетов Tk основана на языке программирования Tool Command Language (Tcl). Tcl и Tk были созданы Джоном Остерхаутом, когда он был профессором в Беркли в конце 1980-х годов, как более простой способ программирования инженерных инструментов, используемых в университете. Благодаря своей скорости и относительной простоте Tcl/Tk быстро набирал популярность среди академических, инженерных и Unix-программистов. Как и сам Python, Tcl/Tk возник на платформе Unix и только позже перешел на Mac OS и Windows.

Tkinter представляет собой интерфейс Python для библиотеки Tk GUI и является частью стандартной библиотеки Python с 1994 года с выпуском Python версии 1.1.

У Python программистов, которые хотят создать графический интерфейс, есть несколько вариантов инструментов на выбор. Tkinter не только подходит для широкого спектра применений, но и обладает некоторыми преимуществами, которые нельзя игнорировать:

- Tkinter находится в стандартной библиотеке: за редким исключением, Tkinter доступен везде, где доступен Python. Нет необходимости устанавливать pip, создавать виртуальные среды, компилировать двоичные файлы или искать установочные пакеты в Интернете. Для простых проектов, которые нужно выполнить быстро, это явное *преимущество*.

- Tkinter стабилен: хотя развитие Tkinter не остановилось, оно идет медленно и эволюционно. API был стабилен в течение многих лет, изменения в основном касались дополнительной функциональности и исправлений ошибок. Ваш код Tkinter, скорее всего, будет работать без изменений в течение многих лет или десятилетий.

- Tkinter — это всего лишь набор инструментов с графическим интерфейсом: в отличие от некоторых других библиотек с графическим интерфейсом, Tkinter не имеет собственной библиотеки потоков, сетевого стека или API файловой системы. Для таких вещей он использует обычные библиотеки Python, поэтому он идеально подходит для применения графического интерфейса к существующему коду Python.

- Tkinter прост и понятен: Tkinter очень прост и точен; его можно эффективно использовать как в процедурных, так и в объектно-ориентированных графических интерфейсах. Чтобы использовать Tkinter, вам не нужно изучать сотни классов виджетов, язык разметки или шаблонов, новую парадигму программирования, клиент-серверные технологии или другой язык программирования.

Конечно, Tkinter не идеален. У него также есть некоторые *недостатки*:

- Внешний вид Tkinter по умолчанию устарел.

- Мир Unix. Несмотря на то, что в нем отсутствуют такие тонкости, как анимированные виджеты, градиенты или масштабируемая графика, он, тем не менее, значительно улучшился за последние несколько лет благодаря обновлениям в самом Tk и добавлению тематических библиотек виджетов.

- В Tkinter отсутствуют более сложные виджеты: В Tkinter отсутствуют расширенные виджеты, такие как редакторы расширенного текста, встраивание 3D-графики, средства просмотра HTML или специализированные виджеты ввода.

- Tkinter может быть неправильным выбором для игрового пользовательского интерфейса или гладкого коммерческого приложения; однако для приложений, управляемых данными, простых утилит, диалоговых окон настройки и других приложений бизнес-логики Tkinter предлагает все необходимое и даже больше.

*Установка Tkinter.* Tkinter включен в стандартную библиотеку Python для дистрибутивов Windows и macOS. Итак, если вы установили Python на эти платформы с помощью официальных установщиков, вам не нужно ничего делать

для установки Tkinter.

## 3 Проектная часть

### 3.1 Архитектура системы взаимодействия

Разрабатываемый проект предназначен для открытого пользования в геологоразведочных работах. Вся реализация состоит из двух частей это программирование алгоритма кригинга и разработка ПИ с использованием этого алгоритма. Взаимодействие первой части проекта со второй происходит следующим образом, при нажатии на соответствующую кнопку происходит вызов функции, по которой производится расчёт по считанным входным данными. В число входных данных входит координаты точек и значениям в этих точках, а также тип вариограммы. Данные по координатам и значениям могут считаться с excel, json или с csv формата файла. А затем пользователь выбирает советующую вариограмму для вычисления и построения сетки значений, по которой строится сама вариограмма. Также в конце пользователь может выгрузить результаты работы кригинга в виде таблицы. Архитектура проекта отражена на рисунке 3.1.1.

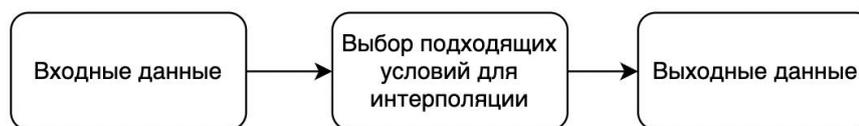


Рисунок-3.1.1 – Архитектура системы взаимодействия

### 3.2 Описание диаграммы деятельности

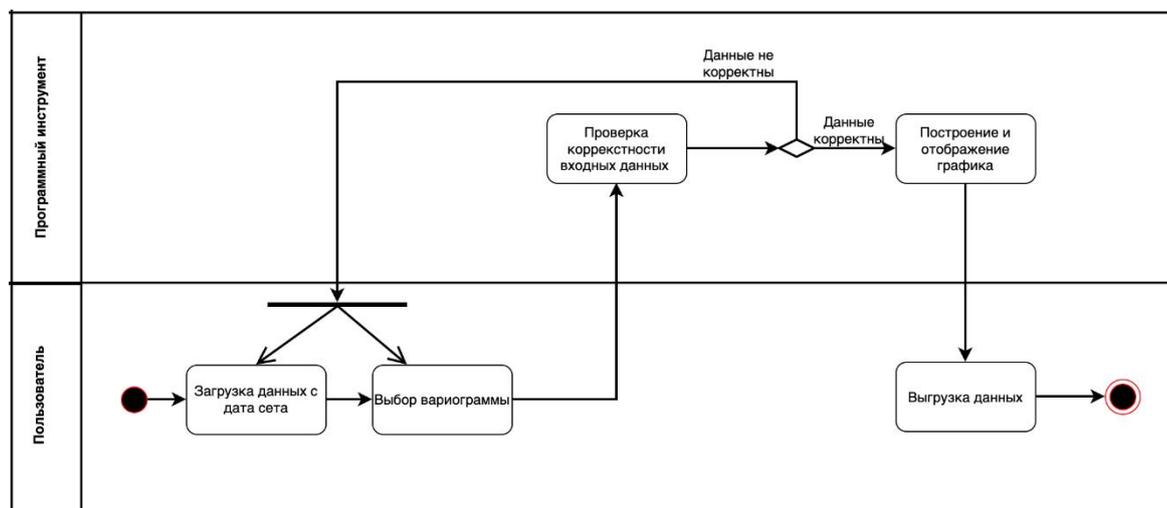
Диаграмма UML показывает унифицированное визуальное представление системы. UML диаграмма позволяет разработчикам понимать, анализировать и создавать структуру и поведение своей системы. Понимание и структурирование работы всего проекта является основой разработки какой-либо системы. Все это оказывает значительное влияние на качество разрабатываемого ПО. Любая новая функция, изменение или удаление старых функций необходимо учитывать в этих структурах, тем самым можно внедрить систему не мешая другим существующим частям системы. Это помогает поддерживать хорошо организованную общую структуру системы, особенно в гибких процессах разработки программного обеспечения.

Среди всех видов UML диаграмм, диаграмма позволяющая визуальное отобразить последовательность действий в системе, является диаграмма

деятельности. Диаграмма деятельности – это визуализированное отображение последовательности действий пользователя и ПО в проекте. Диаграмма представляет собой блок-схему, которая наглядно показывает, как проект должен работать, и также при этом учитывается все возможные условия работы. Ценность этой диаграммы раскрывается при проектировании ПИ так как можно учесть все исходы работы программы. Самая важная функция, которую выполняет эта диаграмма является ее простота в понимании работы ПИ.

Работа приложения начинается с его запуска. При запуске приложения будет отображаться меню и панель с кнопками. Чтобы начать работу расчета кригинга, пользователь должен выбрать файл для считывания данных. Файл должен содержать координаты  $x$ ,  $y$  и значения в этих точках. А затем после загрузки данных из файла пользователь должен выбрать один из видов построения вариограммы. И уже после этого пользователь может нажать кнопку для того, чтобы система начала расчёт с помощью метода кригинга и отобразила график. В случае некорректности входных, то есть, когда не выбрана диаграмма или ошибка в координатах входных данных, система оповестит об этом пользователя, соответственно следует проверять корректность данных перед расчетом. Далее с помощью соответствующей кнопки в меню пользователь может сохранить результат в виду excel формата. При закрытии приложение системы сама удалит сгенерированную фотографию и. Диаграмма деятельности отображена на Рисунке 3.2.1.

Для того что бы создать такую диаграмму в качестве примера использовалась готовые решения. Диаграмма была построена на веб инструменте по ссылке “diagrams.net”, который имеет огромное количество функции при созданиях различных диаграмм.



**Рисунок-3.2.1 – Диаграмма деятельности**

## 4 Экспериментальный раздел

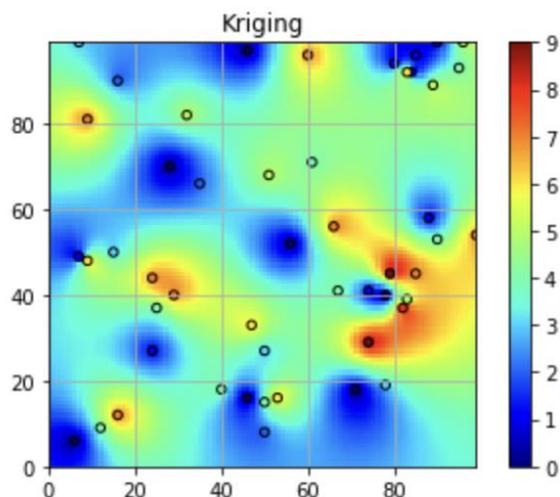
### 4.1 Разработка алгоритма кригинга

Для разработки системы работы кригинга нам потребуется сначала реализовать сам метод кригинга. Для этого специально создадим функцию расчета кригинга, назовем ее `Kriging`. Она будет принимать 5 параметров: массив координат  $x$ , координат  $y$ , массив значений в этих координатах, тип вариограммы и матрица всех точек для нахождения значений.

Как говорилось ранее метод кригинга состоит из 2 частей. Первый из которых это построение вариограммы. Для расчёта нам потребуется 2 главные переменные для построения вариограммы (`sill`, `range`). `Sill` — это так называемая дисперсия случайной величины, она находится по формуле (1.4). Значения параметра `range` берется как половина исследуемой области. Также для построения вариограммы следует вычислить матрицу расстояний между точками. Уже после этого для каждого расстояния между точками находим значение функции по специальной формуле вариограммы и заносим их в специальный массив ( $M$ ). В зависимости от типа вариограммы используется разные формулы (1.2, 1.3) вычисления построения вариограммы. Этот массив ( $M$ ), который получился, мы транспонируем в соответствии с формулой (1.5) и проставляем единицы и нули.

Следующим шагом станет вычисление вектора  $K$  из формулы (1.5) для каждой искомой точки. Для этого потребуется найти все расстояния между искомой точкой и всеми известными точками. И уже затем решить матричное уравнение для нахождения вектора весов для неизвестной точки. По общей формуле кригинга (1.1) находим значение в этой искомой точке. Начиная с нахождения вектора  $K$ , все остальное нужно повторить для каждой неизвестной точки и найти все значения в неизвестных точках.

В итоге функция `Kriging` должна будет вернуть матрицу значений в каждой точке. По этой матрице, используя библиотеку `Matplotlib`, строим график для полноты понимания работы кригинга. Пример работы визуализации результата кригинга можно посмотреть на рисунке 4.1.1.



**Рисунок-4.1.1 – График результата работы Кригинга**

## 4.2 Создание программного инструмента

После того как мы создали функцию вычисления вариограммы, следует разработать программный инструмент. ПИ будет реализоваться с помощью стандартной библиотеки Tkinter.

Tkinter — это кроссплатформенная событийно-ориентированная графическая библиотека, которая была написана Стином Лумхольтом и Гвидо ван Россумом. Пакет tkinter представляет собой стандартный интерфейс Python к инструментарию Tcl/Tk GUI toolkit. Как Tk, так и tkinter доступны на большинстве платформ Unix, включая macOS, а также в системах Windows. Python в сочетании с Tkinter обеспечивает быстрый и простой способ создания приложений с графическим интерфейсом.

Для начала работы с библиотекой Tkinter нужно импортировать библиотеку с помощью команды `import tkinter as tk`.

Для всего ПИ необходимо создать класс с методами и инициализацией окна со всеми пользовательскими интерфейсами. На рисунке 4.2.1 показано как инициализируется окно ПИ и назначается название и размер окна.

```
self.window = tk.Tk()
self.window.title("Kriging")
self.window.rowconfigure(0, minsize=300, weight=1)
self.window.columnconfigure(1, minsize=500, weight=1)
```

**Рисунок-4.2.1 – Инициализация окна ПИ**

Далее создадим рабочую область, которая делится на две части (см. Рисунок 4.2.2). Первая будет содержать информационные поля, кнопку и радио-

батоны для выбора вида вариограммы. Вторая будет содержать картинку. С помощью метода grid позиционируем две области.

```
# создаем рабочую область
self.frm_form = tk.Frame()
self.frm_form.grid(row=0, column=0, sticky="ns")
self.img_form = tk.Frame(bg='black')
self.img_form.grid(row=0, column=1, sticky="nsew")
```

### Рисунок-4.2.2 – Создание рабочей области

Далее инициализируем меню (см. Рисунок 4.2.3). Тут создаем несколько вкладок для меню: Open, Save, Exit. Первая из них позволяет выбрать различный формат считываемого файла. Вторая сохранить результат. Третья выйти из приложения. Для каждой кнопки меню создается отдельный обработчик события, то есть функция с соответствующим функционалом.

```
self.main_menu = tk.Menu()
self.open_menu = tk.Menu()
self.open_menu.add_command(label="Open excel dataset", command=self.open_excel_file)
self.open_menu.add_command(label="Open csv dataset", command=self.open_csv_file)
self.open_menu.add_command(label="Open json dataset", command=self.open_json_file)
self.save_menu = tk.Menu()
self.save_menu.add_command(label="Save result value points", command=self.save_value_points)
self.file_menu = tk.Menu()
self.file_menu.add_command(label="Exit", command=self.close_window)
self.main_menu.add_cascade(label="Open", menu=self.open_menu)
self.main_menu.add_cascade(label="Save", menu=self.save_menu)
self.main_menu.add_cascade(label="Exit", menu=self.file_menu)
self.window.config(menu=self.main_menu)
```

### Рисунок-4.2.3 – Создание меню ПИ

Далее располагаем все информационные строки, кнопку и радио-батоны на правой стороне окна ПИ. На рисунке 4.2.4 отображается конечный интерфейс ПИ.

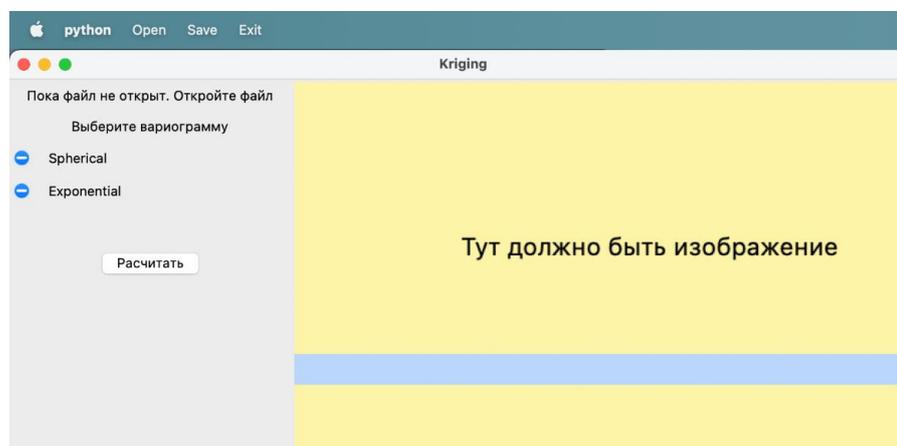
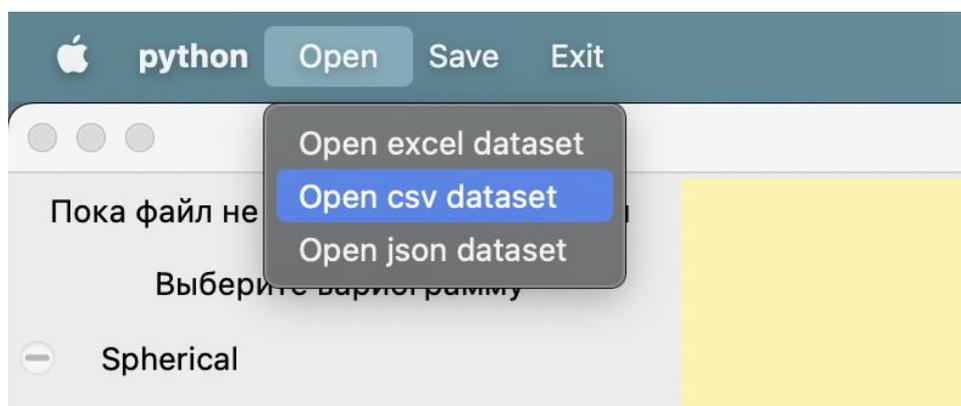


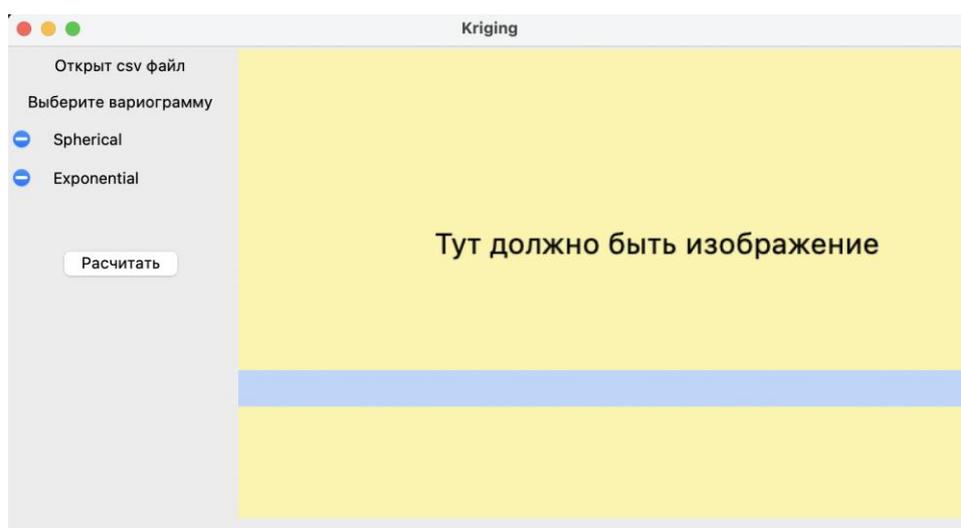
Рисунок-4.2.4 – Интерфейс программного инструмента

С помощью строки *self.window.mainloop()* запускаем приложение.

Рассмотрим работу функции открытия и считывание данных с файла. Изначально пользователю будет отображаться надпись о том, что данные не считаны и нужно открыть файл. Соответственно, нужно с помощью меню (см. Рисунок 4.2.5) и выбрав соответствующую кнопку открыть файл. После этого приложение отобразит открытие соответствующего файла (см. Рисунок 4.2.6).



**Рисунок-4.2.5 – Меню “Open”**



**Рисунок-4.2.6 – Работа информационного поля при открытии файла**

Здесь вызывается соответствующая функция для открытия файла (см. Рисунок 4.2.7). В ней используется специальная библиотека tkinter для открытия файлового окна. Данную библиотеку можно импортировать с помощью следующей команды: `“from tkinter.filedialog import askopenfilename, asksaveasfilename”`.

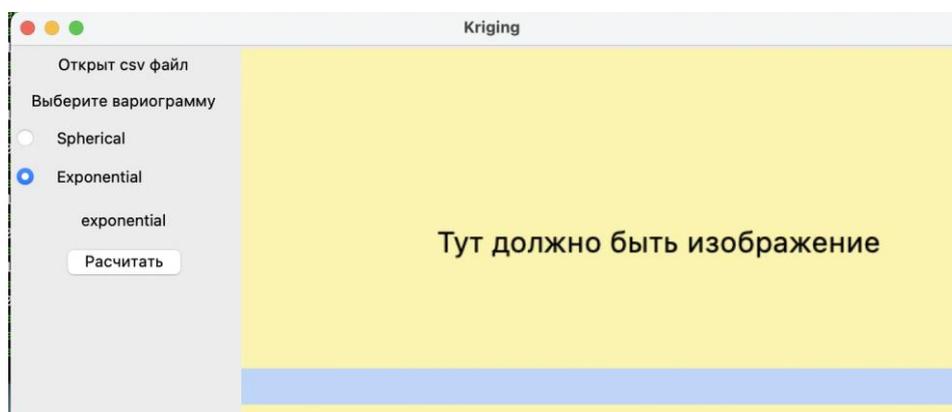
```

def open_csv_file(self):
    """Открываем файл для считывания данных с csv"""
    filepath = askopenfilename(
        filetypes=[("Csv файлы", "*.csv"), ("Все файлы", "*.*")]
    )
    if not filepath:
        return
    self.read_file = pd.read_csv(filepath)
    print(self.read_file)
    self.lbl_info_file.config(text="Открыт csv файл")

```

**Рисунок-4.2.7 – Реализация открытия excel файла**

После открытия файла пользователь должен выбрать способ построения вариограммы. После выбора ниже отобразится выбранный способ (см. Рисунок 4.2.8).



**Рисунок-4.2.8 – Выбор вариограммы**

Далее пользователю следует нажать на кнопку для запуска расчета. При нажатии на эту кнопку вызывается функция *select\_value()*, которая передает все входные данные в функцию *calculate()*. Эта функция преобразовывает эти данные в соответствующий вид, передает их в функцию *Kriging*, а также с помощью библиотеки *matplotlib* строит график и сохраняет его, при этом эта функция возвращает результат после расчета методом кригинга (см. Рисунок 4.2.9).

```

def calculate(rf, var):
    x = rf['x']
    y = rf['y']
    v = rf['v']
    grid = np.zeros((max(x), max(y)))

    grid = Kriging(x,y,v,var,grid)
    plt.imshow(grid.T,origin='lower',interpolation='nearest',cmap='jet')
    plt.scatter(x,y,c=v,cmap='jet',s=20, edgecolors='black')
    plt.xlim(0,grid.shape[0])
    plt.ylim(0,grid.shape[1])
    plt.title("Kriging")
    plt.colorbar()
    plt.grid()
    plt.savefig('img.jpg')
    plt.show()

    return grid

```

### Рисунок-4.2.9 – Реализация функции calculate()

Функция *select\_value()* принимает результат вычисления и сохраняет для дальнейших действий (см. Рисунок 4.2.10).

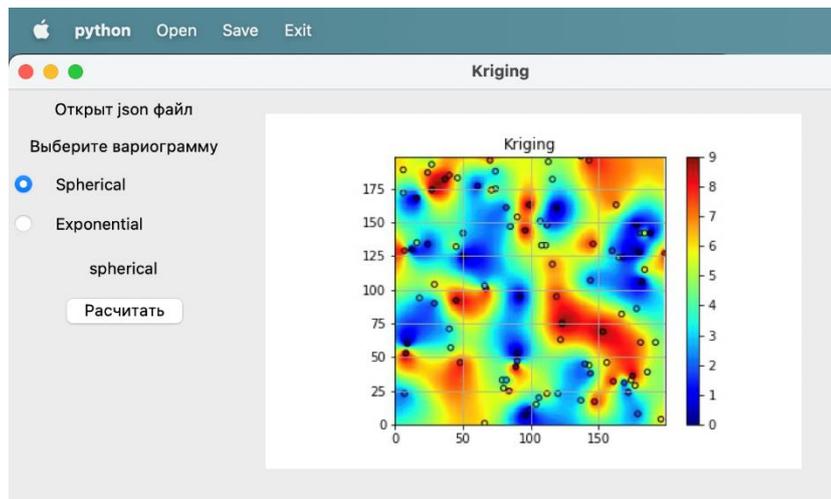
```

def select_value(self):
    try:
        self.variogram = str(self.var.get())
        print(self.variogram)
        self.data_output = calculate(self.read_file, self.variogram)
    try:
        self.image = Image.open("img.jpg")
        self.photo = ImageTk.PhotoImage(self.image)
        self.image = self.canvas.create_image(20, 20, anchor='nw', image=self.photo)
        self.canvas.config()
    except:
        mb.showwarning("Предупреждение", "Отсутствует фото для отображения")
        print('Not exists img')
    except:
        mb.showwarning("Предупреждение", "Неправильные входные данные")
        print('Неправильные входные данные')

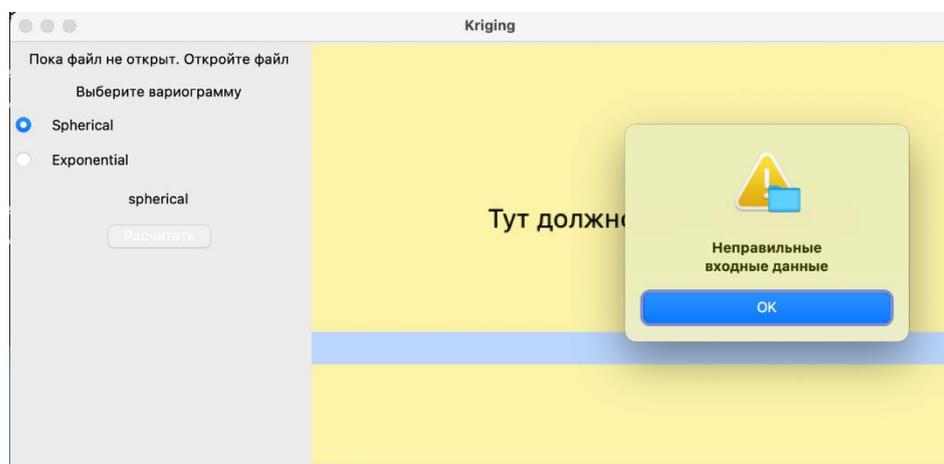
```

### Рисунок-4.2.10 – Реализация метода обработки нажатия на кнопку

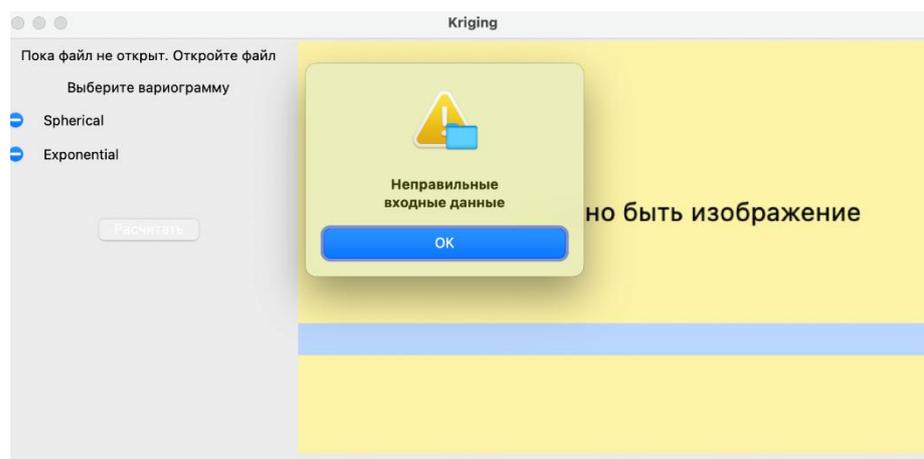
Также этот метод учитывает некорректность входных данных. Если все входные данные верны, то он отобразит сохранённую картинку в соответствии с рисунком 4.2.11. Если же нет, то пользователю выведется уведомление о некорректности данных в соответствии с рисунками 4.2.12, 4.2.13.



**Рисунок-4.2.11 – Правильная работа ПИ**



**Рисунок-4.2.12 – Ошибка при не открытии файла**



**Рисунок-4.2.13 – Ошибка, когда не открыт файл и не выбрана вариограмма**

После выполнения расчёта методом кригинга пользователь по желанию

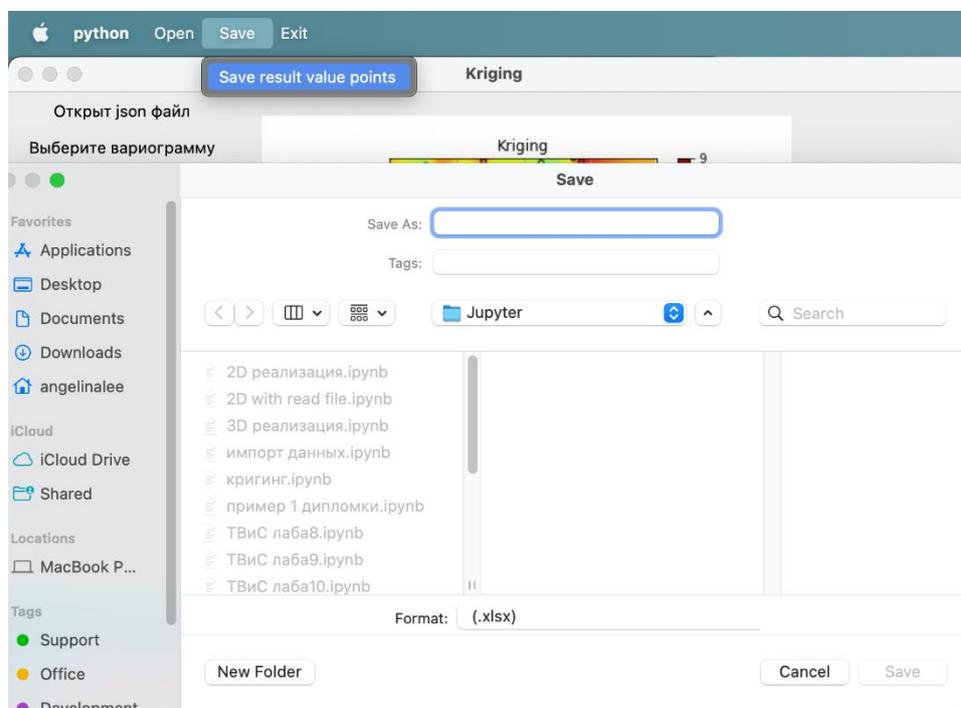
может выгрузить результаты работы кригинга. Для этого в меню есть вкладка “Save”, при нажатии по которой происходит вызов функции `save_value_points()`, реализация которой см на рисунке 4.2.14. Тут также как и со считыванием файла используется специальная библиотека для работы с файловой системой. В этой функции происходит преобразование матрицы в объект `DataFrame`, чтобы сохранить данные в excel формате.

```
def save_value_points(self):
    """Сохраняем текущий файл как новый файл."""
    filepath = asksaveasfilename(
        defaultextension="xlsx",
        filetypes=[("", "*.xlsx"), ("Все файлы", "*.*")],
    )
    if not filepath:
        return
    self.data_output
    self.do = pd.DataFrame({'x':[], 'y':[], 'v':[]})
    for i in range(self.data_output.shape[0]):
        for j in range(self.data_output.shape[0]):
            self.new_row = {'x':i, 'y':j, 'v':self.data_output[i][j]}
            self.do = self.do.append(self.new_row, ignore_index=True)
    self.do.to_excel('result.xlsx')
    self.data_output.to_csv('data_output_points.csv')
    df.to_json('data_points.json')

    print(self.data_output)
    print(self.do)

    self.lbl_info_file.config(text="Файл сохранен")
```

**Рисунок-4.2.14 – Реализация метода сохранения данных результата кригинга**

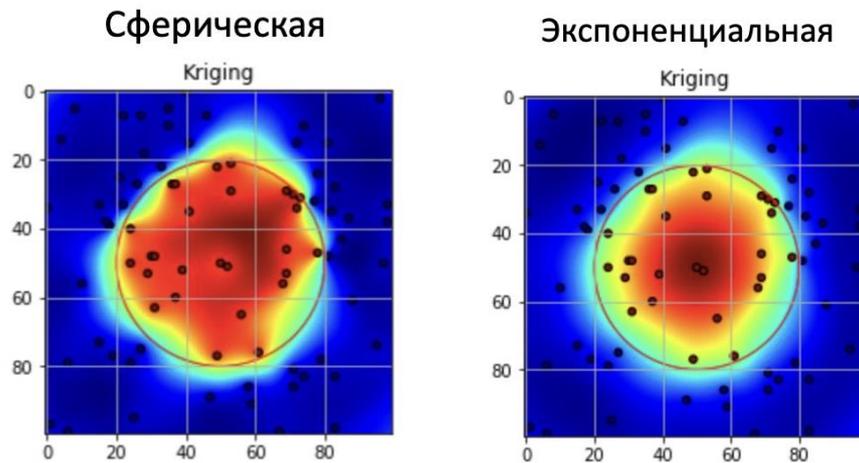


**Рисунок-4.2.15 – Сохранение результата в файле excel**

### 4.3 Верификация метода

После произведения расчётов и визуализации стоит рассмотреть точность метода на основе примера.

Для верификации в качестве примера была взята формула окружности, с радиусом 20 и центральной точкой ( $x = 50, y = 50$ ). Каждой случайно сгенерированной точке в пределах этой окружности было присвоено значение наибольшей концентрации содержания руды, а точки, которые располагаются вне окружности присвоено наименьшее значение. На рисунке 4.3.1 отображено видимое отклонение результата кригинга от прогнозируемой окружности. Можно заметить то, что сферическая модель вариограммы является более точной по сравнению с экспоненциальной модели для данного примера, так как ее красная площадь охватывает большую территорию окружности чем экспоненциальная.



**Рисунок-4.3.1 – Визуализация отклонения результата кригинга с использованием разных моделей вариограмм**

## ЗАКЛЮЧЕНИЕ

Результатом дипломного проекта является разработка программного инструмента для интерполяции скважинных данных на основе геостатистического метода кригинга.

В исследовательском разделе в процессе изучения предметной области была рассмотрена актуальность системы и ее обширные области применения. В этом же разделе были выделены преимущества метода над всеми остальными.

В технологическом разделе были выявлены главные преимущества выбранного языка программирования и среды разработки для использования в данном проекте.

В проектной части было продемонстрирована последовательность действий системы, их процесс работы в различных случаях. В данном разделе рассмотрена общая архитектура взаимодействия системы, по которой производилась разработка системы.

В экспериментальном разделе рассмотрены все функции системы и их реализация в проекте.

Чтобы сделать процесс разведки месторождения более рентабельным и менее трудозатратным, нужно интерполировать данные на исследуемой территории с использованием некоторого геостатистического метода для заполнения пробелов между скважинами. Таким методом наиболее подходящим является метод кригинг с использованием различных видов вариограмм.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 Добыча урана в Казахстане [Электронный ресурс] – Режим доступа: <https://mobi-scripts.ru/stati/493-dobycha-urana-v-kazahstane.html>, свободный.
- 2 Демьянов В. В., Савельева Е. А. Геостатистика: теория и практика под ред. Арутюняна Р. В. — М.: Наука, 2010. — 327 с.
- 3 Kriging Interpolation – The Prediction Is Strong in this One [Электронный ресурс] – Режим доступа: <https://gisgeography.com/kriging-interpolation-prediction>, свободный.
- 4 Jupyter Project Documentation [Электронный ресурс] – Режим доступа: <https://docs.jupyter.org/en/latest>, свободный.
- 5 World Nuclear Association, Uranium Mining Overview, [Электронный ресурс] – Режим доступа: <http://www.world-nuclear.org/info/Nuclear-Fuel-Cycle/Mining-of-Uranium/Uranium-Mining-Overview/>, свободный.
- 6 Справка ArcGIS [Электронный ресурс] – Режим доступа: <https://resources.arcgis.com/ru/help/main/10.1/index.html>, свободный.
- 7 Mitas L., Mitasova H. “Spatial interpolation” In Geographical Information Systems: Principles, Techniques, Management and Applications, под ред: Longley, P. A., Goodchild, M. F., Maguire, D. J., Rhind, D. W., - New York: Wiley, 1999. - 492 с.
- 8 World Nuclear Association, Uranium and Nuclear Power in Kazakhstan [Электронный ресурс] – Режим доступа: <http://www.world-nuclear.org/info/Country-Profiles/Countries-G-N/Kazakhstan/2015>, свободный.
- 9 Jean-Laurent Mallet, Geomodeling, Oxford University Press, New York, 2002
- 10 Alan D. M. Python GUI Programming with Tkinter - UK: Packt Publishing Ltd. 2021. - 860 с
- 11 Любанович Б. Простой Python. Современный стиль программирования. 2-е изд. — СПб.: Питер, 2021. — 592 с.
- 12 Документация tkinter [Электронный ресурс] – Режим доступа: <https://docs.python.org/3/library/tkinter.html>, свободный.
- 13 Как работает инструмент Кригинг (Kriging) [Электронный ресурс] – Режим доступа: <https://desktop.arcgis.com/ru/arcmap/10.3/tools/3d-analyst-toolbox/how-kriging-works.htm>, свободный.
- 14 Кригинг [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B8%D0%B3%D0%B8%D0%BD%D0%B3>, свободный.
- 15 Математические методы и автоматизированные системы в геологии. Практические методы крайгинга [Электронный ресурс] – Режим доступа: <https://www.geokniga.org/labels/870>, свободный.
- 16 Кригинг и базовые модели геостатистики / Демьянов В.В., Каневский М.Ф., Савельева Е.А. и др. - М: ВИНТИ, 1999. - 13 с.
- 17 Геостатистика [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%BE%D1%81%D1%82%D0%B0>

%D1%82%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B0 свободный.

18 Мальцев К.А., Мухарамова С.С. Построение моделей пространственных переменных (с применением пакета Surfer): Учебное пособие – Казань: Казанский университет, 2014. – 103 с.

19 Девис Дж. Статистика и анализ геологических данных - М.:Мир, 1997. - 572 с.

20 Жучкова В.К., Раковская Э.М. Методы комплексных физико-географических исследований. Учебное пособие для студентов вузов - М:Мир , 2004. - 368 с.

## **Приложение А**

### **(обязательное)**

#### Техническое задание

#### **А.1.5 Техническое задание на разработку программного инструмента для интерполяции скважинных данных на основе геостатистического метода кригинга**

Настоящее техническое задание распространяется на разработку с программного инструмента для интерполяции скважинных данных на основе геостатистического метода кригинга. Предполагается, что использовать данную систему будут для проведения геологоразведочных работ для добычи руд на месторождение. Этот инструмент поможет улучшить качество получения информации о межскважинном пространстве и поможет уменьшить стоимость разделочных работ.

##### **А.1.5.1 Основание для разработки**

Система разрабатывается на основании устного распоряжения дипломного руководителя.

##### **А.1.5.2 Назначение**

Система предназначена для вычисления данных о межскважинном пространстве с использованием геостатистического метода кригинга тем самым минимизируя расходы и трудоёмкость процесса на стадии разведки месторождения.

##### **А.1.5.3 Требования к функциональным характеристикам**

Программный инструмент должно обеспечивать возможность выполнения следующих функций:

- должен принимать json, csv, excel формат входных данных;
- выбор типа вариограммы;
- визуализирование графика результата работы кригинга;

## **Продолжение приложения А**

– получение результатов и их возможность выгрузить в отдельный файл.

### **А.1.5.4 Требования к надежности**

Предусмотреть проверку вводимой информации. Предусмотреть некорректные действия со стороны пользователя при работе с системой. Обеспечить целостность хранимой информации.

## Приложение Б (обязательное)

### Текст программы

```
# загрузка библиотек
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
import copy
from sklearn import preprocessing
from __future__ import division
import numpy as np
import matplotlib.pyplot as plt
import scipy.linalg as LA

%matplotlib inline

def spherical( h, a, sill ):
    """
    Spherical model of the semivariogram
    """
    if h <= a:
        return sill*( 1.5*h/a - 0.5*(h/a)**3.0 )
    else:
        return sill
def exponential( h, a, sill ):
    """
    Exponential model of the semivariogram
    """
    if h == 0:
        return 0
    else:
        return sill*( 1 - np.e**(-3*h/a))
```

## Продолжение приложения Б

```
def Kriging(x,y,v,variogram,grid):
    cov_distancias = np.zeros((x.shape[0],x.shape[0]))
    K = np.zeros((x.shape[0]+1,x.shape[0]+1))
    sill = np.average(np.power(v - np.average(v), 2)) # v[:].var()
    a_range = grid.shape[0]/2
    # вычисление матрицы расстояний между точками
    for i in range(x.shape[0]-1):
        cov_distancias[i,i:]=np.sqrt((x[i:]-x[i])**2+(y[i:]-y[i])**2)

    for i in range(x.shape[0]):
        for j in range(x.shape[0]):
            if cov_distancias[i,j]!=0:
                if(variogram=='spherical'):
                    K[i,j]=spherical(cov_distancias[i,j], a_range, sill)
                elif(variogram=='exponential'):
                    K[i,j]=exponential(cov_distancias[i,j], a_range, sill)

    K = K + K.T
    K[-1,:]= 1
    K[:,-1]= 1
    K[-1,-1]= 0

    for i in range(grid.shape[0]):
        for j in range(grid.shape[1]):
            distancias = np.sqrt((i-x[:])**2+(j-y[:])**2)
            M = np.ones((x.shape[0]+1,1))
            for k in range(M.shape[0]-1):
                if(variogram=='spherical'):
                    M[k,0]=spherical(distancias[k], a_range, sill)
                elif(variogram=='exponential'):
                    M[k,0]=exponential(distancias[k], a_range, sill)
            W = LA.solve(K,M)
            grid[i,j] = np.sum(W[:-1,0]*(v[:]))

    return grid
```

## Продолжение приложения Б

```
def calculate(rf, var):
    x = rf['x']
    y = rf['y']
    v = rf['v']
    grid = np.zeros((max(x), max(y)))

    grid = Kriging(x,y,v,var,grid)
    plt.imshow(grid.T,origin='lower',interpolation='nearest',cmap='jet')
    plt.scatter(x,y,c=v,cmap='jet',s=20, edgecolors='black')
    plt.xlim(0,grid.shape[0])
    plt.ylim(0,grid.shape[1])
    plt.title("Kriging")
    plt.colorbar()
    plt.grid()
    plt.savefig('img.jpg')
    plt.show()

    return grid

import tkinter as tk
import pandas as pd
import os
from PIL import ImageTk,Image
from tkinter.filedialog import askopenfilename, asksaveasfilename
import tkinter.messagebox as mb

class App:
    def __init__(self):
        self.window = tk.Tk()
        self.window.title("Kriging")
        self.window.rowconfigure(0, minsize=300, weight=1)
        self.window.columnconfigure(1, minsize=500, weight=1)
        # создаем рабочую область
        self.frm_form = tk.Frame()
        self.frm_form.grid(row=0, column=0, sticky="ns")
        self.img_form = tk.Frame(bg='black')
        self.img_form.grid(row=0, column=1, sticky="nsew")

        self.main_menu = tk.Menu()
        self.open_menu = tk.Menu()
        self.open_menu.add_command(label="Open excel dataset",
        command=self.open_excel_file)
```

## Продолжение приложения Б

```
self.open_menu.add_command(label="Open csv dataset",
command=self.open_csv_file)
self.open_menu.add_command(label="Open json dataset",
command=self.open_json_file)
self.save_menu = tk.Menu()
self.save_menu.add_command(label="Save result value points",
command=self.save_value_points)
self.file_menu = tk.Menu()
self.file_menu.add_command(label="Exit", command=self.close_window)
self.main_menu.add_cascade(label="Open", menu=self.open_menu)
self.main_menu.add_cascade(label="Save", menu=self.save_menu)
self.main_menu.add_cascade(label="Exit", menu=self.file_menu)
self.window.config(menu=self.main_menu)

# Добавим метку
self.lbl_info_file = tk.Label(self.frm_form, text="Пока файл не открыт.
Откройте файл", padx=15, pady=5)
self.lbl_info_file.grid(row=2, column=0, sticky='n')
# Добавим метку
self.lbl_header = tk.Label(self.frm_form, text="Выберите вариограмму",
padx=15, pady=5)
self.lbl_header.grid(row=6, column=0, sticky='n')
self.var = tk.StringVar()
self.chb_spher = tk.Radiobutton(self.frm_form, text='Spherical',
value='spherical', variable=self.var, padx=15, pady=5)
self.chb_spher.grid(row=7, column=0, sticky='w')
self.chb_expl = tk.Radiobutton(self.frm_form, text='Exponential',
value='exponential', variable=self.var, padx=15, pady=5)
self.chb_expl.grid(row=8, column=0, sticky='w')
# Добавим метку
self.selection = tk.Label(self.frm_form, textvariable=self.var, padx=15,
pady=10)
self.selection.grid(row=10, column=0, sticky='n')
# вставляем кнопку
self.btn_calc = tk.Button(self.frm_form, text="Расчитать",
command=self.select_value)
self.btn_calc.grid(row=11, column=0, sticky='n')
self.image = Image.open("default.png")
self.photo = ImageTk.PhotoImage(self.image)
# Добавим изображение
self.canvas = tk.Canvas(self.img_form, width = 600, height = 400)
self.image = self.canvas.create_image(0, 0, anchor='nw', image=self.photo)
```

## Продолжение приложения Б

```
self.canvas.pack()

# Запуск приложения.
self.window.mainloop()

def select_value(self):
    try:
        self.variogram = str(self.var.get())
        print(self.variogram)
        self.data_output = calculate(self.read_file, self.variogram)
    try:
        self.image = Image.open("img.jpg")
        self.photo = ImageTk.PhotoImage(self.image)
        self.image = self.canvas.create_image(20, 20, anchor='nw',
image=self.photo)
        self.canvas.config()
    except:
        mb.showwarning("Предупреждение", "Отсутствует фото для
отображения")
        print('Not exists img')
    except:
        mb.showwarning("Предупреждение", "Неправильные входные данные")
        print('Неправильные входные данные')

def close_window(self):
    # self.window.quit()
    self.window.destroy()

def open_excel_file(self):
    """Открываем файл для считывания данных с excel"""
    filepath = askopenfilename(
        filetypes=[("Excel файлы", "*.xlsx"), ("Все файлы", "*.*")]
    )
    if not filepath:
        return
    self.read_file = pd.read_excel(filepath)
    print(self.read_file)
    self.lbl_info_file.config(text="Открыт excel файл")

def open_csv_file(self):
    """Открываем файл для считывания данных с csv"""
    filepath = askopenfilename(
```

## Продолжение приложения Б

```
filetypes=[("Csv файлы", "*.csv"), ("Все файлы", "*.*")]
)
if not filepath:
    return
self.read_file = pd.read_csv(filepath)
print(self.read_file)
self.lbl_info_file.config(text="Открыт csv файл")

def open_json_file(self):
    """Открываем файл для считывания данных с json"""
    filepath = askopenfilename(
        filetypes=[("Json файлы", "*.json"), ("Все файлы", "*.*")]
    )
    if not filepath:
        return
    self.read_file = pd.read_json(filepath)
    print(self.read_file)
    self.lbl_info_file.config(text="Открыт json файл")

def save_value_points(self):
    """Сохраняем текущий файл как новый файл."""
    filepath = asksaveasfilename(
        defaultextension="xlsx",
        filetypes=[("", "*.xlsx"), ("Все файлы", "*.*")],
    )
    if not filepath:
        return
    self.data_output
    self.do = pd.DataFrame({'x':[], 'y':[], 'v':[]})
    for i in range(self.data_output.shape[0]):
        for j in range(self.data_output.shape[0]):
            self.new_row = {'x':i, 'y':j, 'v':self.data_output[i][j]}
            self.do = self.do.append(self.new_row, ignore_index=True)
    self.do.to_excel('result.xlsx')
# self.data_output.to_csv('data_output_points.csv')
# df.to_json('data_points.json')

print(self.data_output)
print(self.do)

self.lbl_info_file.config(text="Файл сохранен")
```

## Продолжение приложения Б

```
app = App()
os.remove("img.jpg")
print("File removed successfully")
```