

Автономная некоммерческая
образовательная организация высшего образования
«НАУЧНО-ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «СИРИУС»

Научный центр информационных технологий и искусственного интеллекта
направление «Математическое моделирование в биомедицине и геофизике»

К ЗАЩИТЕ ДОПУСТИТЬ
Руководитель направления
«Математическое моделирование в
биомедицине и геофизике»,
к.т.н., д.э.н. М.В. Ширяев
« ____ » _____ 2025 г.

**«ВОССТАНОВЛЕНИЕ ПАРАМЕТРОВ РАЗРЫВНЫХ НАРУШЕНИЙ
ПЛАСТА»**

Магистерская диссертация
по направлению подготовки 01.04.02 «Прикладная математика и
информатика»,
направленность (профиль) «Математическое моделирование процессов и
материалов»

Студент гр. М01ММ-23
____ Ю.Н. Терентьева
« ____ » _____ 2025 г.

Руководитель
профессор направления
«Математическое моделирование в
биомедицине и геофизике»,
к.ф.-м.н., PhD
____ С.Ю. Малясов
« ____ » _____ 2025 г.

Автономная некоммерческая
образовательная организация высшего образования
«НАУЧНО-ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «СИРИУС»

Научный центр информационных технологий и искусственного интеллекта
направление «Математическое моделирование в биомедицине и геофизике»

УТВЕРДИТЬ

Руководитель направления
«Математическое моделирование в
биомедицине и геофизике»,
к.т.н., д.э.н. М.В. Ширяев
«___» _____ 2025 г.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на выполнение магистерской диссертации
по направлению подготовки 01.04.02 «Прикладная математика и
информатика»,
направленность (профиль) «Математическое моделирование процессов и
материалов»

Терентьева Юлия Николаевна

1. Тема: «Восстановление параметров разрывных нарушений пласта».
2. Цель: программная реализация и исследование алгоритма восстановления геологических поверхностей с учетом разрывных нарушений на основе метода дискретной гладкой интерполяции (DSI).
3. Задачи:
 - изучение классификации разрывных нарушений и их геометрических особенностей;
 - обзор и сравнительный анализ методов интерполяции;
 - программная реализация математической модели с использованием метода DSI;
 - генерация синтетических геологических данных, тестирование для разных видов разломов;

– анализ результатов.

4. Рабочий график (план) выполнения магистерской диссертации:

№	Перечень заданий	Сроки выполнения
1.	Написание литературного обзора	18.02.2025 – 15.03.2025
2.	Изложение материалов и методов	15.03.2025 – 20.04.2025
3.	Представление полученных результатов	20.04.2025 – 20.05.2025

Дата выдачи: «18» февраля 2025 г.

Руководитель:

_____ С.Ю. Малясов

Задание принял к исполнению:

Студент группы М01ММ–23

_____ Ю.Н. Терентьева

«18» февраля 2025 г.

Реферат

Выпускная квалификационная работа, 78 страниц, 23 рисунка, 1 таблица, 28 источников, 1 приложение.

ПОВЕРХНОСТЬ, РАЗРЫВНОЕ НАРУШЕНИЕ, МЕТОД ИНТЕРПОЛЯЦИИ, ИЗОЛИНИЯ, ДИСКРЕТНАЯ ГЛАДКАЯ ИНТЕРПОЛЯЦИЯ.

Объектом исследования является восстановление геологических поверхностей с учетом разрывных нарушений на основе метода дискретной гладкой интерполяции (DSI).

Цель работы: разработка и программная реализация алгоритма восстановления геологических поверхностей с разломами, а также анализ качества интерполяции на синтетических данных.

В процессе работы проведён обзор существующих методов интерполяции, изучены особенности разрывных нарушений, реализован метод DSI, выполнено тестирование на различных типах разломов и проведён анализ полученных результатов.

В результате исследования разработана программная реализация метода DSI, позволяющая эффективно моделировать геологические поверхности с сохранением топологии разломов и высокой точностью восстановления. Метод показал устойчивость и высокую точность интерполяции, что подтверждается визуализацией и количественным анализом ошибок.

Результаты работы могут быть интегрированы в геологические системы моделирования и использоваться для повышения точности оценки запасов и планирования разработки месторождений.

Эффективность метода определяется его способностью учитывать сложные разрывные структуры и обеспечивать баланс между точностью и гладкостью интерполированной поверхности. Метод способствует снижению экономических рисков при бурении и разработке нефтегазовых месторождений.

The abstract

Graduation qualification work, 78 pages, 23 figures, 1 table, 28 sources, 1 appendix.

SURFACE, FAULT, INTERPOLATION METHOD, ISOLINE, DISCRETE SMOOTH INTERPOLATION.

The object of study is the restoration of geological surfaces taking into account discontinuous faults based on the discrete smooth interpolation (DSI) method.

The purpose of the work: development of an algorithm for restoring geological surfaces with faults, as well as an analysis of the quality of interpolation on synthetic data.

In the course of the work, a review of existing interpolation methods was conducted, the features of discontinuous faults were studied, the DSI method was implemented, testing was performed on various types of faults and an analysis of the obtained results was carried out.

As a result of the study, a software implementation of the DSI method was developed, allowing for efficient modeling of geological surfaces while preserving the topology of faults and high accuracy of restoration. The method showed stability and high accuracy of interpolation, which is confirmed by visualization and quantitative analysis of errors.

The results of the work can be integrated into geological modeling systems and used to improve the accuracy of reserves assessment and field development planning. The effectiveness of the method is determined by its ability to take into account complex discontinuous structures and to ensure a balance between the accuracy and smoothness of the interpolated surface. The method helps to reduce economic risks in drilling and developing oil and gas fields.

Сокращения, обозначения, термины и определения

В настоящей работе применяют следующие термины с соответствующими определениями:

- разлом – зона тектонического смещения, характеризующаяся нарушением сплошности горных пород и резким изменением геологических параметров;
- дискретная гладкая интерполяция (DSI) – метод интерполяции, учитывающий наличие разрывных нарушений, позволяющий строить гладкие поверхности с сохранением топологии разломов;
- изолинии – линии на карте или графике, соединяющие точки с одинаковыми значениями исследуемого параметра;
- интерполяция – процесс восстановления значений функции в промежуточных точках на основе известных дискретных данных.

Оглавление

Введение	9
1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ВОССТАНОВЛЕНИЯ ПОВЕРХНОСТИ С РАЗРЫВАМИ.....	13
1.1 Виды разрывных нарушений в геологии	13
1.2 Геометрическая интерпретация разрывных нарушений.....	20
1.3 Методы интерполяции поверхности.....	22
1.3.1 Линейная интерполяция.....	22
1.3.2 Интерполяция сплайнами	23
1.3.3 Метод обратных расстояний (IDW)	24
1.3.4 Кригинг	24
1.3.5 Радиальные базисные функции (RBF)	25
1.3.6 Метод интерполяции по принципу минимальной кривизны.....	26
1.3.7 Метод интерполяции поверхности с известными разрывами и штрафом за пересечение разрыва.....	27
1.3.8 Метод Монте-Карло	29
1.3.9 Метод с использованием уравнения Навье-Стокса с обратным временем и отрицательным временным шагом адвекции	31
1.3.10 Трёхмерный дип-доменный подход	33
1.4 Метод дискретной гладкой интерполяции (DSI)	35
2 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ И ЭКСПЕРИМЕНТЫ	43
1.1 Программная реализация метода DSI.....	43
2.2 Результаты моделирования	49
Заключение	59

Список использованных источников	61
Приложение А (обязательное) Программная реализация задачи интерполяции методом DSI	64

Введение

Геологическое моделирование пласта играет важную роль в нефтяной сфере. При разработке месторождения первым этапом изучения является построение модели пласта. Строятся эти модели по данным разведки со скважин, чаще всего эти данные очень разреженные, возникает сложность в восстановлении исходной поверхности. Кроме этого, на пласт могут влиять различные деформации и разрывные нарушения [5,9,19].

Разрывные нарушения представляют большую проблему в моделировании пласта. Разломы – это зоны тектонических смещений, в которых происходит нарушение сплошности пород [1, 17].

Неправильный учет разломов приводит к существенным ошибкам в оценке запасов углеводородов. Неточная модель пласта может повысить риск пробурить низкопродуктивную или сухую скважину, а это экономически невыгодно [3].

Сложной и трудоемкой задачей является разметка данных сейсморазведки вручную. Геологам нужно правильно очертить зоны разрывных нарушений на геологических картах и разрезах. Но процесс может значительно усложниться за счет того, что разломы могут иметь нетривиальную форму, неоднородность, нерегулярную структуру [2, 7].

Визуальное определение разломов субъективно и может повлечь ошибки, особенно при наличии большого количества неоднородных разреженных данных. При разметке данных важен опыт, глубокие профессиональные знания и постоянные сверки с другими специалистами. В итоге такая работа может занять продолжительное время, недели или даже месяцы.

В связи с этим растет необходимость в автоматизированных методах моделирования разломов, которые смогут существенно сократить время на исследование пласта [3,6].

Построение модели пласта значительно усложняется за счет присутствия разломов, потому что становится невозможным применение классических методов интерполяции, которые подразумевают непрерывность поверхности. Однозначное соединение точек не всегда возможно, особенно если эти точки находятся по разные стороны от линии разлома. Также сама структура разломов разнообразна от простых вертикальных смещений до сложных зигзагообразных разрывов. Для того чтобы учитывать эти особенности нужна адаптация или разработка специализированных методов восстановления поверхности.

Классические методы интерполяции, такие как кригинг и сплайны часто дают искажения, когда в поверхности присутствует разлом. Для получения достоверных результатов нужно учитывать наличие зон, в которых значения параметров могут иметь скачок. Особенно это актуально для месторождений со сложными системами разломов, таких как Западная Сибирь, Волго-Уральская провинция и Тимано-Печорская область [2,15].

В данной работе рассматривается задача интерполяции поверхности по заданным фиксированным точкам, с учетом разломов разных типов. Для решения используется метод дискретной гладкой интерполяции (DSI), так как с помощью него можно корректно обрабатывать разрывы и строить непрерывные участки между фиксированными точками.

Актуальность работы обусловлена тем, что ранее в нефтегазовой сфере многие компании использовали, в основном, зарубежные программы для моделирования пласта, но сейчас из-за санкций доступ ко многим ограничен. Это обстоятельство создает необходимость разработки собственных методов, способных эффективно восстанавливать геологические поверхности с учетом разрывных нарушений.

Практическая значимость работы заключается в повышении точности интерполяции поверхности с разломами, что в свою очередь поможет избежать рисков пробуривания сухих или малоэффективных скважин, в

возможности интеграции алгоритма в уже существующие модели, сокращении времени и сил на обработку геологических данных.

Цель исследования: программная реализация и исследование алгоритма восстановления геологических поверхностей с учетом разрывных нарушений на основе метода дискретной гладкой интерполяции (DSI).

В рамках поставленной цели можно выделить следующие задачи:

- изучение классификации разрывных нарушений и их геометрических особенностей;
- обзор и сравнительный анализ методов интерполяции;
- программная реализация математической модели с использованием метода DSI;
- генерация синтетических геологических данных, тестирование для разных видов разломов;
- анализ результатов.

Объект исследования – геологическая поверхность с разрывными нарушениями, описываемая данными по скважинам.

Предмет исследования – метод дискретной гладкой интерполяции для восстановления поверхности с учетом разрывов.

В работе применяются следующие методы исследования:

- анализ научной литературы по теме интерполяции и моделирования поверхности с разломом;
- программная реализация метода дискретной гладкой интерполяции;
- экспериментальное тестирование алгоритма на синтетических данных;
- сравнительный анализ методов интерполяции;
- обработка и интерпретация полученных результатов.

В структурном соотношении работа состоит из двух глав, в первой 4 подпункта, во второй – 2, введения, заключения, списка использованных источников. В первой главе рассматриваются теоретические составляющие:

виды разрывных нарушений с геологической стороны, геометрическая интерпретация, обзор и сравнение существующих методов интерполяции, подробный разбор метода гладкой дискретной интерполяции DSI. Во второй приводится программная реализация метода, тестирование на синтетических данных, анализ полученных результатов.

Результаты исследовательской работы могут быть использованы в геологическом моделировании, интерпретации сейсмических данных и планировании разработки месторождений, что способствует снижению экономических рисков и повышению эффективности добычи углеводородов.

1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ВОССТАНОВЛЕНИЯ ПОВЕРХНОСТИ С РАЗРЫВАМИ

1.1 Виды разрывных нарушений в геологии

Геологическая поверхность – это пространственная граница, разделяющая различные геологические структуры, такие как слои горных пород. Они могут характеризовать кровлю или подошву пласта, литологические границы и другие структурные особенности. При отсутствии тектонических нарушений поверхность предполагается гладкой и непрерывной, что приводит к применению классических методов интерполяции, например, кривым Безье, сплайнам, радиально-базисным функциям и другим методам. Однако в реальных геологических условиях поверхности редко бывают идеально гладкими. Тектонические процессы вызывают разрывы и дислокации, создавая разломы, которые нарушают гладкость и непрерывность поверхности. В таких случаях классические методы интерполяции недостаточно точны, так как не учитывают резкие изменения. Для адекватного моделирования таких структур используются специальные методы, например, дискретная гладкая интерполяция (DSI), которая позволяет учитывать наличие разрывов и сохраняет реалистичную топологию поверхности [5,19].

Скважинные данные как источник информации

Основным источником геолого-структурной информации являются данные, полученные в процессе бурения. Каждая скважина дает точку с координатами (x_i, y_i, z_i) , где z_i – уровень залегания определенного геологического слоя. Набор этих точек формирует выборку, на основе которой строится искомая поверхность [4,8].

Качество и плотность данных скважин сильно влияют на точность построенной модели. В районах с высокой плотностью бурения и высоким качеством измерений точность модели очень высока. Однако под

воздействием тектонических движений даже данные с высокой плотностью могут иметь большие расхождения, что свидетельствует о наличии разломов и смещений, поэтому необходимо это все очень внимательно отслеживать в процессе интерполяции [7].

Геологические карты

Геологическая карта – это двумерное представление распределения горных пород и структур на поверхности или на определенной глубине. Эта карта создается путем объединения данных буровых исследований, сейсмических исследований, геофизических исследований и других данных. Контурные линии, цветовые шкалы и символы часто используются для идентификации и описания различных геологических особенностей и аномалий [10].

Топографические карты являются важным инструментом в геологических исследованиях, особенно при выявлении линейных структур, таких как разломы, трещины и другие тектонические нарушения. Они отображают рельеф поверхности земли с помощью изолиний – линий равной высоты, что позволяет визуально оценить формы рельефа и их изменения. Линейные структуры часто проявляются на таких картах в виде резких изгибов, разрывов или сгущений изолиний, что указывает на наличие разломов или зон тектонических смещений [7].

Однако использование топографических и геологических карт имеет свои ограничения. Во-первых, если глубина данных, на основе которых построена карта, мала, то поверхностные признаки разломов могут быть нечеткими или отсутствовать вовсе. Во-вторых, при создании карт часто применяются методы сглаживания поверхности, которые не учитывают реальные геологические особенности. Это может привести к искажению или скрытию важных структурных элементов, таких как разрывы и смещения, особенно если они имеют сложную или нерегулярную форму.

Геологические карты, в отличие от топографических, содержат дополнительную информацию о литологии, стратиграфии и структурных

особенностях пород. Для определения мест разрывов вручную геологи активно используют именно эти карты, анализируя распределение и характер изолиний, а также другие геологические признаки. Одним из ключевых признаков разломов на геологических картах является сгущение изолиний – участки, где линии равной высоты расположены очень близко друг к другу. Такое сгущение свидетельствует о резких изменениях высоты, что часто связано с тектоническими смещениями и разрывами [10].

Кроме того, геологи обращают внимание на прерывания или смещения изолиний, которые могут указывать на разломные поверхности. Анализ таких признаков требует высокого профессионализма и опыта, поскольку не все сгущения или разрывы изолиний связаны с тектоническими нарушениями – они могут быть вызваны эрозионными процессами, изменениями литологии или другими геологическими факторами [7].

Таким образом, для точного выявления и интерпретации разрывных нарушений необходимо комплексное использование топографических и геологических карт в сочетании с полевыми наблюдениями, сейсмическими данными и другими геофизическими методами. Автоматизация и применение современных цифровых технологий, таких как геологическое моделирование и интерполяция с учётом разрывов, позволяют значительно повысить точность и эффективность выявления таких структур, минимизируя субъективность ошибки, связанные с ручным анализом карт.

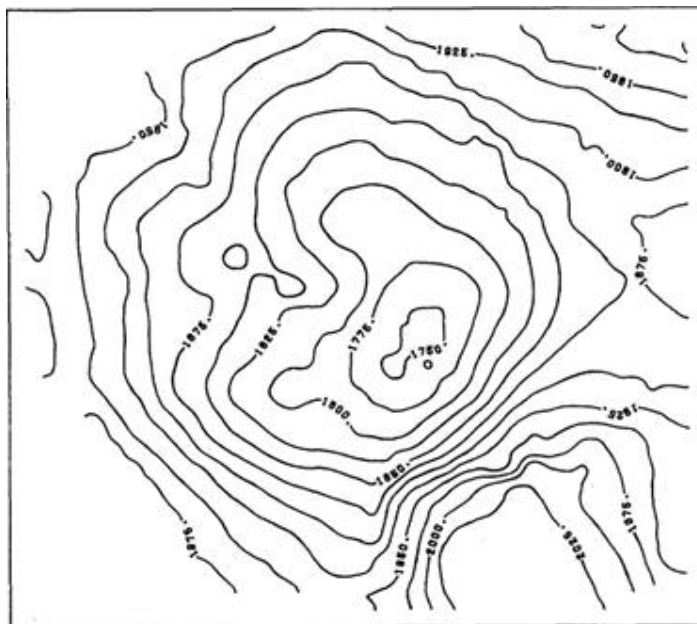


Рисунок 1.1 – Геологическая карта, вид сверху, без отметки разломов [22]

На данном рисунке изображены изолинии – линии равных значений исследуемой величины. Места их сгущения свидетельствуют о резких изменениях параметров, что может указывать на наличие разломных нарушений в геологическом разрезе. Такие зоны сгущения изолиний часто соответствуют границам тектонических блоков или зонам сдвигов и смещений пород, что делает их подозрительными на наличие разлома.

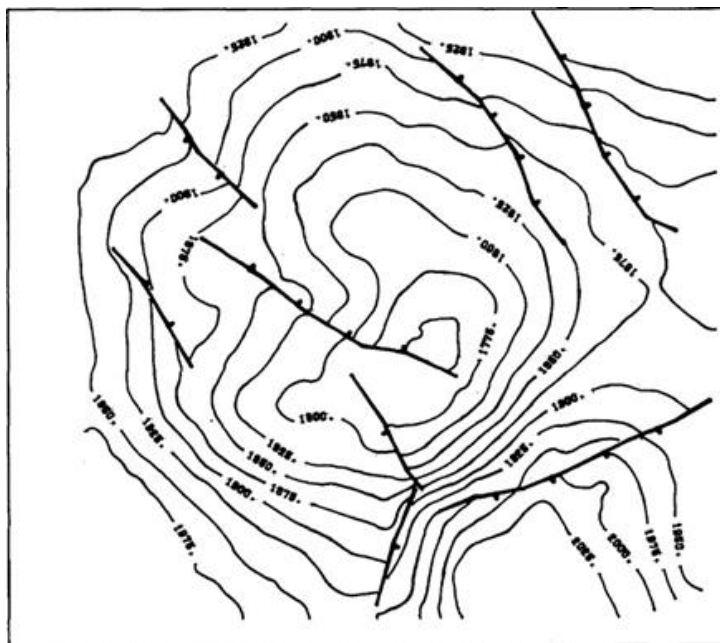


Рисунок 1.2. – Геологическая карта, вид сверху, отмечены места подозрительные на разрыв [22]

Разломы и их влияние на строение пласта

Разрыв – деформация пластов горных пород с нарушением их сплошности, возникающая в случае превышения предела прочности пород тектоническими напряжениями.

В геологической модели разлом изображается в виде линии (в двумерном случае) или плоскости (в трехмерном случае). Разломы оказывают существенное влияние на:

- геометрию месторождений полезных ископаемых – разломы могут изменять форму и размер залежи, образуя отдельные блоки различной продуктивности;
- пути миграции углеводородов – разломы могут образовывать барьеры и каналы для миграции нефти и газа;
- гидродинамические и фильтрационные процессы – смещения и трещиноватость влияют на проницаемость и распределение давления в резервуаре;
- расчеты запасов – игнорирование разрывов приводит к ошибкам в оценке запасов и планировании добычи.

Исключение разломов из расчета модели может привести к значительным ошибкам в буровых прогнозах, увеличивая экономические риски и снижая эффективность разработки месторождений. Поэтому в последние десятилетия возрос интерес к включению тектонических элементов в интерполяционные и геологические модели [6, 17].

Основные виды разрывных нарушений

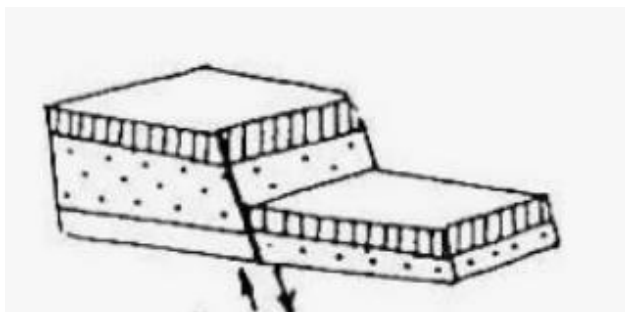


Рисунок 1.3 – Вид разрывного нарушения – сброс [11]

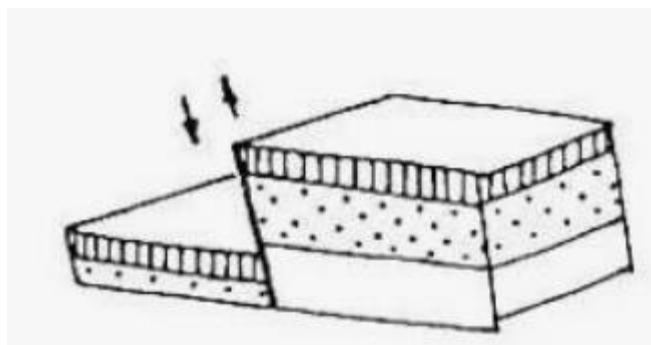


Рисунок 1.4 – Вид разрывного нарушения – взброс [11]

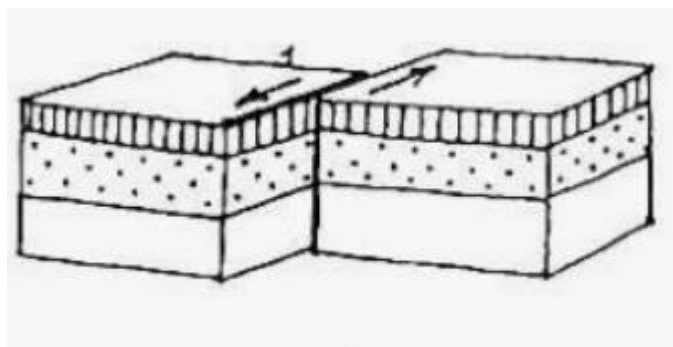


Рисунок 1.5 – Вид разрывного нарушения – сдвиг [11]

Разломы классифицируются по типу смещения и углу наклона:

Сброс (нормальный разлом) – вертикальное опускание висячего блока относительно лежачего.

Взброс (обратный разлом) – подъём висячего блока.

Сдвиг – преимущественно горизонтальное смещение блоков.

В целом, часто встречаются сложные системы разломов — зоны разломов, которые отражают множественные тектонические влияния и сложную геометрию разломов [10].

Сложность ручной идентификации разломов

Ручная разметка разломов – сложный процесс, требующий значительных усилий, высокого уровня знаний и глубокого понимания региональной тектоники.

Основные проблемы ручной разметки разломов и других структурных элементов в геологических данных связаны с рядом существенных факторов, которые существенно влияют на качество и эффективность работы специалистов [7].

Субъективность интерпретации

Одной из главных проблем является высокая степень субъективности при интерпретации данных. Разные геологи и специалисты могут по-разному оценивать одни и те же признаки на картах, сейсмических разрезах или других геологических материалах. Это связано с различным опытом, уровнем подготовки, а также с индивидуальными особенностями восприятия и анализа информации. В результате одна и та же зона может быть отмечена как разлом у одного специалиста и не замечена другим, что снижает воспроизводимость и надёжность результатов.

Низкое качество и точность данных, а также шум

Геологические и сейсмические данные часто содержат значительный уровень шума, артефактов и неточностей, вызванных техническими ограничениями оборудования, условиями сбора данных и природными факторами. Это затрудняет выделение чётких границ разломов и других структурных нарушений. Наличие шума усложняет фильтрацию и обработку данных, увеличивает вероятность ошибок при ручной разметке и требует дополнительного времени на проверку и корректировку результатов [8].

Отсутствие чётких характеристик смещений при малой амплитуде разломов

Некоторые разломы и трещины имеют малую амплитуду смещения, что делает их практически незаметными на геологических картах и сейсмических профилях. Такие слабовыраженные нарушения сложно идентифицировать, особенно при отсутствии дополнительных данных или при низком разрешении исходных материалов. Это приводит к пропуску важных структур, что может негативно сказаться на точности геологического моделирования и прогнозировании поведения пласта [15].

Большой объём обрабатываемых данных и значительные затраты времени и усилий

Современные геологические исследования и сейсмические съёмки генерируют огромные объёмы данных, которые необходимо тщательно

проанализировать и разметить. Ручная обработка таких массивов информации требует значительных временных и трудовых ресурсов. Специалисты вынуждены проводить долгие часы за анализом карт, профилей и других материалов, что повышает риск усталости, снижения концентрации и, как следствие, ошибок. Кроме того, длительный процесс разметки замедляет принятие решений и разработку месторождений, что негативно влияет на экономическую эффективность проектов [8].

Указанные проблемы обосновывают необходимость разработки и внедрения автоматизированных методов и алгоритмов для интерпретации и разметки разломов, которые позволят повысить точность, снизить субъективность и существенно сократить время обработки геологических данных.

Для решения этих задач разрабатываются автоматические и полуавтоматические методы на основе алгоритмов с использованием графов, вариационных подходов, методов машинного обучения, и в частности, используя метод дискретной гладкой интерполяции (DSI), который позволяет обеспечить учет разломов в моделируемых поверхностях и повысить точность построения геологических моделей.

1.2 Геометрическая интерпретация разрывных нарушений

В геологии разломы образуются в результате тектонических процессов, которые приводят к нарушению сплошности породы. В математическом смысле разлом – это линия или кривая, вдоль которой наблюдается скачок значений геологической поверхности. Таким образом, поверхность становится непрерывной по частям, но разрывной вблизи линии смещения [17].

В рамках численного моделирования разрывное нарушение определяется как множество точек, которое образует ломаную линию, вдоль нее нарушается гладкость интерполированной поверхности [7].

Эта линия делит поверхность на области, где могут быть применены стандартные методы интерполяции, но функция не сохраняет гладкость через разлом. Это ключевое условие в методах, учитывающих разрывы, таких как метод дискретной гладкой интерполяции (DSI) [7,23].

Геометрически разлом в задачах интерполяции определяется как полилиния, соединяющая вручную или автоматически известные входные точки. Эта линия: не пересекается с другими разломами; может быть открытой или замкнутой; может состоять из нескольких сегментов, образующих сложную форму; определяется как список сегментов или множества точек координат, через которые проходит разлом [10].

При разработке интерполяционной модели точки, которые находятся по разные стороны от разлома, не должны включаться при расчете разностей или градиентов, они исключаются из сглаживания, чтобы сохранить физически обусловленный скачок значений [7, 14].

Типы разломов в данной работе

Рассматриваются следующие геометрические типы разрывных нарушений:

- полный разлом – проходит через всю область моделирования и делит поверхность на две части, обычно он представлен прямой линией;
- разлом, выходящий из центра поверхности и идущий к ее краю, – начинается с центральной области и распространяется к краю области, создавая локализованный разлом;
- разлом, заданный замкнутой линией в центре, – это внутренний замкнутый разлом, который полностью находится в пределах области модели;
- разлом, заданный ломаной линией, состоит из нескольких сегментов, он моделирует сложную тектоническую среду;
- множественные разломы, на поверхности есть несколько независимых разломов.

Значение геометрической интерпретации

Правильное представление разломов в виде ломаных линий позволяет:

- точно учитывать структуру геологических нарушений;
- корректно сформулировать численную задачу интерполяции;
- исключить искажения поверхности вблизи разлома;
- правильно применять методы, основанные на локальных свойствах гладкости [7].

Визуально разломы представлены в виде ломаных линий на геологических картах или 3D-моделях. Их наличие оказывает значительное влияние на восстановленную поверхность, поэтому требует точной и корректной интерпретации [10].

1.3 Методы интерполяции поверхности

Интерполяция – это процесс восстановления значений функции в промежуточных точках с использованием известных дискретных данных. В геологическом моделировании он позволяет создать цифровую модель поверхности пласта или геологического горизонта, например, с использованием точечных данных из скважины. Существуют классические методы интерполяции, которые различаются по математическому аппарату, точности и способности учитывать свойства данных, включая разрывы [7].

1.3.1 Линейная интерполяция

Линейная интерполяция восстанавливает значение функции в точке между двумя известными точками как значение на прямой, соединяющей эти точки.

Для одномерного случая при известных значениях (x_0, y_0) и (x_1, y_1) значение y в точке x , где $x_0 < x < x_1$, вычисляется как:

$$y = y_0 + \frac{(x - x_0)}{(x_1 - x_0)} (y_1 - y_0) \quad (1.1)$$

В двумерном пространстве тот же метод используется с плоскостями.

Преимущества:

- простота реализации и вычислений;
- хорошо работает с плотно и равномерно распределенными данными.

Ограничения:

- не обеспечивает гладкости (кусочно-линейная функция);
- не учитывает разрывы – сглаживает их, тем самым искажая фактическую геологическую структуру [14].

1.3.2 Интерполяция сплайнами

Сплайны – это кусочно-полиномиальные функции, которые обеспечивают гладкое соединение между узловыми точками, гарантируя непрерывность первой и второй производных. Наиболее популярны кубические сплайны.

Формула (в качестве примера кубический сплайн)

На каждом интервале $[x_i, x_{i+1}]$ функция задается полиномом третьей степени:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \quad (1.2)$$

Коэффициенты a_i, b_i, c_i, d_i выбираются так, чтобы обеспечить гладкость и непрерывность в стыках.

Преимущества:

- гладкая и естественная поверхность;
- хорошо аппроксимируют сложные формы.

Ограничения:

- предполагают гладкость и непрерывность, что не соответствует разрывам;
- сглаживают разломы, теряя важную геологическую информацию [14].

1.3.3 Метод обратных расстояний (IDW)

Значение функции в точке рассчитывается как взвешенное среднее значений соседних точек, где вес обратно пропорционален расстоянию:

$$f(x) = \frac{\sum_{i=1}^N w_i f(x_i)}{\sum_{i=1}^N w_i}, \quad (1.3)$$

$$w = \frac{1}{d(x, x_i)^p}, \quad (1.4)$$

где $d(x, x_i)$ – расстояние между точкой x и известной точкой x_i , p – степень обратной зависимости (обычно $p = 2$).

Преимущества:

- простота и локальность;
- хорошо работает с равномерным распределением данных.

Ограничения:

- не учитывает разрывы – сглаживает скачки;
- чувствителен к выбору параметра p [26].

1.3.4 Кригинг

Кригинг (геостатические методы) основан на статистической модели пространственной корреляции, представленной вариограммой. Метод уменьшает дисперсию ошибок интерполяции, принимая во внимание структуру данных [14].

Интерполируемое значение:

$$\hat{Z}(x_0) = \sum_{i=1}^N \lambda_i Z(x_i), \quad (1.5)$$

где веса λ_i находятся из решения системы уравнений, основанной на вариограмме (статистический момент второго порядка, применяемый в геостатистике для анализа и моделирования пространственной корреляции

между значениями переменной, измеряемой в разных точках пространства)
 $\gamma(h)$:

$$\sum_{j=1}^N \lambda_j \gamma(|x_i - x_j|) + \mu = \gamma(|x_i - x_0|), \quad i = 1, \dots, N \quad (1.6)$$

μ – множитель Лагранжа для обеспечения несмещенности (свойство статистической оценки, означающее, что в среднем по всем возможным выборкам оценка равна истинному значению оцениваемого параметра).

Преимущества

- учитывает пространственную структуру и анизотропию;
- позволяет строить оптимальные по дисперсии оценки.

Ограничения:

- предполагает некоторую степень гладкости;
- классический кригинг не моделирует резкие разрывы без модификаций [15].

1.3.5 Радиальные базисные функции (RBF)

RBF аппроксимируют функцию как сумму радиальных функций, зависящих от расстояния до центров:

$$s(x) = \sum_{i=1}^N w_i \phi(\|x - x_i\|), \quad (1.7)$$

где $\phi(r)$ – радиальная базисная функция, w_i – веса.

Преимущества:

- высокая гладкость и точность;
- способны моделировать сложные поверхности.

Ограничения:

- предполагают непрерывность;
- не учитывают разрывы без специальной адаптации [28].

1.3.6 Метод интерполяции по принципу минимальной кривизны

Этот метод предназначен для построения гладких поверхностей (например, контурных карт) по набору точечных наблюдений, характерных для геофизических данных.

Дифференциальное уравнение для минимальной кривизны

$$\frac{\partial^4 u}{\partial x^4} + 2 \frac{\partial^4 u}{\partial x^2 \partial y^2} + \frac{\partial^4 u}{\partial y^4} = 0, \quad (1.8)$$

Где u – значение интерполируемой функции в точке с координатами (x, y) .

Это уравнение описывает поверхность с минимальной общей кривизной (аналогично тонкой металлической пластине, изгибающейся под действием точечных сил).

Принцип минимальной суммарной кривизны

Минимизируется интеграл:

$$C(u) = \iint \left[\left(\frac{\partial^2 u}{\partial x^2} \right)^2 + \left(\frac{\partial^2 u}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 u}{\partial y^2} \right)^2 \right] dx dy. \quad (1.9)$$

Метод конечных разностей

Для регулярной сетки используется дискретная аппроксимация кривизны в узле (i, j) :

$$C_{i,j} = \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{h^2}. \quad (1.10)$$

Метод минимальной кривизны позволяет строить высококачественные контурные карты, которые хорошо согласуются с работой опытных картографов. Поверхность получается гладкой, без артефактов и искусственных перегибов. Метод применим для любых точечных данных, не обязательно лежащих на регулярной сетке.

Обычная схема метода минимальной кривизны разрывы сглаживает, не отображая их явно. Для интерполяции по данным с реальными разрывами

необходимы специальные дополнения к методу либо ручное разбиение расчетной области на "до разрыва" и "после разрыва" [12].

1.3.7 Метод интерполяции поверхности с известными разрывами и штрафом за пересечение разрыва

В данном методе применяется классический кригинг для получения значений поверхности в узлах регулярной сетки, учитывая вариабельность данных и известные места разрывов. Этот этап обеспечивает исходные данные с регулярной сеткой.

Эстетический этап оценки – на основе полученных данных на сетке строится гладкая поверхность с учётом разрывов, чтобы избежать искажений и «ступенчатого» эффекта на контурных линиях. Для этого в каждом узле сетки вычисляется касательный квадратичный многочлен, аппроксимирующий поверхность в окрестности узла.

Основная формула интерполяции:

$$Z_E = \frac{\sum_{I \in V_E} p_{EI} Q_I(x_E, y_E)}{\sum_{I \in V_E} p_{EI}}, \quad (1.11)$$

где Z_E – искомое значение поверхности в точке E с координатами (x_E, y_E) ;

V_E – окрестность точки E , то есть набор узлов сетки, находящихся на расстоянии меньше радиуса R от E (с учетом разрывов);

I – индекс узла сетки в окрестности V_E ;

p_{EI} – весовой коэффициент, зависящий от расстояния между E и I ;

$Q_I(x_E, y_E)$ – значение квадратичного многочлена (локальной аппроксимации) в точке (x_E, y_E) , построенного на основе данных в узле I .

Функция веса

$$p(r) = \frac{1}{r^2} f(r), \quad (1.12)$$

где

$$f(r) = 1 - \left(\frac{r}{R}\right)^2 \left[3 - 2\frac{r}{R}\right], \quad 0 < r < R \quad (1.13)$$

$p(r)$ – вес, который определяет вклад узла на расстоянии r в интерполяцию;

r – модифицированное расстояние между точками;

R – радиус окрестности, в пределах которого учитываются точки;

$f(r)$ – сглаживающая функция, обеспечивающая плавное уменьшение веса к нулю на границе окрестности ($r=R$).

Модифицированное расстояние с учётом разрывов

$$r_{EI} = dE_{EI} + 2 \sum \min(dE_p^{F_0}, dE_p^{F_n}), \quad (1.14)$$

r_{EI} – "реальное" расстояние между точкой оценки E и узлом I , учитывающее разрывы;

d_{EI} – обычное евклидово расстояние между E и I ;

F_0, F_n – начальная и конечная точки разрыва;

P – точка пересечения прямой EI с линией разрыва;

$d_{EP}^{F_0}$ – расстояние вдоль разрыва от точки пересечения P до его ближайшего конца F_0 ;

$d_{EP}^{F_n}$ – расстояние вдоль разрыва от точки пересечения P до другого конца F_n .

Суммирование ведётся по всем пересечениям линии EI с разрывами между E и I . Множитель 2 усиливает «штраф» за пересечение разрыва.

Локальный квадратичный многочлен

$$Q_I(x, y) = Z_I + a_1(x - x_I) + a_2(y - y_I) + a_3(x - x_I)(y - y_I) + a_4(x - x_I)^2 + a_5(y - y_I)^2. \quad (1.15)$$

$Q_I(x, y)$ – значение аппроксимирующей функции в точке (x, y) , построенной на основе данных в узле I ;

Z_I – значение поверхности в узле I (вычисленное на первом этапе, например, методом кригинга);

(x_I, y_I) – координаты узла I ;

a_1, a_2, a_3, a_4, a_5 – коэффициенты, определяющие наклон, кривизну и взаимодействие по осям x и y . Они находятся путём минимизации ошибки аппроксимации по соседним точкам [22].

1.3.8 Метод Монте-Карло

Следующий метод уже не будет относиться к классическим. Метод восстановления поверхности с использованием метода Монте-Карло для перебора вариантов смещений по разлому. Метод основан на стохастическом подходе. Используется последовательный метод отбора Монте-Карло для генерации множества возможных моделей смещения разломов. Это можно рассматривать как метод пропорционального сопоставления и интерполяции по ломано-криволинейной схеме, специально рассчитанной на области с разрывами.

На каждом разломе берутся крайние точки (экстремумы) с обеих сторон разлома. Дальше любые промежуточные значения смещений по разлому находятся с помощью нормированной кривой вдоль границы.

Положение крайних точек выбирается случайным образом много раз (метод Монте-Карло), чтобы смоделировать разные возможные варианты разлома и оценить, насколько это влияет на итоговую модель. Для каждого набора таких точек строится своя модель смещений разлома.

Интерполяция по нормированной криволинейной координате (вдоль границы разлома):

$$s = \frac{l}{L}, \quad (1.16)$$

s – нормированная криволинейная координата точки на границе разлома. Это число от 0 (начало границы) до 1 (конец границы).

l – длина дуги от выбранной начальной точки (экстремума) вдоль границы разлома до интересующей нас точки на этой же границе.

L – общая длина рассматриваемой границы разлома (от начала до конца).

Параметр s определяет «относительное положение» точки на границе разлома. Если $s = 0,5$, это середина границы; если $s = 0,25$, четверть пути от начала.

Интерполяция смещения вдоль разлома:

$$d_{fault}(s) = x_{hanging}(s) - x_{footwall}(s), \quad (1.17)$$

$d_{fault}(s)$ – вектор смещения по разлому в точке с координатой s .

Показывает, как одна сторона разлома сместилась относительно другой.

$x_{hanging}(s)$ – координаты точки на верхнем/висячем блоке разлома (hanging wall) при том же s .

$x_{footwall}(s)$ – координаты соответствующей точки на нижнем/лежащем блоке разлома (footwall) при том же s .

Для каждой относительной точки (с равным s) на обеих сторонах разлома рассчитывается разница координат. Это и есть искомое смещение по разлому.

Вероятностный выбор положения экстремума:

$$S_i \sim P_0(S_i), \quad (1.18)$$

S_i – случайная переменная, определяющая положение i -го экстремума (крайней точки, например, угла или пересечения границы разлома) вдоль границы.

$P_0(S_i)$ – априорное распределение этой переменной. Оно задаёт диапазон возможных положений экстремума, отражающий геологическую неопределённость или интерпретацию (где реально может быть эта точка).

Положение каждого важного экстремума на разломе выбирается случайно из заранее заданного диапазона, чтобы смоделировать ансамбль возможных вариантов.

1. Критерий качества через дилатацию:

$$\theta = \frac{dS - ds_{(0)}}{dS}, \quad (1.19)$$

θ – дилатация, показывает, насколько сильно изменяется площадь выбранного участка при реставрации (то есть при «развертывании» горизонта в исходное, недеформированное состояние).

dS – элемент площади на поверхности после разлома (исходная, наблюдаемая площадь участка).

$ds_{(0)}$ – элемент площади того же участка после его восстановления.

θ отражает относительное растяжение или сжатие участка поверхности после реставрации (насколько искажается поверхность). Если θ близко к нулю, значит, восстановленная поверхность похожа по площади на исходную, то есть рассматриваемый вариант смещения по разлому физически допустим.

Таким образом, метод Монте-Карло не относится к классическим методам интерполяции, а представляет собой современный стохастический инструмент для учёта структурной неопределённости в геологических моделях [13].

1.3.9 Метод с использованием уравнения Навье-Стокса с обратным временем и отрицательным временным шагом адвекции

Адвекция в данном случае – это пересчёт положения точек модели на каждом шаге времени согласно полю скоростей, чтобы восстановить исходную конфигурацию пород.

Этот подход позволяет не только распределять значения перемещений (или скоростей) внутри массива по физическим законам, но и учитывать сложные граничные условия, наличие разломов, неоднородные параметры материалов. В рамках метода проводится численное моделирование реставрации структуры "назад во времени" посредством решения полной задачи механики сплошной среды с учётом гравитации.

Смещения внутри массива не задаются вручную, а рассчитываются так, как если бы структура “разворачивалась” обратно к исходному (до деформационного) состоянию под действием гравитации и с учетом заданных граничных условий.

Все параметры (например, свойства материала: вязкость, плотность) также интерполируются по частицам и сетке.

Основные уравнения

Уравнение сохранения импульса:

$$\nabla \cdot \sigma + f = 0, \quad (1.20)$$

где ∇ – оператор набла, σ – тензор напряжений, а f – удельная сила тела (обычно объемный вес ρg).

Уравнение сохранения массы для несжимаемых жидкостей (уравнение неразрывности):

$$\nabla \cdot v = 0, \quad (1.21)$$

где v – скорость.

Разложение тензора напряжений:

$$\sigma = \tau - pI, \quad (1.22)$$

где τ – девиатор тензора напряжений, p – давление, I – единичный тензор.

В предположении вязкого течения девиаторная часть напряжения равна:

$$\tau = 2\eta D, \quad (1.23)$$

где η – динамическая вязкость, а D – тензор скорости бесконечно малой деформации, определяемый как

$$D = \frac{1}{2} [\nabla v + (\nabla v)^T]. \quad (1.24)$$

Объединяя уравнения, получаем:

$$\nabla \cdot [\eta(\nabla v + (\nabla v)^T)] - \nabla p = -\rho g. \quad (1.25)$$

Обратная адвекция:

$$x(t - \Delta t) = x(t) - v(t) \cdot \Delta t, \quad (1.26)$$

где $x(t)$ – положение частицы, $v(t)$ – скорость в момент t , Δt – шаг по времени.

Применение только физических (“естественных”) граничных условий возможно, но требует внимательного подбора эффективных значений вязкости для разных частей структуры; без правильных параметров восстановление будет недостоверным. Для реалистичного восстановления

сложных структур важно использовать физически осмысленные условия на границах модели, а не жёстко предписывать их форму.

Разработанный метод способен не просто интерполировать формы горизонтов, а восстанавливать историю развития всей структуры с учётом разломов и всех сложных взаимодействий, что недостижимо для традиционных геометрических подходов [24, 25].

1.3.10 Трёхмерный дип-доменный подход

Трёхмерный дип-доменный подход для реконструкции сложных складчатых и разновозрастных слоёв, что позволяет корректно моделировать поверхности с разрывами и переменной мощностью слоёв.

Этот подход основан на разбиении пространства на области (dip-domains), внутри которых падение слоёв считается постоянным, а границы между доменами могут быть как непрерывными (например, осевые поверхности складок), так и разрывными (например, разломы, стратиграфические несогласия).

Построение отдельных горизонтов

Для каждого геологического горизонта создается отдельная 3D-модель, при этом используются все доступные данные (полевые измерения, данные сейсморазведки, результаты анализа керна и др.). Данные из разных стратиграфических уровней также используются для уточнения формы горизонта.

Восстановление геологических структур

Для учета деформаций, таких как складкообразование, применяется метод восстановления скольжением по плоскостям напластования. В данном случае напластования – это не просто слои осадочных пород, а динамичные элементы, геометрия и толщина которых отражают тектонические процессы и историю осадконакопления в бассейне. Этот метод позволяет "развернуть"

геологические структуры и восстановить их первоначальную форму до деформации [16].

Разбиение пространства на области (dip-domains)

$$m(x) = \sum_{j=1}^J \sum_{k=1}^K m_k \varphi_k^j(x), \quad (1.27)$$

где $m(x)$ – значение интересующего параметра в точке x ;

J – количество подобластей (domains, dip-domains), на которые разбита вся модель;

K – количество базисных функций, используемых для аппроксимации параметра внутри каждой подобласти;

m_k – коэффициент (вес), определяющий вклад k -й базисной функции;

$\varphi_k^j(x)$ – базисная функция с номером k в j -й подобласти, определяет, как параметр меняется внутри этой области.

Моделирование горизонтов

$$m(x) = m(u(x)) = \sum_{k=1}^K m_k \psi_k(u(x)), \quad (1.28)$$

где $m(x)$ – значение параметра (например, высоты горизонта) в точке x ;

$u(x)$ – параметризация точки x в некоторой локальной системе координат (вдоль профиля или по нормали к поверхности);

K – число базисных функций;

m_k – коэффициенты (веса) для каждой базисной функции;

$\psi_k(u(x))$ – базисная функция, определяющая вклад в значение параметра в зависимости от положения точки x в локальной системе координат.

Учет эрозии

$$H_{erod} = \sum_{i=1}^I \mathbb{I}_E c_i \chi_i^H(x), \quad (1.29)$$

где H_{erod} — значение функции, описывающей эродированный горизонт (после учета эрозии или несогласия);

I — количество базисных функций, используемых для описания горизонта;

$\mathbb{I}_E$ — индикаторная функция, равная 1 внутри области ниже эрозионной поверхности (или в другой заданной области), и 0 вне её (то есть, "выключает" горизонт вне нужной области);

c_i — коэффициенты (веса) для i -й базисной функции;

$\chi_i^H(x)$ — базисная функция для i -го узла, определяющая форму горизонта H в точке x [27].

В основном, методы интерполяции, часто предполагают гладкость и непрерывность поверхности. При наличии разломов это приводит к их сглаживанию, ухудшению модели и снижению точности прогноза. Эффективное моделирование поверхностей с разрывом требует специальных методов, таких как дискретная гладкая интерполяция (DSI), которая сохраняет топологию разлома и обеспечивает гладкость только в непрерывных частях поверхности.

1.4 Метод дискретной гладкой интерполяции (DSI)

Интерполяция функции $f(\cdot)$, известной в некоторых точках множества R^p , является распространённой задачей. Многие компьютерные приложения (например, автоматическая генерация контуров) требуют интерполяции только в узлах заданной сетки. Большинство классических методов находят функцию, определённую во всём пространстве, но метод гладкой дискретной интерполяции избегает этого и вычисляет значения только в узлах сетки. Для двумерных регулярных сеток частным случаем этого метода является метод Бриггса [26].

Рассмотрим область $\Omega^* \subset R^p$, покрытую сеткой Ω , состоящей из конечного множества точек. Если эти точки расположены в пересечениях

равноудалённых параллельных прямых, такая сетка называется регулярной. В противном случае она называется нерегулярной [20].

Функция $f(\cdot)$ является кусочно-непрерывной, а её разрывы представлены множеством Δ гиперповерхностей размерности $(p - 1)$. Функция $f(\cdot)$, известна в подмножестве узлов L , а в подмножестве I её нужно интерполировать:

L – множество узлов, в которых известны значения $f(\cdot)$;

I – множество узлов, где значения нужно интерполировать;

$\Omega = L \cup I$ – вся сетка.

Узлы сетки Ω нумеруются. Вводится система обозначений:

$k \in \Omega$, означает, что узел k принадлежит сетке Ω ;

$l \in L$, означает, что узел l принадлежит множеству известных значений L ;

$i \in I$, означает, что узел i принадлежит множеству интерполируемых узлов I [18].

Таким образом, проблема формулируется следующим образом, требуется найти вектор:

$$\varphi = [\varphi_1, \varphi_2, \dots, \varphi_N]^T, \quad (1.30)$$

где $N = |\Omega|$ – количество узлов в сетке.

Условия, которым должна удовлетворять интерполяция:

– для узлов из множества L значения функции известны

$$\varphi_l = f(\text{узел } l), \forall l \in L; \quad (1.31)$$

– для узлов из множества I значения должны быть аппроксимированы

$$\varphi_i \approx f(\text{узел } i), \forall i \in I. \quad (1.32)$$

Вектор $\varphi = [\varphi_1, \varphi_2, \dots, \varphi_N]^T$ – это набор значений функции $f(\cdot)$ в узлах сетки Ω . То есть, каждая φ_i – это значение функции в некоторой точке сетки.

Таким образом, метод должен обеспечить гладкую интерполяцию значений, сохраняя известные данные в точках множества L и вычисляя приближенные значения в точках множества I [18].

Проблема интерполяции имеет бесконечное число решений, поэтому нужен способ выбрать наилучший вариант. Как правило, L отличается от Ω , и в этом случае получаем бесконечное количество решений. Чтобы выбрать «хорошее» решение, определим глобальный критерий шероховатости $R(\varphi)$, и интерполятор φ будет выбран таким образом, чтобы выполнялись следующие условия:

- $R(\varphi)$ – минимально;
- $\varphi_l = f(\text{узел } l), \quad \forall l \in L$ [20].

Метод дискретной гладкой интерполяции (DSI) предлагает измерять шероховатость каждого возможного решения φ с помощью квадратичного критерия $R(\varphi)$, имеющего следующий вид:

$$R(\varphi) = \varphi^T \cdot [W] \cdot \varphi, \quad (1.33)$$

где $[W]$ – положительно полуопределенная симметричная матрица [21].

Таким образом, задача формулируется следующим образом:

минимизировать $R(\varphi) = \varphi^T \cdot [W] \cdot \varphi$, при ограничении
 $\varphi_l = f(\text{узел } l), \quad \forall l \in L.$

Локальный критерий шероховатости $R_i(\varphi|k)$

Для каждого узла k , принадлежащего сетке Ω , можно определить несколько локальных критериев шероховатости: $\{R_1(\varphi|k), R_2(\varphi|k) \dots\}$.

Метод DSI требует, чтобы каждое из этих измерений $R_i(\varphi|k)$ имело квадратичный вид:

$$R_i(\varphi|k) = \left| \sum_{n \in N_i(k)} v_i^\alpha(k) \cdot \varphi_\alpha \right|^2, \quad (1.34)$$

$N_i(k)$ – подмножество соседних узлов сетки Ω , окружающих узел k .

$v_i^\alpha(k)$ – весовые коэффициенты, регулирующие вклад соседних узлов в измерение шероховатости.

Матричная формулировка

Пусть $N = |\Omega|$, и пусть $[W_i(k)]$ – это симметричная матрица размерности $N \times N$, такая что:

$$\{W_i(k)\}^{\alpha\beta} = \begin{cases} v_i^\alpha(k) \times v_i^\beta(k), & \text{если } \alpha \in N_i(k) \text{ и } \beta \in N_i(k), \\ 0, & \text{иначе} \end{cases} \quad (1.35)$$

[18].

Матрица зависит от индексов узлов и весовых коэффициентов. То есть, матрица заполняется только в тех случаях, когда оба индекса α и β принадлежат множеству соседних узлов $N_i(k)$. Если же узлы не являются соседями, то значение равно нулю [20].

На основе введённой выше матрицы мы можем вычислить локальную шероховатость в узле k :

$$R_i(\varphi|k) = \varphi^T \cdot [W_i(k)] \cdot \varphi. \quad (1.36)$$

Как можно заметить, имеем:

$$R_i(\varphi|k) = 0 \Leftrightarrow \begin{cases} \varphi_k = \sum_{\substack{\alpha \in N_i(k), \\ \alpha \neq k}} \lambda_i^\alpha(k) \cdot \varphi_\alpha, \\ \lambda_i^\alpha = -\frac{v_i^\alpha(k)}{v_i^k(k)}. \end{cases} \quad (1.37)$$

Эта формулировка показывает, что если $R_i(\varphi|k) = 0$, то φ_k можно рассматривать как *собственное скользящее среднее*, связанное с весами $\lambda_i^\alpha(k)$. Поскольку скользящие средние часто являются гладкими, естественно считать, что $R_i(\varphi|k)$ является критерием, измеряющим локальную шероховатость в узле k . [18].

Соседства $N_i(k)$

Метод DSI не зависит от того, как определяются соседства узлов. Тем не менее, для практического использования удобно задать следующие два правила:

1. Узлы сетки связаны так, чтобы образовывать n -мерный граф, если $\varphi(\cdot)$ является функцией от n переменных. Этот граф строится таким образом, что два узла (α, β) связаны только в том случае, если они являются непосредственными соседями друг друга.

Если интерполяция выполняется для двумерных данных (например, изображений), узлы связаны так, чтобы образовывать решётку.

Если данные трехмерные (например, высота рельефа в геодезии), узлы должны быть связаны в 3D-сетку.

В общем случае, если φ зависит от n переменных, то узлы образуют n -мерный граф.

2. Определение соседства

$N_i(k)$ определяется как множество узлов α из Ω , таких что существует путь длиной не более $n_i(k)$ шагов между узлами k и α . Количество шагов $n_i(k)$ является заданным параметром [20].

Подмножество K_i , связанное с критерием $R_i(\varphi|k)$

При интерполяции могут возникнуть проблемы, если узел находится у границы области или рядом с резким разрывом. Чтобы избежать ошибок, вводится специальное множество K_i , где интерполяция остаётся корректной.

Для заданного узла k локальный критерий $R_i(\varphi|k)$ может быть невычислимым по двум причинам:

1. k находится рядом с границей области Ω , и некоторые узлы множества $N_i(k)$ выходят за эту границу;

2. k находится рядом с разрывом, и внутренняя часть выпуклой оболочки $N_i^*(k)$ множества соседей $N_i(k)$ пересекается с разрывом δ , принадлежащим множеству разрывов Δ [20].

По этим двум причинам мы определяем K_i как подмножество узлов $k \in \Omega$, в которых $R_i(\varphi|k)$ может быть вычислено.

Второе условие, указанное выше, является ключевым для учета разрывов. На практике выгодно, чтобы $|K_i|$ было как можно больше, и по этой причине разумно делать соседства $N_i(k)$ небольшими, как можно более компактными, чтобы избежать пересечения $N_i^*(k)$ с границей или разрывами.

Глобальный критерий шероховатости $R(\varphi)$

После выбора нескольких локальных мер шероховатости $\{R_1(\varphi|k), R_2(\varphi|k), \dots\}$, глобальный критерий $R(\varphi)$ может быть определен следующим образом:

$$R(\varphi) = c_1 \cdot R_1(\varphi) + c_2 \cdot R_2(\varphi) + \dots \quad (1.38)$$

при этом

$$R_i(\varphi) = \sum_{k \in K_i} \mu_i(k) \cdot R_i(\varphi|k). \quad (1.39)$$

В этой формуле коэффициенты $\mu_i(k)$ и c_i являются весами, которые выбираются с учетом относительной важности: узла k при вычислении $R_i(\varphi)$, критерия $R_i(\varphi)$ при вычислении $R(\varphi)$ [20].

Используя матричное представление критерия $R_i(\varphi|k)$, получаем:

$$R_i(\varphi) = \varphi^T \cdot [W] \cdot \varphi, \quad (1.40)$$

при этом

$$[W] = c_1 \cdot \sum_{k \in K_1} \mu_1(k) \cdot [W_1(k)] + c_2 \cdot \sum_{k \in K_2} \mu_2(k) \cdot [W_2(k)] + \dots \quad (1.41)$$

[18].

В случае с регулярными сетками форма локального соседства и веса внутри не изменяются от одного узла сетки к другому. Аппроксимация первых и вторых производных в узле определяется нахождением центральной разности.

Если сетка нерегулярная, определить критерии шероховатости через производные невозможно. Тогда интерполятор может иметь форму:

$$\varphi_k = \sum_{\alpha \in N_i(k)} \lambda_i^\alpha(k) \cdot \varphi_\alpha. \quad (1.42)$$

Существует множество способов выбора коэффициентов, например:

$$v_i^l(k) = \lambda_i^\alpha(k), \quad \forall \alpha \neq k, \quad v_i^l(k) = -1, \quad (1.43)$$

$\lambda_i^\alpha(k)$ — коэффициенты кригинговой интерполяции. Эти коэффициенты также могут быть получены из интерполяции Шепарда, которую можно рассматривать как упрощенную версию кригинга [20,26].

Не существует общего правила для выбора конкретного критерия $R(\varphi)$. Этот выбор будет зависеть от специфического характера функции подлежащей интерполяции [20].

Например, можно рассмотреть задачу интерполяции верхней границы стратиграфического слоя, имеющего высоту $z = f(x, y)$. Поскольку такие слои обычно почти горизонтальны, разумно предположить, что интерполятор удовлетворяет условиям $\left| \frac{\partial \varphi}{\partial x} \right|^2$ и $\left| \frac{\partial \varphi}{\partial y} \right|^2$ минимальны [20].

Более того, если рассматривать слой как упругую балку, которая была деформирована под действием геологических напряжений, известно, что касательные напряжения в слое пропорциональны вторым производным.

В любой точке, где касательные напряжения превышают заданный порог, происходит разрыв, вызывающий разрывность (то есть геологический разлом).

По этой причине разумно принять следующее условие в любой точке (x, y) где отсутствуют разломы:

$$\left| \frac{\partial^2 \varphi}{\partial x^2} \right|^2, \left| \frac{\partial^2 \varphi}{\partial y^2} \right|^2, \left| \frac{\partial^2 \varphi}{\partial x \partial y} \right|^2 \text{ минимальны.}$$

Чтобы выполнялись данные условия для регулярной сетки можно применить следующий глобальный критерий шероховатости:

$$R(\varphi) = \gamma \cdot \{R_1(\varphi) + R_2(\varphi)\} + (1 - \gamma) \cdot \{R_3(\varphi) + R_4(\varphi)\}, \quad (1.44)$$

где γ — это действительное число, принадлежащее интервалу $[0, 1]$ [18].

Из условия минимизации глобального критерия гладкости при жёстком учёте известных значений получаем систему линейных уравнений, определяющую значения интерполируемой функции в неизвестных узлах регулярной или нерегулярной сетки:

(DSI):

$$\begin{cases} [W_{II}] \cdot \varphi_I = \psi_I, \\ \text{где } \psi_I = -[W_{IL}] \cdot \varphi_L. \end{cases} \quad (1.45)$$

[18].

Метод DSI легко учитывает разрывы. DSI очень эффективен для интерактивного моделирования: если начальное решение φ^0 уже известно, и если мы изменяем или добавляем новые данные, методы сопряжённых градиентов и Гаусса-Зейделя требуют очень немногих итераций для

обновления интерполяции. Непосредственно получаем значения в узлах сетки [20].

Единственный способ для классических методов учитывать разрывы – это разбить функцию таким образом, чтобы разрывы отбрасывались на границах некоторых участков. Этот подход не решает всех проблем; например, могут возникнуть серьёзные трудности, такие как: выбор границ участков, выбор данных для интерполяции на конкретном участке, непрерывность интерполяции на общей границе двух соседних участков, особенно если эта граница не соответствует реальному разрыву, что часто приводит к появлению артефактов.

Однако DSI учитывает разрывы, не разделяя функцию, и таким образом избегает этих проблем. Метод DSI работает значительно быстрее, чем классические методы.

2 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ И ЭКСПЕРИМЕНТЫ

2.1 Программная реализация метода DSI

Программная реализация метода DSI была выполнена на языке программирования Python. Используются такие библиотеки как:

- `numpy` для выполнения математических операций, работы с многомерными массивами, построения сеток;
- `matplotlib.pyplot` для визуализации поверхностей в трехмерном пространстве;
- `mpl_toolkits.mplot3d` также для трехмерной визуализации;
- `matplotlib.cm` для работы с цветовыми картами, окрашивающими точки поверхности, в соответствии с их расположением относительно оси OZ;
- `matplotlib.colors.Normalize` для нормализации данных перед отображением цветом на графиках.

В коде используется регулярная сетка 100×100 .

Тестовые данные генерируются в `generate_data`, это набор точек (x, y, z) с возможным нарушением гладкости (разломом) поверхности, которое зависит от выбранного режима `Mode`. Цель генерации – построение искусственной поверхности с заданной структурой разлома и параметрами, позволяющими контролировать сложность интерполяционной задачи. Для каждого режима добавляется геометрическое описание разломов в виде отрезков. Эти данные сохраняются в списке `faults` и используются: для визуализации разрывов; при построении системы интерполяции – для учета того, что значения на разных сторонах разлома не следует сглаживать между собой.

Для визуализации входных данных функция `visualize_input_data`. Описываются различные виды разломов для создания синтетических данных. Виды разломов, которые описаны: поверхность без разломов; поверхность с полным разломом, выходящим из центра к краю; поверхность с замкнутым

разломом в центре; поверхность с множественными разломами; поверхность с разломом, заданным ломаной линией. Также неравномерно выбираются фиксированные точки в качестве предполагаемых данных со скважин. Разломы изображаются как часть вертикальной плоскости или нескольких, если разлом задан ломаной линией.

Для применения численного метода поверхность дискретизируется с помощью регулярной прямоугольной сетки. Создание сетки выполняется функцией `create_regular_grid`, которая: определяет прямоугольник, охватывающий все входные точки; добавляет к нему отступ (`padding`) 20% по каждой оси; равномерно делит его на ячейки с учетом желаемой плотности (параметр `grid_size`).

На выходе формируются массивы координат узлов сетки по осям x и y : `grid_x` и `grid_y`.

Для визуализации результатов в коде реализована функция `visualize_results`. Так как алгоритм хорошо подходит для геологических поверхностей, которые характеризуются небольшими изменениями, а не резкими скачками, то добавляются поля, но визуализируется только рабочая область. Это делается для того, чтобы исключить возможные выбросы, потому что они могут происходить не из-за дефектов в алгоритме, а из-за резкого обрыва в данных. Отображается центральная часть с достоверной интерполяцией.

Ключевым шагом является формирование линейной системы уравнений, описывающей интерполяцию поверхности, она строится с учетом: исходных данных, разметки разломов, гладкости интерполяции.

Для того, чтобы правильно построить матрицу необходимо заранее определить, какие пары соседних узлов находятся по разные стороны от разрыва, это реализовано через маски разломов. Для этого используется функция `segments_intersect`, проверяющая, пересекаются ли два отрезка в плоскости.

После построения матрицы системы и вектора правой части необходимо решить систему линейных уравнений:

$$Ax = b. \quad (2.1)$$

Для корректного учета разломов необходимо определить, пересекает ли отрезок между соседними узлами сетки какую-либо из линий разломов. Для этого реализована функция `segments_intersect`, которая проверяет наличие пересечения двух отрезков на плоскости, используя параметрическое представление и метод определения пересечения отрезков по направленным площадям. Это позволяет точно установить, какие связи между узлами должны быть исключены при формировании гладкостных условий.

Для решения симметричной положительно определённой системы используется метод сопряжённых градиентов. Он обеспечивает высокую эффективность при разреженных системах, к числу которых относится сформированная матрица A . Метод минимизирует функционал энергии невязки на каждом шаге, направляя поиск вдоль сопряжённых направлений. Остановка осуществляется при достижении малой нормы невязки или при превышении максимального числа итераций.

Альтернативным подходом является метод Гаусса-Зейделя, простой в реализации и устойчивый при выполнении условий диагонального преобладания. Этот метод итеративно уточняет значения решения, используя уже обновлённые значения на текущем шаге. Оценка ошибки между итерациями позволяет контролировать сходимость, а накопление значений ошибки даёт возможность построить график сходимости метода.

Оба метода реализованы и протестированы. Метод сопряжённых градиентов показал более быструю сходимость при больших размерностях задачи, в то время как Гаусс-Зейдель применим для наглядного анализа поведения решения.

Таким образом, для каждого случая на вход получаем сгенерированные данные, которые представляют собой поверхность с разрывом или несколькими. Разрывы отмечаются красной линией, далее выбирается

количество точек, предполагаемых данных со скважин. Эти точки расположены неравномерно. Для того, чтобы оценить результат интерполяции можно сравнить визуализацию входных данных и полученную поверхность.

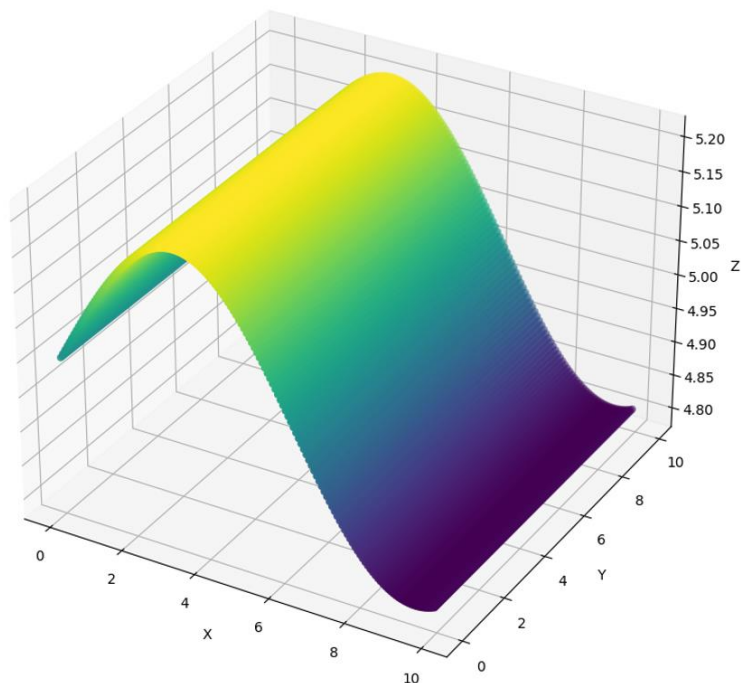


Рисунок 2.1 – Поверхность без разлома

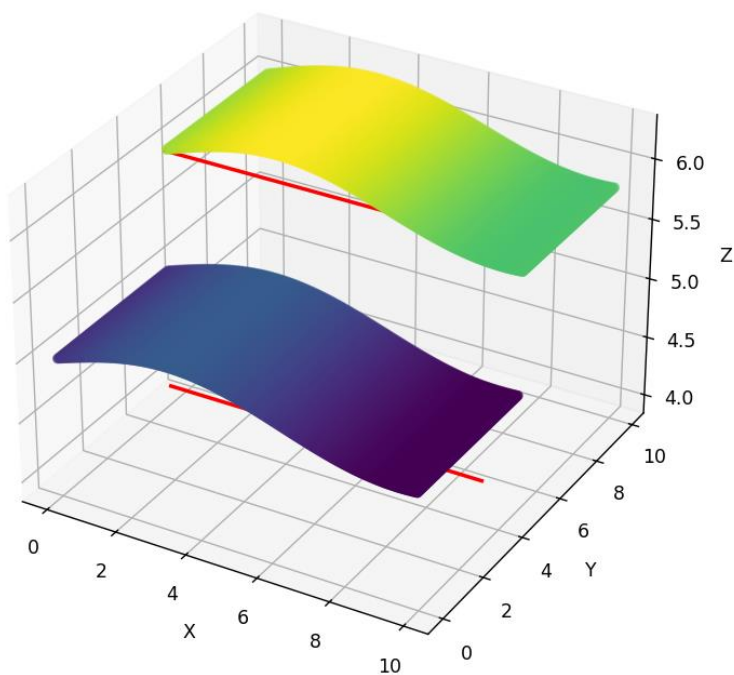


Рисунок 2.2 – Поверхность с полным разрывом

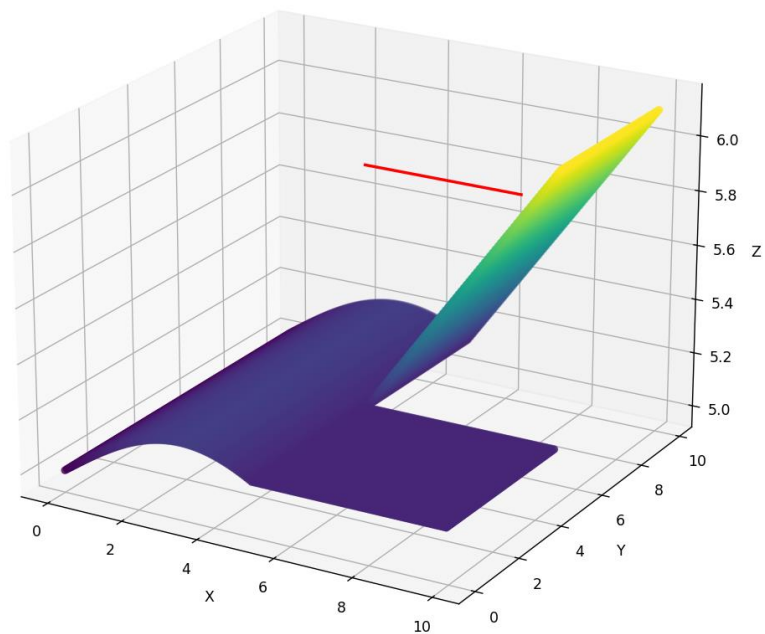


Рисунок 2.3 – Поверхность с разрывом, выходящим из центра к краю

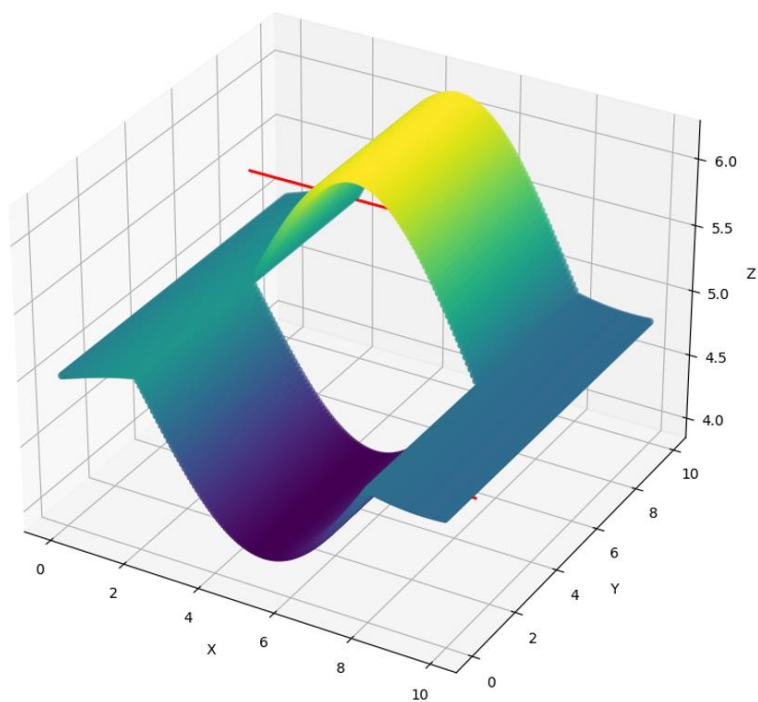


Рисунок 2.4 – Поверхность с замкнутым разрывом в центре

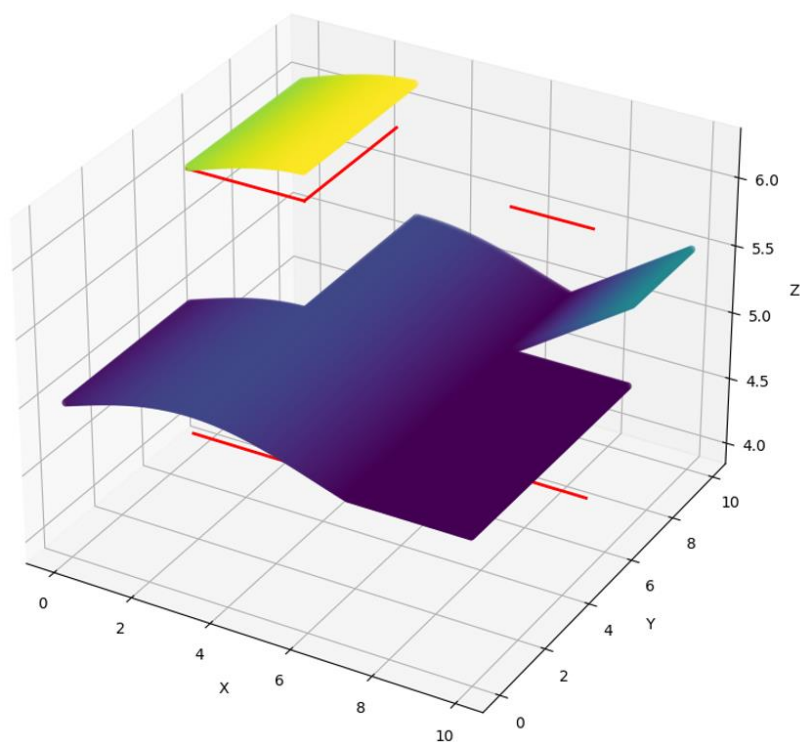


Рисунок 2.5 – Поверхность с несколькими разрывами

Ранее, в основном, все разломы были обозначены одним или двумя отрезками, в следующем примере входных данных структура разлома усложнилась, и задается как ломаная линия.

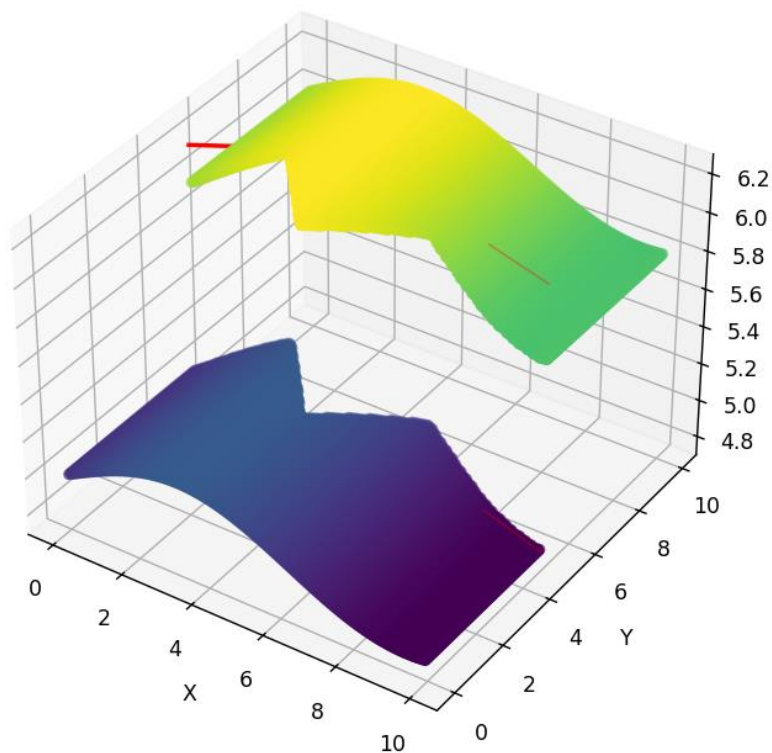


Рисунок 2.6 – Поверхность с разломом в форме ломаной линии

2.2 Результаты моделирования

Результатами моделирования являются интерполированные поверхности методом гладкой дискретной интерполяции с учетом разрывных нарушений.

Для оценки интерполяции каждого вида геометрической интерпретации разлома построены трехмерные графики поверхности с разным количеством точек. Наибольшее – 550, меньшее – 100. Очевидно, что чем больше точек (предполагаемых данных по скважинам) мы используем тем лучше и глаже результат, но и при небольшом количестве заданных точек мы получаем неплохой результат. Только теперь поверхность может не проходить через некоторые точки, потому что образуются перепады в значениях гораздо существеннее, чем были ранее при большем количестве данных. А метод DSI рассчитан на плавные изменения, поэтому хорошо подходит для интерполяции геологических поверхностей.

На графиках поверхности отображаются с использованием цветовой карты, отражающей значения функции. Исходные точки для наглядности выделены красным и распределены в случайном порядке. Разломы представляют собой вертикальные полупрозрачные сегменты плоскости или несколько таких частей, если разлом задан ломаной.

При такой визуализации хорошо видно насколько точно метод DSI восстанавливает поверхность, и есть ли несоответствия.

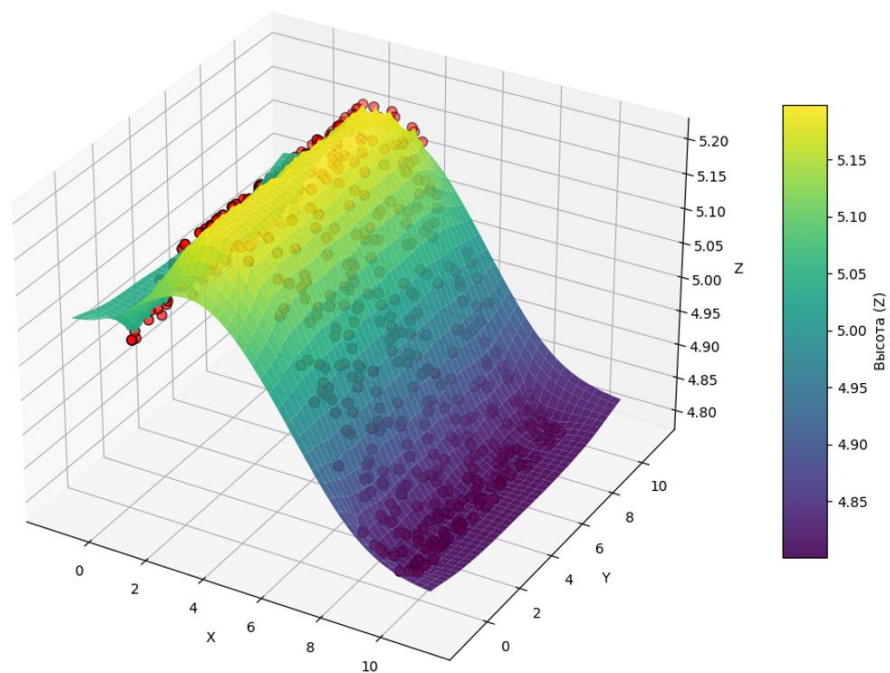


Рисунок 2.7 – Интерполированная поверхность без разлома, 550 точек

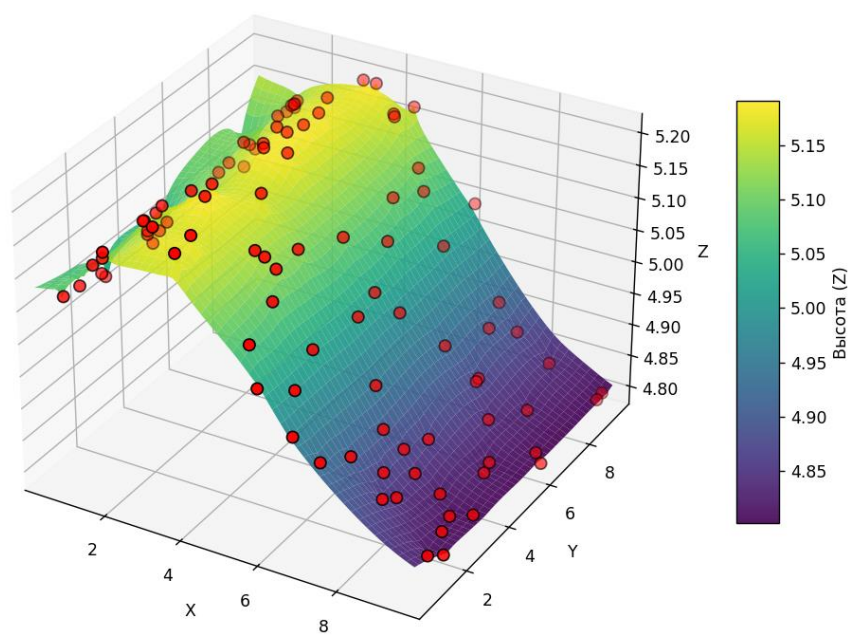


Рисунок 2.8 – Интерполированная поверхность без разлома, 100 точек

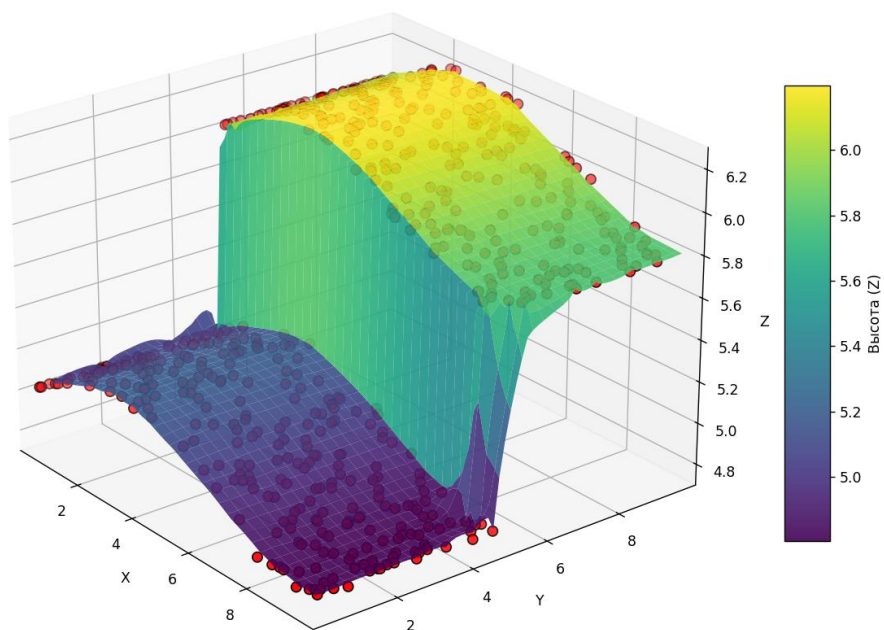


Рисунок 2.9 – Интерполированная поверхность с полным разломом, 550 точек

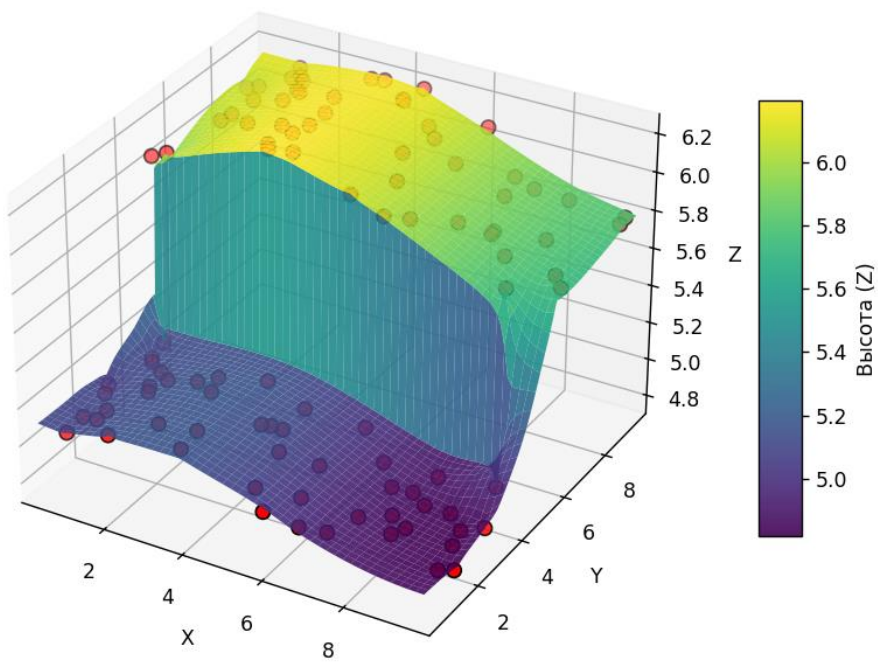


Рисунок 2.10 – Интерполированная поверхность с полным разломом, 100 точек

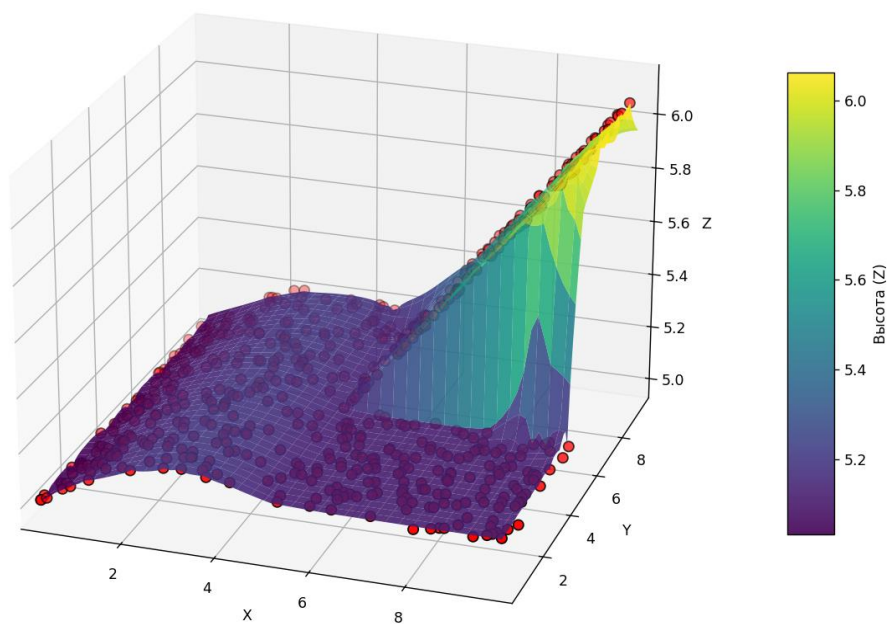


Рисунок 2.11– Интерполированная поверхность с неполным разломом, выходящим из центра к краю, 550 точек

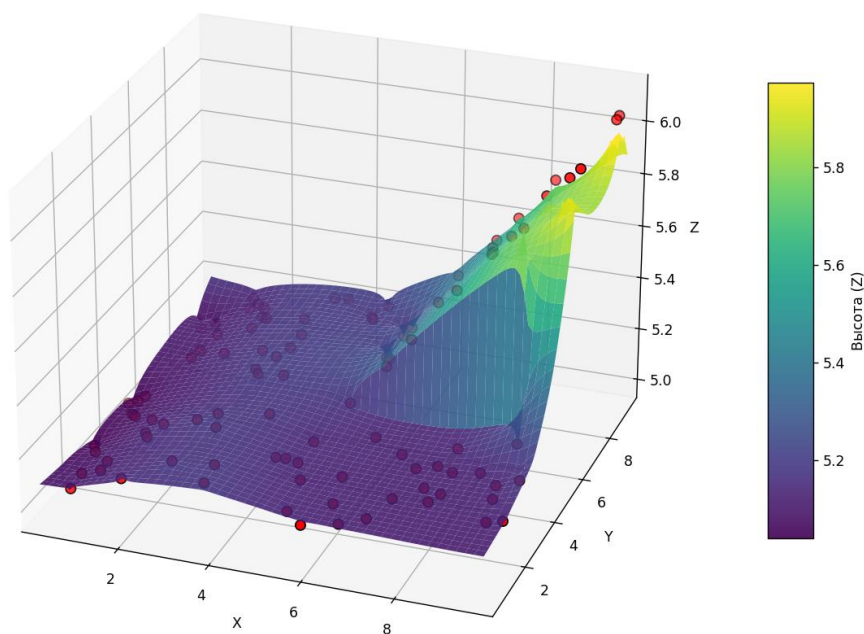


Рисунок 2.12 – Интерполированная поверхность с неполным разломом, выходящим из центра к краю, 100 точек

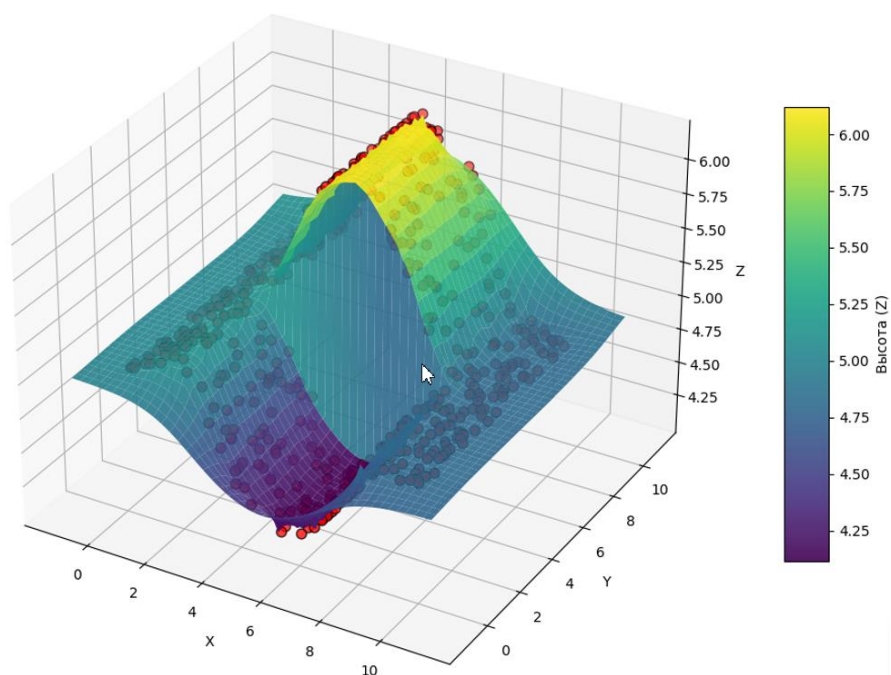


Рисунок 2.13 – Интерполированная поверхность с замкнутым разломом в центре, 550 точек

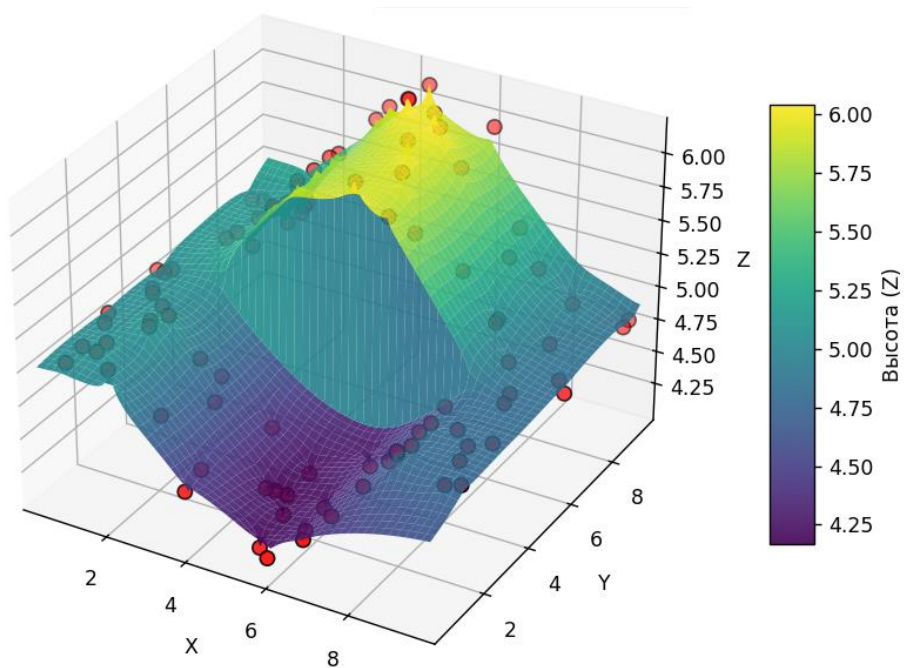


Рисунок 2.14 – Интерполированная поверхность с замкнутым разломом в центре, 100 точек

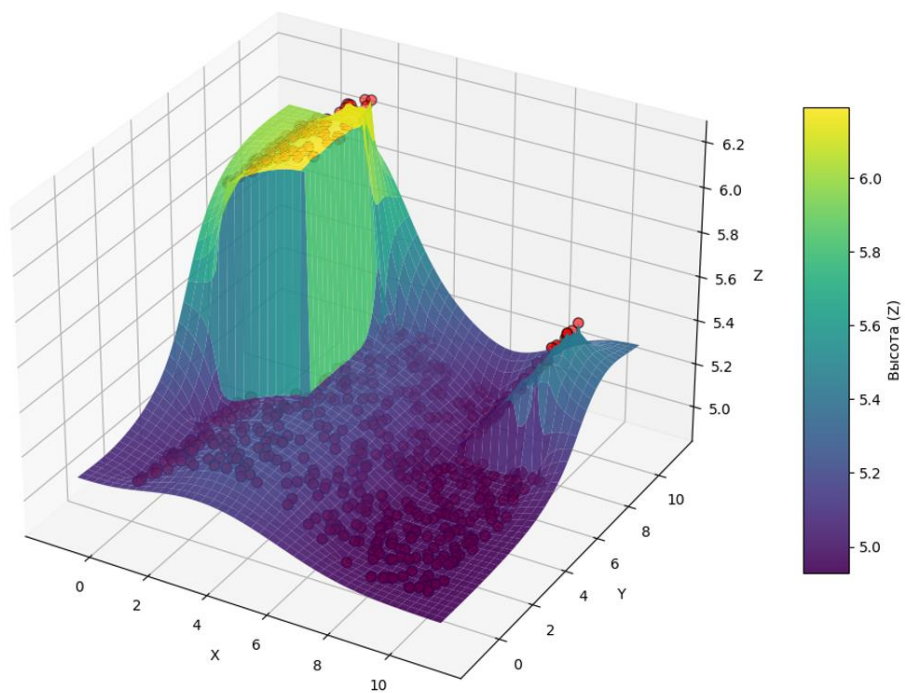


Рисунок 2.15 – Интерполированная поверхность с несколькими разломами,
550 точек

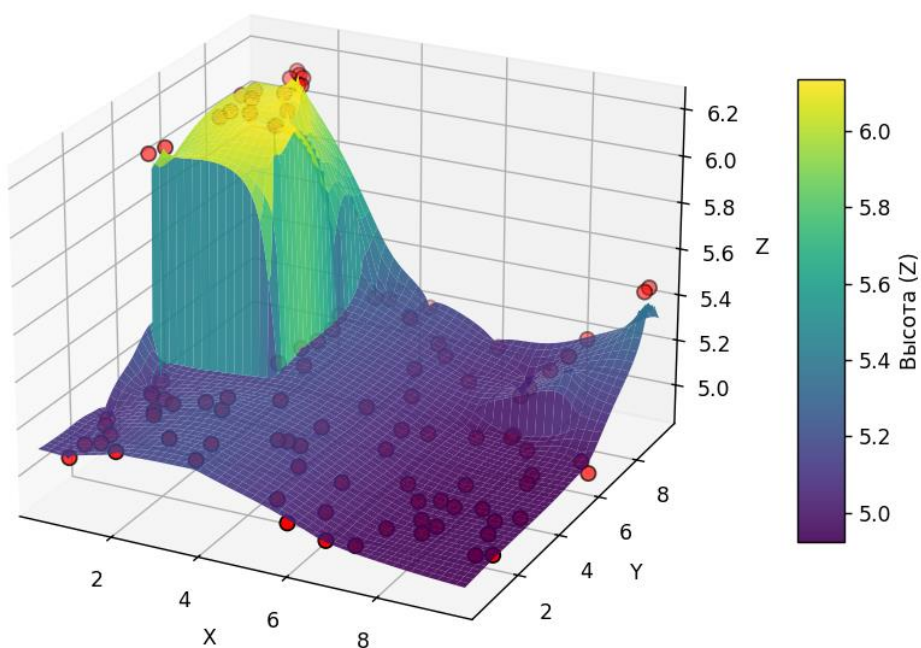


Рисунок 2.16 – Интерполированная поверхность с несколькими разломами,
100 точек

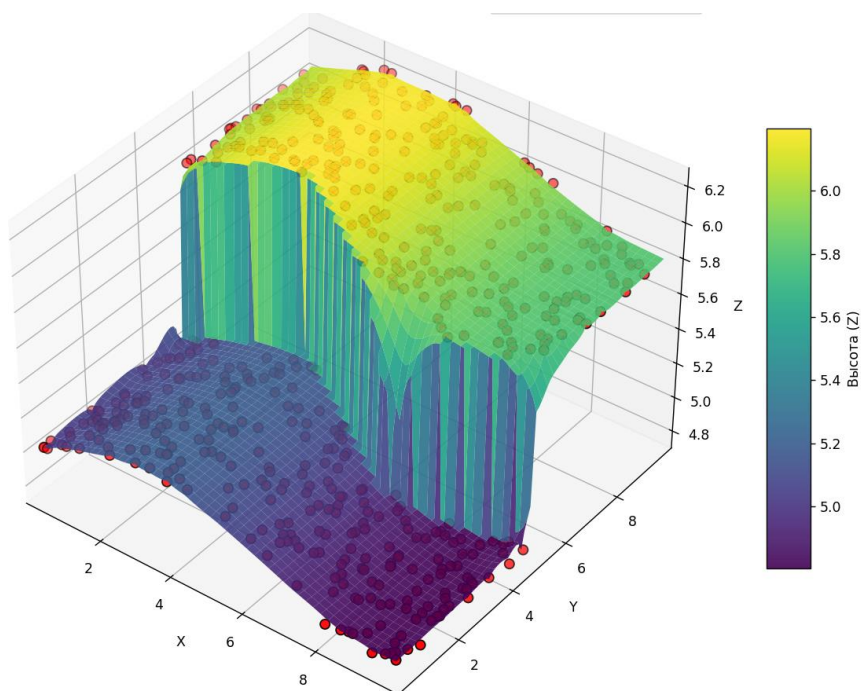


Рисунок 2.17 – Интерполированная поверхность с разломом, заданным ломаной линией, 550 точек

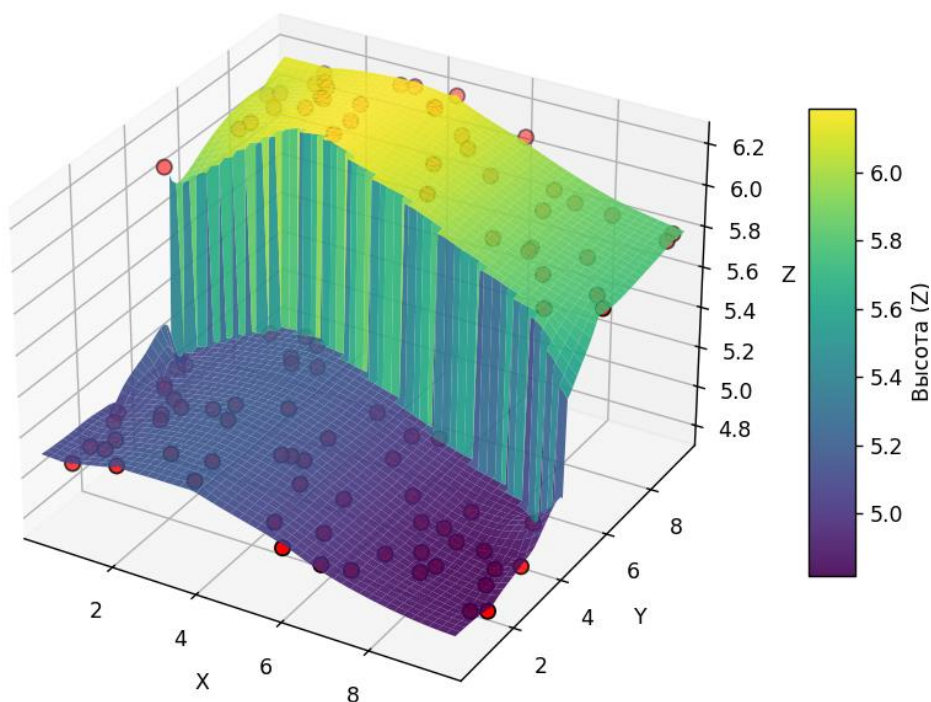


Рисунок 2.18 – Интерполированная поверхность с разломом, заданным ломаной линией, 100 точек

В процессе итерационного решения системы уравнений методом сопряженных градиентов отслеживается невязка, характеризующая, насколько текущее приближенное решение удовлетворяет системе. Для количественной оценки используется квадрат нормы невязки:

$$\|r_k\|^2 = r_k^T r_k = (b - Ax_k)^T (b - Ax_k). \quad (2.2)$$

По мере роста числа итераций квадрат нормы невязки монотонно убывает (с незначительными колебаниями) достигая значения порядка 10^{-6} – 10^{-9} . Это подтверждает устойчивость метода сопряженных градиентов для данной задачи. Также еще был реализован метод Гаусса-Зейделя, но он требовал большего количества итераций и хуже сходил.

Оценка качества интерполяции проводилась для различного количества исходных точек 100 и 550 точек. В каждом случае рассчитывались среднее значение всех ошибок, медиана, стандартное отклонение и максимальное отклонение. Это позволило проанализировать влияние плотности исходных данных на точность восстановления поверхности.

Для всех расчётов использовался метод сопряжённых градиентов с максимальным числом итераций 10 000.

Для количественной оценки результатов для каждой точки вычислялась абсолютная ошибка:

$$\varepsilon_i = |z_i - z'_i|, \quad (2.3)$$

где z'_i – значение в ближайшем узле регулярной сетки, z_i – исходное фиксированное значение в точке.

Таблица 2.1 – Статистические характеристики ошибок интерполяции

Виды поверхности, количество точек	Среднее	Стандартное отклонение	Медиана	Максимум
Интерполированная поверхность без разлома, 550 точек	0.0100	0.0776	0.0001	0.9023
Интерполированная поверхность без разлома, 100 точек	0.0083	0.0065	0.0070	0.0363

Продолжение таблицы 2.1

Виды поверхности, количество точек	Среднее	Стандартное отклонение	Медиана	Максимум
Интерполированная поверхность с полным разломом, 100 точек	0.0031	0.0030	0.0022	0.0156
Интерполированная поверхность с неполным разломом, выходящим из центра к краю, 550 точек	0.0047	0.0515	0.0001	0.8101
Интерполированная поверхность с неполным разломом, выходящим из центра к краю, 100 точек	0.0037	0.0029	0.0031	0.0117
Интерполированная поверхность с замкнутым разломом в центре, 550 точек	0.0096	0.0908	0.0000	1.3951
Интерполированная поверхность с замкнутым разломом в центре, 100 точек	0.0020	0.0016	0.0015	0.0074
Интерполированная поверхность с несколькими разломами, 550 точек	0.0042	0.0450	0.0001	0.6988
Интерполированная поверхность с несколькими разломами, 100 точек	0.0038	0.0033	0.0025	0.0149
Интерполированная поверхность с разломом, заданным ломаной линией, 550 точек	0.0029	0.0222	0.0001	0.2578
Интерполированная поверхность с разломом, заданным ломаной линией, 100 точек	0.0029	0.0023	0.0026	0.0115

При сравнении результатов для 550 и 100 точек наблюдается, что снижение количества исходных данных не приводит к резкому ухудшению точности. Наиболее стабильные и точные результаты получены при полном

линейном разломе и разломе, заданном ломаной линией. Наиболее сложными для интерполяции являются замкнутые разломы, особенно в центре области.

Были проведены эксперименты с разным количеством точек и различными видами разрывов. Метод DSI сохраняет резкие изменения значений функции вдоль линии разлома без сглаживания, что очень важно для геологического моделирования поверхности. На участках без разрывных нарушений поверхность остается гладкой, что соответствует поведению пласта. Преимущество данного метода в том, что он не сглаживает разрывы, как это происходит в большинстве классических методов.

Заключение

Разломы и другие тектонические нарушения существенно влияют на форму и свойства геологических пластов. Игнорирование этих особенностей приводит к ошибкам в моделировании, снижению точности оценки запасов и увеличению экономических рисков при разработке месторождений.

Разломы в геологическом моделировании представляются как ломанные линии, разделяющие область на части с разной топологией. Традиционные методы – линейная интерполяция, сплайны, метод обратных расстояний, кригинги и радиальные базисные функции – предполагают непрерывность и гладкость поверхности. В присутствии разломов они сглаживают скачкообразные изменения, что искажает реальную геологическую структуру и снижает качество моделирования.

Метод DSI обеспечивает локальное измерение шероховатости и минимизацию глобального критерия гладкости с учётом известных значений функции и разрывов. Это позволяет сохранять топологию разломов, обеспечивая гладкость только на непрерывных участках поверхности. DSI является эффективным инструментом для восстановления геологических поверхностей с разрывными нарушениями.

В данной работе проведена практическая реализация и анализ метода дискретной гладкой интерполяции (DSI) для восстановления геологических поверхностей с учётом разрывных нарушений. Создана модель, учитывающая особенности разломов и обеспечивающая гладкую интерполяцию значений на дискретной сетке.

Входные данные включают координаты фиксированных точек, значения функции в них и описание разломов, что позволяет адекватно моделировать сложные геологические структуры.

Использованы эффективные алгоритмы формирования локальных и глобальных критериев шероховатости, а также численные методы решения задачи оптимизации.

Представлены трёхмерные графики визуализации исходных данных и интерполированных поверхностей, демонстрирующие сохранение топологии разломов и плавность на непрерывных участках.

Визуализация фиксированных точек и разломов способствует наглядному контролю корректности моделирования.

Вычисление ошибок интерполяции в фиксированных точках позволяет количественно оценить точность восстановления поверхности. Анализ динамики ошибок по итерациям подтверждает сходимость и стабильность метода DSI. Метод демонстрирует преимущества по сравнению с классическими методами, которые сглаживают разрывы и теряют важную геологическую информацию.

Полученные результаты моделирования позволят повысить точность оценки запасов в пласте. Метод DSI может быть интегрирован в существующие системы геологического моделирования, улучшая качество принятия решений на этапах разведки и разработки месторождения.

Список использованных источников

1. Академия наук СССР. Геотектоника : монография / В. В. Тихомиров, И. Г. Малахова, В. Е. Хаин, В. В. Чекунов, В. В. Сологуб [и др.] ; Академия наук СССР. – Москва : Наука, 1984. – 275 с.
2. Бондаренко, В. М. Интерпретация геофизических данных : учебное пособие / В. М. Бондаренко, Е. Е. Лумпов, А. А. Лыхин. – Москва : МГГА, 1993. – 114 с.
3. Гаврилов, В. П. Геодинамический анализ нефтегазоносных бассейнов : учебник / В. П. Гаврилов, Ю. И. Галушкин. – Москва : Издательский дом «Недра», 2010. – 227 с. – ISBN 978-5-8365-0359-8.
4. Дахнов, В. Н. Интерпретация результатов геофизических исследований разрезов скважин / В. Н. Дахнов. – Москва : Недра, 1982. – 448 с. – УДК 550.83 (075).
5. Закревский, К. Е., Попов, В. Л. История развития трехмерного геологического моделирования как метода изучения залежей нефти и газа / К. Е. Закревский, В. Л. Попов // Известия Томского политехнического университета. Инжиниринг георесурсов. – 2021. – Т. 332, № 5. – С. 89–100.
6. Киссин, И. Г. (ред.) Флюиды в земной коре : геофизические и тектонические аспекты / И. Г. Киссин (ред.). – Москва : Наука, 2009. – 328 с. – УДК 551.1. – ISBN 978-5-02-036655-8.
7. Метод интерполяции геологической среды в условиях фрагментарно заданных данных / А. И. Кобрунов, Е. Н. Мотрюк, Д. О. Ломинский // Фундаментальные исследования. – 2015. – № 2-24. – С. 5340–5345.
8. Методические рекомендации к корреляции разрезов скважин / под ред. И. С. Гутмана. – Москва : Недра, 1985. – 120 с.
9. Основы геологического моделирования : пособие для студентов и магистров / В. А. Белкина, С. Р. Бембель, А. А. Забоева, Н. В. Санькова. – Тюмень : Тюменский государственный нефтегазовый университет, 2015. – 168 с.

10. Пушаровский, Ю. М. Тектоническое развитие земной коры и разломы / Ю. М. Пушаровский, А. Л. Яншин ; под ред. Ю. М. Пушаровского. – Москва : Наука, 1979. – 275 с.
11. Разрывные нарушения. Элементы разлома // Studfile.net : сайт. – URL: <https://studfile.net/preview/9794958/page:16/> (дата обращения: 08.06.2025).
12. Briggs, I. C. Machine contouring using minimum curvature / I. C. Briggs // *Geophysics*. – 1974. – Vol. 39, No. 1. – P. 39–48.
13. Caumon, G. Surface restoration as a means to characterize transverse fault slip uncertainty / G. Caumon, P. Muron // 23rd Gocad Meeting : materials. – Nancy, France, 2006.
14. de Boor, C. A Practical Guide to Splines / C. de Boor. – New York : Springer, 2001. – 357 с.
15. Deutsch, C. V. GSLIB: Geostatistical Software Library and User's Guide / C. V. Deutsch, A. G. Journel. – New York : Oxford University Press, 1998. – 369 с.
16. Fernández, O. A new approach for the 3-D reconstruction of geological surfaces: an example of growth strata and turbidite systems from the Ainsa basin (Pyrenees, Spain) / O. Fernández, J. A. Muñoz, P. Arbués, O. Falivene, M. Marzo // *AAPG Bulletin*. – 2004. – Vol. 88, № 8. – P. 1049–1067.
17. Fossen, H. Structural Geology / H. Fossen. – Cambridge : Cambridge University Press, 2016. – 510 с.
18. Mallet, J.-L. Discrete smooth interpolation / J.-L. Mallet // *ACM Transactions on Graphics*. – 1989. – Vol. 8, No. 2. – P. 121–144. – Centre de Recherche en Informatique de Nancy and Ecole Nationale Supérieure de Géologie.
19. Mallet, J.-L. Geomodeling / J.-L. Mallet. – Oxford : Oxford University Press, 2002. – 600 с.
20. Matheron, G. Les variables régionalisées et leur estimation : une application de la théorie des fonctions aléatoires aux sciences de la nature / G. Matheron. – Paris : Masson et Cie, 1965. – 306 с.

21. Matheron, G. Spline and kriging, their formal equivalence / G. Matheron // Geological Contributions (Syracuse). – 1981. – No. 8. – P. 77–85.
22. Pouzet, J. Estimation of a Surface with Known Discontinuities for Automatic Contouring Purposes / J. Pouzet // Math. Geol. – 1980. – Vol. 12, No. 6. – P. 559–570.
23. Sambridge, M., Braun, J. Discrete smooth interpolation in geophysical problems / M. Sambridge, J. Braun // Geophysical Journal International. – 1995. – Vol. 122, Iss. 3. – P. 837–857.
24. Schuh-Senlis, M. Structural Restoration of Geological Structures With Viscous Stokes Flow - Principle and First Results / M. Schuh-Senlis, C. Thieulot, P. Cupillard, G. Caumon // 82nd EAGE Annual Conference & Exhibition : materials / European Association of Geoscientists and Engineers. – 2021. – Vol. 2021. – P. 1–5.
25. Schuh-Senlis, M. What does it take to restore geological models with «natural» boundary conditions? / M. Schuh-Senlis, G. Caumon, P. Cupillard // Solid Earth. – 2024. – Vol. 15. – P. 945–965. – DOI: 10.5194/se-15-945-2024.
26. Shepard, D. A two-dimensional interpolation function for irregularly-spaced data / D. Shepard // Proceedings of the 1968 23rd ACM National Conference. – 1968. – P. 517–524.
27. Wellmann, F. 3-D Structural geological models: Concepts, methods, and uncertainties / F. Wellmann, G. Caumon // Advances in Geophysics. – 2018. – Vol. 59. – P. 1–121. – DOI: 10.1016/bs.agph.2018.09.001.
28. Wendland, H. Scattered Data Approximation / H. Wendland. – Cambridge: Cambridge University Press, 2005. – 348 c.

Приложение А

(обязательное)

Программная реализация задачи интерполяции методом DSI

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from mpl_toolkits.mplot3d.art3d import Poly3DCollection
from matplotlib.cm import ScalarMappable
from matplotlib.colors import Normalize
grid_size=(100, 100)
Lambda = 1
CrMode = 0
# тип разлома
# 0 – поверхность без разлома
# 1 – поверхность с полным разломом
# 2 – поверхность с разломом, который выходит из центра к краю
# 3 – поверхность с замкнутым разломом в центре
# 4 – поверхность с несколькими разломами
# 5 – поверхность с разломом в виде ломаной
UseSinus = True
def visualize_input_data(points, faults):
    """
    Визуализация входных данных:
    - Точки с цветами по высоте
    - Разломы, как полупрозрачные плоскости
    """
    fig = plt.figure(figsize=(12, 8))
    ax = fig.add_subplot(111, projection='3d')
    zmin = points[:, 2].min()
```

```

zmax = points[:, 2].max()
# Цвета точек по высоте
norm = Normalize(vmin=zmin, vmax=zmax)
colors = plt.cm.viridis(norm(points[:, 2]))
sc = ax.scatter(points[:, 0], points[:, 1], points[:, 2],
                c=colors)

# Разломы, как плоскости
for fault in faults:
    # Создаем полигон разлома (вертикальная плоскость)
    x = fault[:, 0]
    y = fault[:, 1]
    z_bottom = np.full_like(x, zmin) # Нижняя граница
    z_top = np.full_like(x, zmax) # Верхняя граница
    # Визуализация плоскости разлома
    verts = [list(zip(x, y, z_bottom)), list(zip(x, y, z_top))]
    ax.add_collection3d(Poly3DCollection(verts, alpha=0.3, color='red'))
    # Контур разлома
    ax.plot(x, y, z_bottom, 'r-', linewidth=2)
    ax.plot(x, y, z_top, 'r-', linewidth=2)
    # Цветовая шкала
cbar = fig.colorbar(ScalarMappable(norm=norm, cmap='viridis'), ax=ax)
cbar.set_label('Высота (Z)')
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('Входные данные: точки и разломы')
plt.tight_layout()
plt.show()

def visualize_results(grid_x, grid_y, solution, points):
    """

```

Визуализация результатов интерполяции:

- Только область с исходными точками
- Фиксированные точки выделены красным
- Без отображения пустых полей

"""

```
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')

# Определяем границы области с данными
sz = len(points)
x_min, x_max = points[:, 0].min(), points[:, 0].max()
y_min, y_max = points[:, 1].min(), points[:, 1].max()
# Добавляем 10% отступ от крайних точек
padding_x = int(grid_size[0] * 0.2) + 1
padding_y = int(grid_size[1] * 0.2) + 1
plot_x = grid_x[padding_x:-padding_x]
plot_y = grid_y[padding_y:-padding_y]
plot_Z = solution.reshape(len(grid_x), len(grid_y))[padding_x:-padding_x,
padding_y:-padding_y].T

# Визуализация поверхности
X, Y = np.meshgrid(plot_x, plot_y)
surf = ax.plot_surface(X, Y, plot_Z, cmap='viridis', alpha=0.9)

# Исходные точки (красные)
ax.scatter(points[:, 0], points[:, 1], points[:, 2],
           c='red', s=50, edgecolor='k', label='Фиксированные точки')

# Настройки графика
ax.set_xlim(x_min, x_max)
ax.set_ylim(y_min, y_max)
ax.set_xlabel('X')
ax.set_ylabel('Y')
```

```

ax.set_zlabel('Z')
ax.set_title('Результаты интерполяции (только область с данными)')
# Цветовая шкала
fig.colorbar(surf, shrink=0.6, aspect=10, label='Высота (Z)')
plt.legend()
plt.tight_layout()
plt.show()

def generate_data(n=100, Regular=False, Mode=0, UseSinus=True):
    """Генерация гладких тестовых данных"""
    np.random.seed(42)
    if not Regular:
        # Случайные точки в прямоугольной области 10x10
        x = np.random.uniform(0, 10, n)
        y = np.random.uniform(0, 10, n)
        R = 1
    else:
        x = np.zeros(n * n)
        y = np.zeros(n * n)
        for X in range(n):
            for Y in range(n):
                x[X * n + Y] = 10 * X / n
                y[X * n + Y] = 10 * Y / n
        n = n * n
        R = 1
    z = np.empty((n,))
    # Центр области
    center_x, center_y = 5, 5
    FaultRadius = 3
    FaultHeight = 1
    faults = []

```

```

# Генерируем зигзаг только для Mode 5
if Mode == 5:
    # Параметры зигзага
    num_segments = np.random.randint(3, 6) # 3-5 сегментов
    segment_length = 10 / num_segments
    zigzag_points = [(0, center_y)]
    # Генерируем точки зигзага
    for i in range(1, num_segments + 1):
        prev_x, prev_y = zigzag_points[-1]
        new_x = i * segment_length
        # Случайное отклонение по Y ( $\pm 2$  от центра)
        new_y = center_y + np.random.uniform(-2, 2)
        zigzag_points.append((new_x, new_y))
    for i, P in enumerate(zip(x, y)):
        X = P[0]
        Y = P[1]
        CrZ = 5 + (0.2 * np.sin(X / 2) if UseSinus else 0)
        if Mode == 3:
            # Учитываем разлом
            dx = abs(X - center_x) # Расстояние по X от центра
            dy = Y - center_y # Расстояние по Y от центра
            if dx < FaultRadius: # Близко к линии разлома
                # Холм ( $y < center_y$ ) и овраг ( $y > center_y$ )
                if dy > 0:
                    CrZ += FaultHeight * np.sin((FaultRadius - dx) / 2 / FaultRadius
* np.pi)
                else:
                    CrZ -= FaultHeight * np.sin((FaultRadius - dx) / 2 / FaultRadius
* np.pi)

```

```

elif Mode == 2:
    dx = X - center_x # Расстояние по X от центра
    dy = Y - center_y # Расстояние по Y от центра
    if dx > 0: # Близко к линии разлома
        # Холм (y < center_y) и овраг (y > center_y)
        if dy > 0:
            CrZ = 5 + (0.2 * np.sin(center_x / 2) if UseSinus else 0) +
FaultHeight * dx / center_x
        else:
            CrZ = 5 + (0.2 * np.sin(center_x / 2) if UseSinus else 0)
elif Mode == 1:
    dy = Y - center_y # Расстояние по Y от центра
    if dy > 0: # Близко к линии разлома
        # Холм (y < center_y) и овраг (y > center_y)
        CrZ += FaultHeight
elif Mode == 4:
    dx = X - center_x - 2 # Расстояние по X от центра
    dy = Y - center_y - 2 # Расстояние по Y от центра
    if dx > 0: # Близко к линии разлома
        # Холм (y < center_y) и овраг (y > center_y)
        if dy > 0:
            CrZ = 5 + (0.2 * np.sin((center_x + 2) / 2) if UseSinus else 0) +
FaultHeight * dx / center_x
        else:
            CrZ = 5 + (0.2 * np.sin((center_x + 2) / 2) if UseSinus else 0)
    dx = X - center_x + 2 # Расстояние по X от центра
    dy = Y - center_y # Расстояние по Y от центра
    if dx < 0 and dy > 0: # Близко к линии разлома
        # Холм (y < center_y) и овраг (y > center_y)
        CrZ += FaultHeight

```

```

elif Mode == 5:
    # Проверяем, находится ли точка справа и сверху от зигзага
    above_zigzag = False

    # Находим сегмент зигзага, в котором находится X-координата
    точки
    for j in range(len(zigzag_points) - 1):
        x1, y1 = zigzag_points[j]
        x2, y2 = zigzag_points[j + 1]
        if x1 <= X <= x2:
            # Уравнение линии между точками (x1,y1) и (x2,y2)
            if x1 == x2: # вертикальный сегмент
                zigzag_y_at_x = min(y1, y2)
            else:
                slope = (y2 - y1) / (x2 - x1)
                zigzag_y_at_x = y1 + slope * (X - x1)
            # Если точка выше зигзага в этом сегменте
            if Y > zigzag_y_at_x:
                above_zigzag = True
                break
        if above_zigzag:
            CrZ += FaultHeight
    z[i] = CrZ

    # Добавляем линии разломов для визуализации
    if Mode == 1:
        faults.append(np.array([[0, center_y, 0], [2 * center_x - 1, center_y, 0]]))
    elif Mode == 2:
        faults.append(np.array([[center_x, center_y, 0], [2 * center_x - 1,
center_y, 0]]))

```

```

elif Mode == 3:
    faults.append(np.array([[center_x - FaultRadius, center_y, 0], [center_x +
FaultRadius, center_y, 0]]))
elif Mode == 4:
    faults.append(np.array([[0, center_y, 0], [center_x - 2, center_y, 0]]))
    faults.append(np.array([[center_x - 2, center_y, 0], [center_x - 2, 2 *
center_y - 1, 0]]))
    faults.append(np.array([[center_x + 2, center_y + 2, 0], [2 * center_x - 1,
center_y + 2, 0]]))
elif Mode == 5:
    # Добавляем все сегменты зигзага как линии разломов
    for j in range(len(zigzag_points) - 1):
        x1, y1 = zigzag_points[j]
        x2, y2 = zigzag_points[j + 1]
        faults.append(np.array([[x1, y1, 0], [x2, y2, 0]]))
    return np.column_stack([x, y, z]), faults
# 3. Функция создания регулярной сетки
def create_regular_grid(points):
    x_pad = (points[:, 0].max() - points[:, 0].min()) * 0.2
    y_pad = (points[:, 1].max() - points[:, 1].min()) * 0.2
    grid_x = np.linspace(points[:, 0].min() - x_pad,
        points[:, 0].max() + x_pad,
        int(grid_size[0]*1.4)+1)
    grid_y = np.linspace(points[:, 1].min() - y_pad,
        points[:, 1].max() + y_pad,
        int(grid_size[1]*1.4) + 1)
    return grid_x, grid_y
# 4. Функция построения матриц системы
def build_matrices(points, faults, grid_x, grid_y):
    nx, ny = len(grid_x), len(grid_y)
    n = nx * ny

```

```

A = np.zeros((n, n))
b = np.zeros(n)
# 1. Создаем маску разломов (True если между узлами есть разлом)
fault_mask_x = np.zeros((nx, ny), dtype=bool) # Для вертикальных
разломов
fault_mask_y = np.zeros((nx, ny), dtype=bool) # Для горизонтальных
разломов
# Заполняем маски разломов
for fault in faults:
    for k in range(len(fault) - 1):
        x1, y1, _ = fault[k]
        x2, y2, _ = fault[k + 1]
        # Определяем все пары соседей, разделенных этим сегментом
разлома
        for i in range(nx - 1):
            for j in range(ny - 1):
                # Проверяем пересекает ли линия между соседями разлом
                if segments_intersect(
                    grid_x[i], grid_y[j], grid_x[i + 1], grid_y[j],
                    x1, y1, x2, y2):
                    fault_mask_x[i, j] = True
                if segments_intersect(
                    grid_x[i], grid_y[j], grid_x[i], grid_y[j + 1],
                    x1, y1, x2, y2):
                    fault_mask_y[i, j] = True
# 2. Учет известных значений
known_mask = np.zeros(n, dtype=bool)
for point in points:
    i = np.argmin(np.abs(grid_x - point[0]))
    j = np.argmin(np.abs(grid_y - point[1]))
    idx = i * ny + j

```

```

A[idx, idx] = 1
b[idx] = point[2]
known_mask[idx] = True
# 3. Условия гладкости (исключая связи через разломы)
for i in range(nx):
    for j in range(ny):
        idx = i * ny + j
        if known_mask[idx]:
            continue
        neighbors = []
        # Проверяем всех возможных соседей
        if i > 0 and not fault_mask_x[i - 1, j]:
            neighbors.append((i - 1) * ny + j) # left
        if i < nx - 1 and not fault_mask_x[i, j]:
            neighbors.append((i + 1) * ny + j) # right
        if j > 0 and not fault_mask_y[i, j - 1]:
            neighbors.append(i * ny + (j - 1)) # up
        if j < ny - 1 and not fault_mask_y[i, j]:
            neighbors.append(i * ny + (j + 1)) # down
        # Формируем уравнение лапласиана только с допустимыми
соседями
        if neighbors:
            A[idx, idx] = len(neighbors) * Lambda
            for neighbor in neighbors:
                A[idx, neighbor] = -1 * Lambda
    return A, b

def segments_intersect(x1, y1, x2, y2, x3, y3, x4, y4):
    # Реализация алгоритма проверки пересечения отрезков
    denom = (y4 - y3) * (x2 - x1) - (x4 - x3) * (y2 - y1)
    if denom == 0:

```

```

        return False
    ua = ((x4 - x3) * (y1 - y3) - (y4 - y3) * (x1 - x3)) / denom
    if ua < 0 or ua > 1:
        return False
    ub = ((x2 - x1) * (y1 - y3) - (y2 - y1) * (x1 - x3)) / denom
    if ub < 0 or ub > 1:
        return False
    return True

# 5. Метод сопряженных градиентов
def conjugate_gradient(A, b, max_iter=10000, tol=1e-9):
    n = len(b)
    x = np.zeros(n)
    r = b - A.dot(x)
    p = r.copy()
    rs_old = r.dot(r)

    for i in range(max_iter):
        Ap = A.dot(p)
        alpha = rs_old / p.dot(Ap)
        x += alpha * p
        r -= alpha * Ap
        rs_new = r.dot(r)
        if np.sqrt(rs_new) < tol:
            break
        if i % 10 == 9:
            print(i, rs_new)
        beta = rs_new / rs_old
        p = r + beta * p
        rs_old = rs_new

    return x

```

```

def gauss_seidel(A, b, max_iter=1000, tol=1e-6):
    #Решение системы  $Ax = b$  методом Гаусса-Зейделя
    n = len(b)
    x = np.zeros(n)
    errors = []
    for k in range(max_iter):
        x_old = x.copy()
        for i in range(n):
            sigma = 0
            for j in range(n):
                if j != i:
                    sigma += A[i, j] * x[j]
            if A[i, i] != 0:
                x[i] = (b[i] - sigma) / A[i, i]
        # Вычисление ошибки
        error = np.linalg.norm(x - x_old)
        errors.append(error)
        # Вывод прогресса каждые 100 итераций
        print(f'Итерация {k}, ошибка: {error:.2e}')
        if error < tol:
            print(f'Сошлось за {k} итераций')
            break
    return x, errors

def plot_2d_projections(grid_x, grid_y, solution, points):
    fig = plt.figure(figsize=(16, 6))
    Z = solution.reshape(len(grid_x), len(grid_y)).T
    X, Y = np.meshgrid(grid_x, grid_y)
    ax1 = fig.add_subplot(121)
    profile_y = np.mean(Z, axis=1)

```

Y')

```
ax1.plot(grid_y, profile_y, 'b-', linewidth=2, label='Средний профиль по
ax1.set_xlabel('Y координата')
ax1.set_ylabel('Z (высота)')
ax1.set_title('Вид сбоку (Y-Z проекция)')
ax1.grid(True)
ax1.legend()
ax2 = fig.add_subplot(122)
norm = Normalize(vmin=np.min(Z), vmax=np.max(Z))
im = ax2.pcolormesh(X, Y, Z, cmap='viridis', shading='auto', norm=norm)
# Исходные точки
for pt in points:
    # Находим ближайший узел сетки
    i = np.argmin(np.abs(grid_x - pt[0]))
    j = np.argmin(np.abs(grid_y - pt[1]))
    grid_z = Z[j, i] # Значение интерполяции в этой точке
ax2.set_xlabel('X координата')
ax2.set_ylabel('Y координата')
ax2.set_title('Вид сверху (X-Y проекция) с отклонениями точек')
cbar = fig.colorbar(im, ax=ax2)
cbar.set_label('Высота (Z)')
handles, labels = ax2.get_legend_handles_labels()
by_label = dict(zip(labels, handles))
ax2.legend(by_label.values(), by_label.keys(), loc='upper right')

plt.tight_layout()
plt.show()
def calculate_point_errors(points, grid_x, grid_y, interpolated_values):
    errors = []
    nearest_points = []
```

```

Z_grid = interpolated_values.reshape(len(grid_x), len(grid_y))
for pt in points:
    # Находим ближайший узел сетки
    i = np.argmin(np.abs(grid_x - pt[0]))
    j = np.argmin(np.abs(grid_y - pt[1]))
    # Получаем значение в узле сетки
    grid_z = Z_grid[i, j]
    # Вычисляем разницу
    errors.append(pt[2] - grid_z)
    nearest_points.append([grid_x[i], grid_y[j], grid_z])
return np.abs(np.array(errors)), np.array(nearest_points)

if __name__ == "__main__":
    print("1. Генерация входных данных...")
    points, faults = generate_data(150, True, CrMode)
    print("2. Визуализация входных данных...")
    visualize_input_data(points, faults)
    points, faults = generate_data(100, False, CrMode)
    print("3. Создание регулярной сетки...")
    grid_x, grid_y = create_regular_grid(points)
    print("4. Построение матриц системы...")
    A, b = build_matrices(points, faults, grid_x, grid_y)
    #gauss_seidel(A, b)
    print("5. Решение системы методом сопряженных градиентов...")
    #Гарантируем, что матрица всегда симметричная и положительно
определенная
    A_sym = A.T @ A
    b_sym = A.T @ b
    solution = conjugate_gradient(A_sym, b_sym)
    print("6. Визуализация результатов...")
    visualize_results(grid_x, grid_y, solution, points)

```

```

plot_2d_projections(grid_x, grid_y, solution, points)
errors, nearest_grid_points = calculate_point_errors(
    points, # Исходные точки [x,y,z]
    grid_x, grid_y, # Координаты сетки
    solution # Интерполированные значения
)
m = points[:,2].max()- points[:,2].min()
print("Статистика ошибок:")
print(f"Среднее: {100*np.mean(errors)/m:.4f}")
print(f"Стандартное отклонение: {100*np.std(errors)/m:.4f}")
print(f"Медиана: {100*np.median(errors)/m:.4f}")
print(f"Максимум: {100*np.max(errors)/m:.4f}")
m = points[:,2].max()- points[:,2].min()

```