Теория и практика анализа океанологических данных

**1 Неделя**

***Предисловие***

Данный курс направлен на оптимизацию процесса обработки данных в сфере наук о Земле. В рамках этого семестра мы познакомимся с современными форматами хранения информации, научимся проводить пред- и обработку сырых данных, познакомимся с методикой получения этих данных. Теоретическая часть курса также будет включать в себя необходимую информацию в океанологии о течениях, волновых процессах, разномасштабной циркуляции океана и другие знания, необходимые для первичной обработки данных.

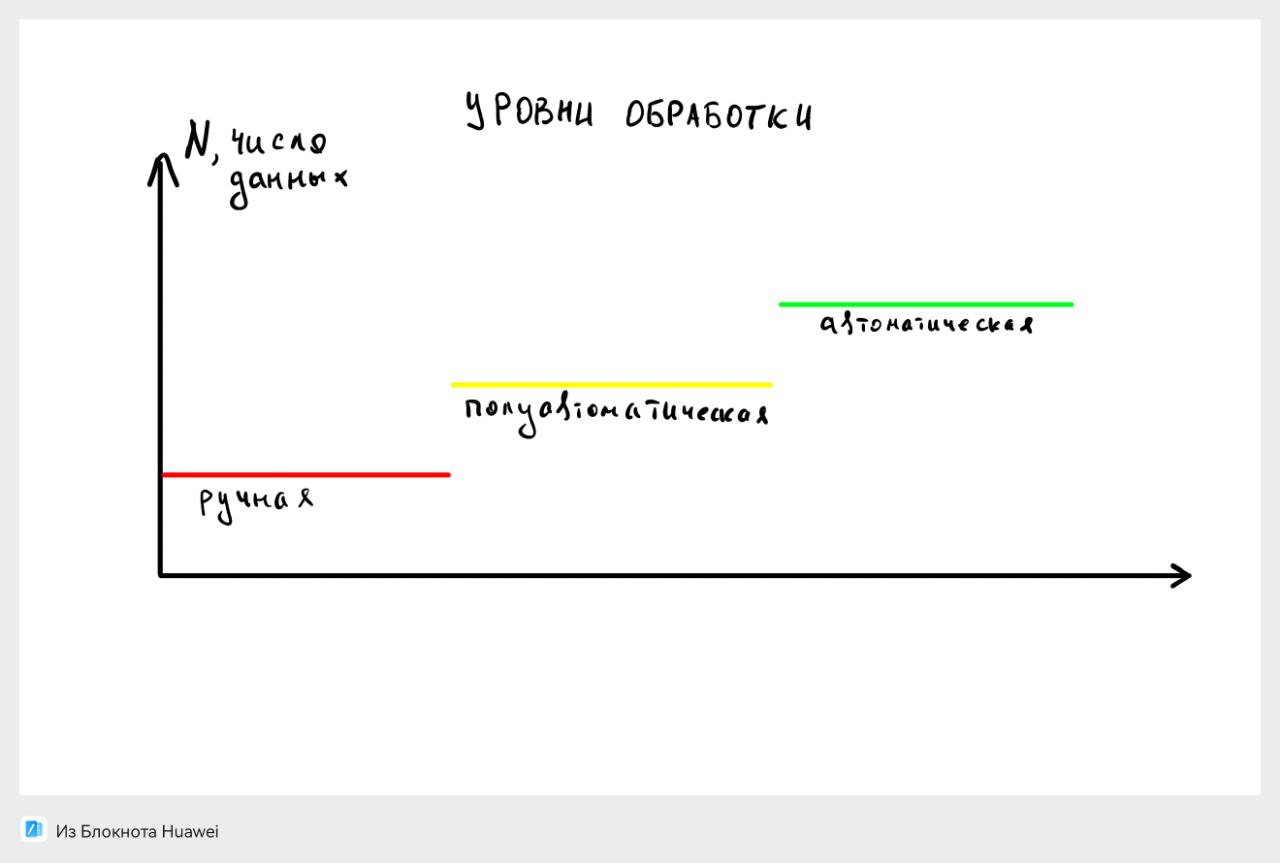
Помимо теории, в рамках курса мы организуем 2 однодневные экспедиции, где вы сами попробуете организовать и провести измерения на льду и на открытой воде. Далее полученные данные будут вами обработаны и представлены на итоговом занятии.

Таким образом, семестр будет разбит на 3 части, которые разделены 2 практиками, ориентировочно 2 марта и 27 апреля. В первой части курса мы поговорим о хранении данных на компьютере, о типах данных: как их создавать и просматривать. Затронем вопрос открытости информации и ответим на вопрос, какие сейчас существуют открытые базы данных в океанологии. Вторая часть курса будет посвящена введению в океанологию, базовые понятия и процессы. Научимся визуализировать реальные данные методами matplotlib и cartopy. Познакомимся с различными реанализами, распространёнными моделями, спутниковыми данными и натурными измерениями. В третьей части курса вы полностью будете обрабатывать данные ваших измерений изученными методами. Перед вами будут поставлены небольшие реальные физические задачи и вопросы, ответ на которые предстоит дать, основываясь на обработанных данных.

Полученные знания должны сильно упростить написание дипломных работ.

***Об уровнях обработки информации***

Прежде всего стоит рассказать про сам процесс обработки экспериментальных данных. В нашем рассказе мы будем разделять обработку данных на 3 различных типа: ручной, полуавтоматический и автоматический. Их можно проиллюстрировать как 3 различных уровня (рис. 1). Поясним каждый из них.



*Рисунок 1. Визуализация уровней обработки данных*

Процессу ручной обработки данных нас учат в школе и на первых курсах института. Он подразумевает перебор всех измеренных значений для и вычисления последующих характеристик. Приведём пример простого эксперимента. Нам необходимо определить среднюю температуру в комнате. Для этого мы используем простой термометр и измеряем температуру в 10 разных местах помещения и записываем к себе в блокнот. Далее достаём калькулятор, складываем все числа и делим на их число. Мы провели *ручную* обработку полученных данных.

Следующий уровень – полуавтоматический процесс обработки. Вооружившись продвинутым калькулятором мы также 10 раз измеряем температуру и записываем эти значения уже не в блокнот, а в запоминающее устройство нашего калькулятора. Далее мы просим наш компьютер: рассчитай мне среднее. Устройство само рассчитывает среднее на основе запомненных значений и выдаёт ответ.

И самый верхний уровень, автоматическая обработка. Представим, что теперь комната у нас не одна, а 100, а точек для расчёта нужно брать не 10, а 1000. И ответ нам нужен не 1, а 100. Тут на помощь нам приходит более сложная система, которая включает в себя термометр и отдельный компьютер для обработки. Каждый раз проводя измерения мы указываем комнату и компьютер теперь знает, в каком помещении проводилось измерение. Далее в программе на расчётном компьютере мы рассчитываем среднее для каждой комнаты. Важным отличием от полуавтоматического метода является универсальность. Единожды написав программу для этой задачи, мы запросто можем провести измерение ещё в 100 комнатах без дополнительных усилий в обработке.

Важным преимуществом каждого последующего уровня является его бóльшая универсальность. Именно этот факт является наиболее важным при оценке метода обработки в широком смысле. Но можно возразить: при стремлении обработать данные температуры в 1 комнате полностью автоматическим методом мы потратим гораздо больше времени на написание программы, чем если бы мы использовали калькулятор или проводили расчёты в блокноте. Да, поэтому на автоматический метод обработки данных нужно переходить, когда данных становится много. Зачастую не стоит тратить часы на написание программы, если гораздо эффективнее будет ручная обработка. Однако в науках о Земле есть существенное отличие от вышеприведённого примера: данных *очень* много, и они с каждым днём только множатся. Именно поэтому изучение автоматического метода обработки является необходимым сейчас.

**Тема 1. Хранение информации**

*«Зачем мне что-то запоминать, когда я могу легко посмотреть это в книге» (Альберт Эйнштейн).*

После того, как физическая величина измерена, её необходимо запомнить. Память в компьютере представляет собой длинную последовательность ячеек, в которую можно положить либо 0, либо 1. Люди давно ушли от того, чтобы проводить запись в двухзначном формате, поэтому процесс записи любой информации на компьютере происходит не напрямую, а через центральный процессор (ЦП). Именно он переводит все символы в машинный код и производит сохранение.

Один из самых распространённых форматов данных, который используется для сохранения информации это .txt (сокр. text) и .csv (от англ. Comma Separated Values). Оба формата напрямую кодируют текстовые символы и переводят в двоичный формат, а затем складывают в память. Поскольку не привлекается никаких сторонних программ для кодировки информации, а используются стандартные (UTF-8/16, ANSI), то такие данные можно просматривать простыми текстовыми редакторами. Формат .csv является тем же .txt, который является таблицей с разделителем (по умолчанию – запятая).

Простота использования вышеупомянутых форматов кроет в себе один очень важный недостаток. Из-за того, что происходит прямая кодировка текста без сжатия, при записи нескольких сотен тысяч строк (что далеко не редкость в науках о Земле) объём такого файла будет очень большой, соответственно и процесс автоматической обработки будет кратно замедлен.

В далёком 1990 году, метеорологи из Университетской корпорации атмосферных исследований (UCAR) попытались решить ряд вышеупомянутых недостатков и создали специальный формат хранения данных, ориентированный на использование и хранение массиво-ориентированной научной информации. Такой формат получил название NetCDF (Network Common Data Form) и имеет расширение «.nc». Сейчас им кодируют практически все данные дистанционного зондирования Земли (ДЗЗ), результаты работ моделей и реанализов.

Доступ для записи и чтения к файлам программных библиотек NetCDF осуществляется через сторонние интерфейсы (напрямую через программный код или же через дополнительные приложения). Таким образом нельзя просматривать закодированную информацию напрямую, как это можно осуществить с форматом «.txt». Для Python написано несколько различных библиотек для работы с файлами NetCDF: netCDF4, scipy.io.