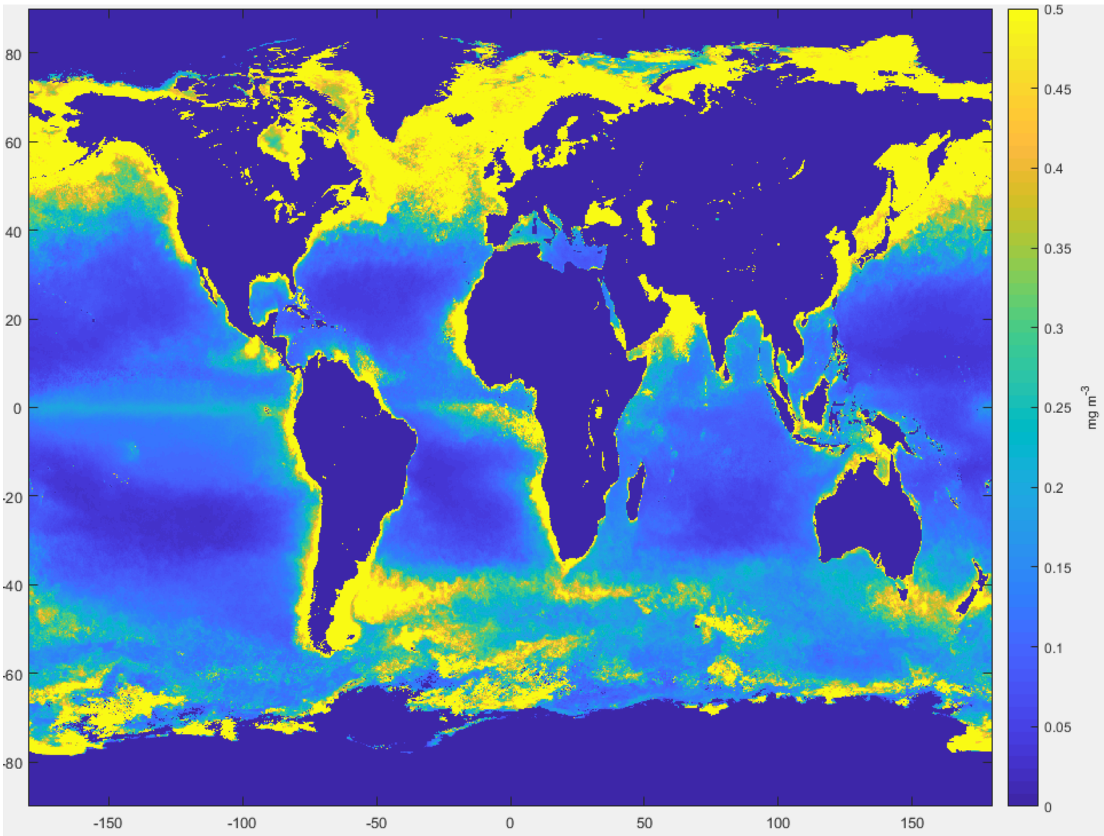
А.С. Маюрова, Е.А. Быковская, Е.П. Тюрикова,

И.В. Тимофеева

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ДИСЦИПЛИНЕ «СИТУАЦИОННОЕ МОДЕЛИРОВАНИЕ»



Санкт-Петербург

2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

А.С. Маюрова, Е.А. Быковская, Е.П. Тюрикова, И.В. Тимофеева

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ДИСЦИПЛИНЕ «СИТУАЦИОННОЕ МОДЕЛИРОВАНИЕ»

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО
по направлению подготовки 20.04.01 Техносферная безопасность
в качестве Учебно-методическое пособие для реализации основных профессиональных образовательных программ высшего образования магистратуры

ИТМО

Санкт-Петербург
2023

А.С. Маюрова, Е.А. Быковская, Е.П. Тюрикова, И.В. Тимофеева, Методические указания по дисциплине «Ситуационное моделирование»– СПб: Университет ИТМО, 2022. – 59 с.

Рецензент: Тамбулатова Екатерина Викторовна, кандидат технических наук, , декан факультета экотехнологий, Университета ИТМО.

Учебно-методическое пособие предназначено для поддержки практических занятий и самостоятельной работы студентов в рамках курса «Ситуационное моделирование» с использованием пакета MatLab. Данный курс относится к дисциплинам профильного профессионального модуля учебного плана магистерской программы. При освоении дисциплины значительное внимание уделяется практическим занятиям, в рамках которых рассматриваются модели, описывающие взаимоотношения природы и общества и рассказывается о применении пакета MatLab для анализа многомерных данных.

Университет ИТМО - ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО - участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО - становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

Содержание

Введение

В предлагаемом пособии представлен теоретический материал и перечень практических заданий по различным темам дисциплины «Ситуационное моделирование» для основных профессиональных образовательных программ высшего образования магистратуры по направлению подготовки 20.04.01 «Техносферная безопасность» в части освоения инструментария пакета MatLab для решения широкого спектра задач, являющихся основой компьютерного моделирования.

Многомодульная структура MatLab может стать основным инструментом исследования как в учебном, так и научно-исследовательском процессе, а предлагаемое пособие обеспечивает освоение его основ. Оперативный и интерактивный характер взаимодействия пользователя и системы MatLab позволяет экстраполировать функционал пакета на решение прикладных, оптимизационных задач и других задач более высокого уровня.

Данное учебно-методическое пособие может быть весьма полезно для решения научно-технических задач, возникающих при работе над курсовыми и дипломными проектами.

Большинство практических работ данного учебно-методического пособия рассчитаны на создание студентами некоторой программы (кода), решающей определенные задачи и графически отображающей результаты решения. Практические задания связанны с оценкой состояния окружающей среды и характером влияния на нее человеческой деятельности, в том числе с анализом климатических характеристик, видового разнообразия и гидрологических моделей.

Кейс 1. Переменные, матрицы и составление транспортного потока

Задачи:

1. Задание значений переменных, создание матрицы в MatLab, присвоение значений и импорт данных.
2. Выполнение простых арифметических действий с матрицами в MatLab.
3. Графическое отображение зависимостей.

ЧАСТЬ 1. ПЕРЕМЕННЫЕ И МАТРИЦЫ

Необходимо задать значения некоторых переменных.

Можно задать единичное значение переменной, например:

```
x=5
```

```
y=2
```

Задание 1. Используя MatLab, задайте переменные:

```
x=0.75
```

```
y=0.36
```

Большая часть данных в MatLab хранится в виде матриц разного вида. Задача на сегодня – научиться составлять матрицы в MatLab, есть несколько способов это сделать.

Первый способ – назначить каждое из значений вручную. Можно просто присвоить значения всем элементам матрицы. Например, необходимо чтобы наша матрица выглядела так:

$$A = \begin{bmatrix} 3 & 5 & 2 \\ 6 & 3 & 9 \end{bmatrix}$$

Можно задать ее вручную командой:

```
A=[3,5,2;6,3,9]
```

В итоге получилась матрица с 6 элементами, она имеет 2 строки и 3 столбца.

Размер матрицы 2x3 (в MatLab всегда сначала указывается количество строк, потом количество столбцов), матрица всегда называется с заглавной буквы. Вручную матрицы задаются таким образом, как будто вслух произносим значения матрицы (слева направо, затем сверху вниз) с запятой или пробелом между двумя элементами в одной строке и точкой с запятой, разделяющей строки.

Задание 2. Используя MatLab, задайте матрицу следующего вида:

$$B = \begin{bmatrix} 3 & 2 \\ 4 & 27 \\ 23 & 9 \\ 93 & 3 \end{bmatrix}$$

Задание 3. Используя MatLab, задайте матрицу следующего вида:

$$C = [7 \quad 9]$$

Задание 4. Используя MatLab, задайте матрицу следующего вида:

$$D = \begin{bmatrix} 7 \\ -9 \end{bmatrix}$$

Второй способ составления матриц – это автоматическая их генерация. Для длинных матриц вводить их значения вручную слишком долго, поэтому можно автоматизировать этот процесс, но тоже несколькими способами.

Можно использовать оператор «:», например:

```
E = 5:8
```

$$E = [5 \quad 6 \quad 7 \quad 8\,]$$

По умолчанию оператор «:» задает значения с шагом 1, однако шаг можно тоже задать вручную, например:

```
F=20:2:26
```

$$F = [20 \quad 22 \quad 24 \quad 26\,]$$

Задание 5. Используя MatLab, задайте матрицу следующего вида:

$$G = [1 \quad 4 \quad 7 \;\; 10 \ldots 100\,]$$

Сколько значений получилось в этой строке?

Если вы знаете количество элементов, которые вам нужны (вместо шага между элементами), то можно использовать функцию `linspace` (первое значение элемента, последнее значение элемента, количество элементов), например:

```
H= linspace(0,1,5)
```

$$H = [0 \quad 0.250 \quad 0.500 \quad 0.750 \quad 1.000\,]$$

Задание 6. Используя MatLab, задайте матрицу следующего вида:

$$G = [0 \quad 15.6667 \quad 31.3333 \quad 47.0000\,]$$

Вы можете создать матрицу случайных чисел, используя команду:

```
K=rand(2)
```

$$K = \begin{bmatrix} 0.8147 & 0.5146 \\ 0.6354 & 0.9852 \end{bmatrix}$$

Число в скобках этой команды задает размер матрицы, в нашем случае это матрица 2x2.

Задание 7. Используя MatLab, задайте матрицу случайных чисел L размера 2x3.

Функция rand создает матрицу случайных чисел в диапазоне от 0 до 1. Если вам нужна матрица случайных чисел в другом диапазоне, то вы можете использовать функцию `randi`, которая позволяет задать диапазон случайных чисел:

`K1=randi(10,3)` – задаст матрицу размером 3x3 с диапазоном случайных чисел от 1 до 10.

$$K1 = \begin{bmatrix} 1 & 5 & 7 \\ 8 & 9 & 1 \\ 2 & 5 & 6 \end{bmatrix}$$

`K2=randi(10,3,2)` – задаст матрицу размером 3x2 с диапазоном случайных чисел от 1 до 10.

$$K2 = \begin{bmatrix} 5 & 6 \\ 7 & 2 \\ 4 & 1 \end{bmatrix}$$

`K3=randi([-10,10],3)` – задаст матрицу размером 3x3 с диапазоном случайных чисел от -10 до 10.

$$K3 = \begin{bmatrix} -5 & -4 & 1 \\ -6 & 3 & 10 \\ 2 & 5 & 6 \end{bmatrix}$$

Задание 8. Используя MatLab, задайте матрицу K4 размера 3x4, состоящую из случайных чисел в диапазоне от -1 до 1.

Также вы можете задать матрицу, состоящую из одних единиц или нулей, используя команды `ones` и `zeros` соответственно.

Задание 9. Используя MatLab, задайте матрицу M размера 3x4, состоящую из единиц, и матрицу N размера 5x2, состоящую из нулей.

Примечание 1. Вы можете использовать функцию `size`*, чтобы узнать размер матрицы, или задать новую матрицу с таким же размером, например* `O=rand(size(N))` *задаст матрицу случайных чисел* `O` *размера 5x2.*

Вы можете узнать значение того или иного числа в матрице и задать его в переменную. Например, давайте зададим переменную `x1`, которая будет равна значению 0.5146 из матрицы `K`.

`x1=K(1,2)` – присваивает `x1` значение из первой строки и второго столбца матрицы `K`.

Также можно задать значение из матрицы, используя оператор `end`, который выдаст последнее значение строки или столбца. Например, код:

`X2=K(end,end)` – выдаст самое последнее значение матрицы `K` – 0.9852

Также вы можете вручную изменить какое-либо значение матрицы. Например, код:

`K(1,2) = 0.5` – поменяет значение в первой строке, втором столбце матрицы `K` с 0.5146 на 0.5

В процессе работы необходимо будет обрабатывать большое количество данных, хранимых в виде матрицы. Однако в одной матрице может находиться информация не только по искомому параметру, но и ряд дополнительных данных – например, температура, цветность и соленость воды за каждый день в течение года. Одна из задач, которую необходимо выполнить – это «вытащить» из подобной матрицы данные только температуры. Это можно сделать, используя уже знакомый оператор «:». Пример:

`P=K(:,1)` – создаст столбец всех элементов первого столбца матрицы `K`.

Представим, что у нас есть другая матрица `K`, которая имеет размер 5x5:

`P1=K(1:4,:)` – создаст матрицу, состоящую из первых четырех строк матрицы `K`.

Задание 10. Используя MatLab, задайте матрицу случайных чисел `Q` размера 3x4, создайте матрицу `Q1`, состоящую из последнего столбца матрицы `Q`.

Итоговое задание по части 1. Используя полученные навыки, постройте матрицу *Results* следующего вида:

$$Results = \begin{bmatrix} 2 & 15 & 1 \\ 4 & 30 & 2 \\ 6 & 45 & 3 \\ \cdots & \cdots & \cdots \\ 100 & 750 & 50 \end{bmatrix}$$

ЧАСТЬ 2. ВЫГРУЗКА ДАННЫХ И ИХ ОТОБРАЖЕНИЕ

Существует большое количество операторов загрузки данных из разных источников. Самые распространенные форматы работы с MatLab - это форматы .txt, .dat, or .csv .xls, .xlsb, .xlsm, .xlsx, .xltm, .xltx, или .ods

Для чтения и открытия файлов есть большое количество функций, например: `load`, `import`, `fread`, `fscanf` – для открытия текстовых файлов, `xlsread`, `readtable` – для открытия табличных данных. Также в MatLab версии 2019 года и выше появилась функция для открытия табличных данных `readmatrix`.

У нас есть файл с данными о глубине залива Монтерей в Калифорнии в течение одного месяца от Национального управления по исследованию океанов и атмосферы Америки (NOAA) - water_level_matrix_file.csv. Обращаем ваше внимание на то, что большая часть климатических данных обычно находится в формате csv. Чтобы посмотреть данные в таких файлах через MatLab, необходимо чтобы файл был в той же папке, где и ваш код. Данные откроются при запуске команды:

```
readtable water_level_matrix_file.csv
```

Чтобы задать данные из файла в матрицу, можно также использовать эту команду:

```
A= readtable ('water_level_matrix_file.csv')
```

Какой размер имеет матрица A?
Первый столбец файла содержит данные о времени за 1/10 часа, второй – уровень воды в футах.

Чтобы построить график, можно использовать функцию `plot (x,y)`. Давайте построим график колебания уровня воды в течение 48 часов:

`plot(A{1:480,1},A{1:480,2})` – где `x` – время, при этом были выбраны только строки с 1 по 480; `y` – уровень воды в этих же строках.
`set(gca,'FontSize',14)`
`xlabel('Time [hr]')` – задает подпись оси `x`
`ylabel('Water level [ft]')` – задает подпись оси `y`

Есть ли какая-то закономерность в этом графике? Можно ли ее как-то объяснить?

Задание 1. Постройте график по данным за месяц, где по оси x отображены дни.

Построим теперь 3D график от функции

$$T(v,p) = p\sin(v) - v\cos(p)$$

Пусть p и v изменяются от -2π до 2π.

```
P=linspace(-2*pi,2*pi);
V=linspace(-2*pi,2*pi);
```

Сейчас обе переменные являются строкой матрицы, чтобы сделать из них распределение, используется команда `meshgrid`, которая создает список массивов координатных сеток N-мерного координатного пространства для указанных одномерных массивов координатных векторов.

```
[v1,p1]=meshgrid(V,P);
```

Пропишем нашу искомую функцию:

```
T=p1.*sin(v1)-v1.*cos(p1);
```

Знак «.» необходим при действиях с матрицами, чтобы эти действия выполнялись верно.

Существуем много различных команд для 3D отображения графика. Например:

```
plot3(v1,p1,T) – построит график из отдельных линий
mesh(v1,p1,T) – построит сплошной график
xlabel('1st Indep Var')
ylabel('2nd Indep Var')
zlabel('Function T')
mesh(first,second,T)
```

ЧАСТЬ 3. ПОСТРОЕНИЕ МОДЕЛИ ТРАНСПОРТНОГО ПОТОКА

Модели транспортных потоков могут напрямую влиять на тип и размещение сигналов управления дорожным движением или привлекать внимание к транспортной инфраструктуре. Математическая модель транспортного потока будет иметь две независимые переменные: пространство и время.

Чтобы создать модели транспортного потока, можно математически представить либо плотность движения (сколько автомобилей на километр дороги), либо скорость (насколько быстро автомобили движутся в километрах в час). Эти два показателя сильно зависят друг от друга, потому что водители принимают решения о своей скорости, основываясь на количестве автомобилей рядом с ними.

Плотность будет представляться как зависимая переменная. Создадим простейшую модель движения на участке однополосной автомагистрали с односторонним движением (обгон запрещен).

Предположим, что каждый автомобиль, въезжающий на нашу магистраль с одного конца, уезжает с другого конца. Переменная `t` представляет время в минутах, переменная `x` означает местоположение на шоссе в километрах.

Предположим, что плотность связана со временем и пространством следующим образом:

$$\rho = \frac{100}{x-2t} sin\left(\frac{x-2t}{3}\right) + 10$$

Задание:

Часть 1. Фиксированное время.

Шаг 1. Установите время `t=0` и постройте график плотности движения как функции пространства `(x)` от 0.5 до 100 километров. Это изначальное распределение плотности машин на шоссе.

Какой вывод можно сделать из этого графика? Где самые загруженные места в начале симуляции? Где дорога пуста?

Шаг 2. Установите `t=10` и постройте график плотности при новых данных. Где теперь самое загруженное место?

Шаг 3. Установите `t=20` и постройте график плотности при новых данных. Используя три графика, опишите как меняется трафик в нашей симуляции.

Часть 2. Фиксированное пространство.

Шаг 1. Установите `x=0` и постройте график плотности движения как функции от времени от 0 до 20 минут. Когда в нулевой точке отсчета больше/меньше всего машин?

Шаг 2. Установите `x=20` и постройте еще один график. Что изменилось? Опишите, как меняется наша модель.

Часть 3. Ничего не фиксировано.

Постройте 3D график плотности потока. Какие выводы из него можно сделать? Насколько эта модель отражает реальность?

Кейс 2. Океанографические данные. Построение графиков, нахождение средних значений данных

Задачи:

1. Выгрузка отдельных данных из Excel
2. Определение среднего значения данных, стандартного отклонения
3. Графическое отображение зависимостей (графики, гистограммы, графики с обратными осями).

ЧАСТЬ 1. ДАННЫЕ О ПРИЛИВАХ

В прошлом занятии нами были использованы команды для того, чтобы загрузить данные из файла сразу в матрицу. В этот раз пойдем другим путем. У нас есть несколько файлов Excel, содержащих данные NOAA об уровне воды на двух станциях (Атлантик-Сити и Кейп-Мэй) и метеорологические данные от станций (Кейп-Мэй, Нью-Джерси). Чтобы импортировать данные напрямую из файла, нужно:

1. перейти во вкладку Home и нажать кнопку Import Data;
2. в открывшимся окне выбрать необходимый файл. В нашем случае это файл tide_wind_data.v2;
3. в новом окне есть три листа с данными, нас интересует первый, где есть три столбца данных. Сверху есть графа Output type. Если вы выберете формат Table, то программа выгрузит вам все три столбца и все строки в одну матрицу размера 100x3. В данный момент нам необходимо выгрузить все три столбца в отдельные вектора данных. Для этого необходимо в графе Output type выбрать Column Vector. Эта команда позволяет автоматически разделить выбранные столбцы в разные вектора;
4. справа нажмите на кнопку Import Selection.

В этом упражнении будем использовать функции `mean`, `max` и `min` для определения среднего значения и диапазона данных.

Команда `mean` выдает среднее значение всех элементов вектора или вектор средних значений каждого из столбца матрицы. Команды max и min выдают максимально и минимальное значение по той же схеме.

Задание 1:

1. Выгрузите данные об уровне воды согласно инструкции выше.
2. Используя функции `mean`, `max` и `min`, найдите среднее, максимальное и минимальное значение ожидаемого уровня воды.

3. Найдите разницу между ожидаемым уровнем воды и реальным, сохраните значения в переменную с названием `difference`.
4. Найдите среднее, максимальное и минимальное значение разницы между уровнями воды.
5. Постройте кривые ожидаемого и реального уровня воды на одном графике, используя функцию `hold on`.
6. Добавьте графику оси, легенду и название, используя команды `xlabel, ylabel, legend, title`.

Задание 2:

1. Повторите те же действия для станции Кейп-Мэй.
2. Какие сходства и отличия у графиков двух станций?

ЧАСТЬ 2. АНАЛИЗ МЕТЕОРОЛОГИЧЕСКИХ ДАННЫХ

Во втором разделе поработаем с некоторыми метеорологическими характеристиками, такими как скорость и направление ветра, температура воздуха и атмосферное давление, а также построим розу ветров.

Построим несколько более сложных графиков. Для этого нам понадобится функции `figure` (позволяет создать окно, где будут отображаться графики) и `subplot`, позволяющая построить несколько графиков в одном окне. Например, код:

`subplot(2,1,1);` - где 2 означает количество графиков по вертикали (количество строк), 1 – количество графиков по горизонтали (количество столбцов) и последняя единица – описание какого из графиков приводится далее.

```
x = linspace(0,10);
y1 = sin(x);
plot(x,y1)

subplot (2,1,2);
y2 = sin(5*x);
plot(x,y2)
```

позволит получить вам следующие графики в одном окне:

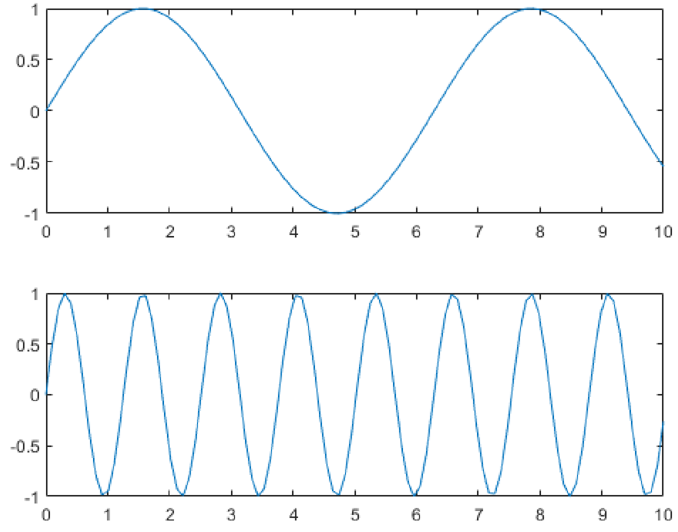


Рисунок 2.1 – Визуализация данных с помощью команды `subplot`

Задание 1:

1. Выгрузите метеорологические данные из того же файла (третья вкладка) в разные вектора.
2. Откройте часть кода (matlab_script_wind_data.v2), которая находится в папке.
3. До включения программы ответьте на вопросы:
   - Сколько графиков будет изображено?
   - Какие данные там будут отображены?
   - Какие команды кода вам неизвестны?
   - Как вам кажется, что они делают?
4. Включите программу. Что делают команды, которые вам неизвестны?
5. Были ли какие-то значительные изменения в скорости ветра? Как менялось давление и температура при изменении скорости ветра?

Задание 2:

С помощью команды: «`doc название команды`» посмотрите, как работают команды `compass` и `rose`. Постройте в одном окне два графика: график векторов (команда `compass`) и график направлений ветра (команда `rose`).

ЧАСТЬ 3. АНАЛИЗ ДАННЫХ ВЕРТИКАЛЬНОГО СРЕЗА

В этой части рассмотрим зависимость температуры воды и ее солености от глубины.

Задание:

1. Выгрузите данные из файла ctd_data по векторам.
2. Постройте два графика в одном окне: график зависимости температуры воды от глубины и график зависимости солености воды от глубины.
3. Теперь постройте те же два графика в новом окне, но переведите единицы измерения глубины из футов в метры (1 фут = 0.3048 метра).
4. Постройте график таким образом, чтобы ось глубины (y) представляла собой реальный срез воды (была перевернута по значениям):

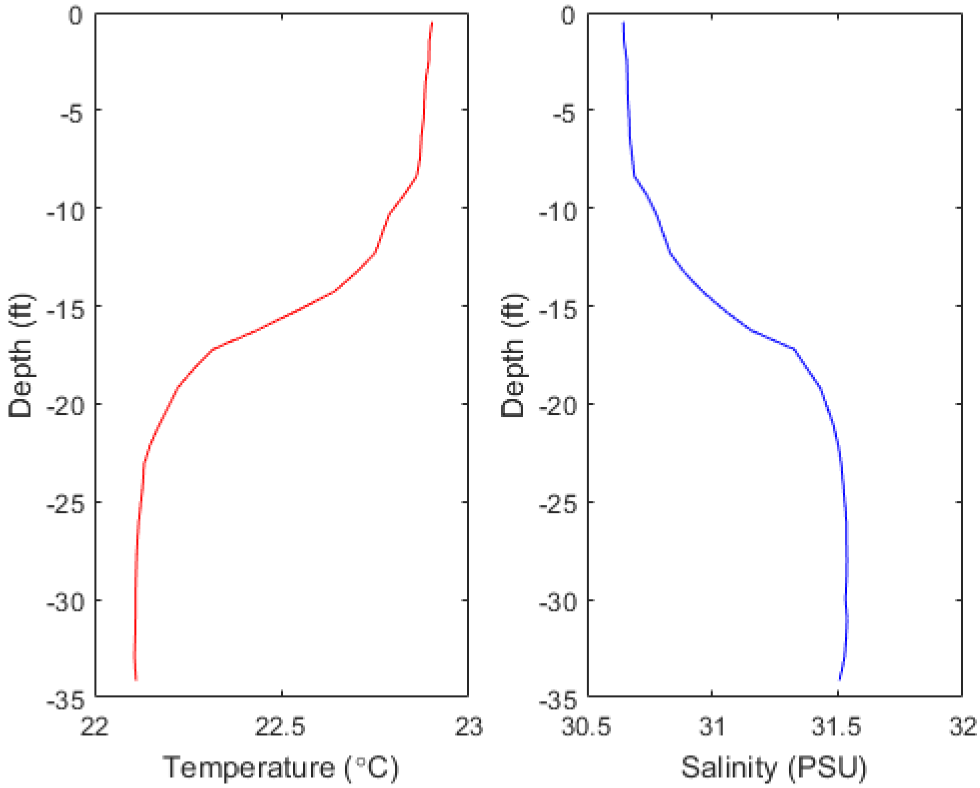


Рисунок 2.2 – Пример построения графиков зависимости температуры воды от глубины (красный) и зависимости солености воды от глубины (синий)

Дополнительный раздел. Построение вертикальных профилей солености и температуры воды на одном графике

В океанологии профили распределения солености и температуры обычно строятся на одном графике. Если значения температуры и солености достаточно близки, то их можно построить в одних осях, но чаще всего этот вариант не работает, и необходимо построить две кривые на одном графике так, чтобы на чтобы на нём присутствовали две независимые оси x (обычно одну располагают сверху, а вторую снизу). Это и попробуем сделать в этом дополнительном задании.

Задание:

1. Постройте отдельно график зависимости температуры от глубины.

2. Затем при помощи `get` запишем в переменную `gp` информацию о том, где расположены оси этого графика:
   `gp = get(gca,'pos');`
3. Теперь создаём новые оси в том же самом месте (`'pos',gp`), делаем их бесцветными (`'color','none'`) и располагаем подписи наверху (`'XAxisLocation','top'`):
   `axes('pos',gp,'color','none','XAxisLocation','top');.`
4. Вводим команду `hold on`, чтобы последующее построение строилось в указанных осях. Теперь постройте график солености.
   Вы должны получить график следующего вида:

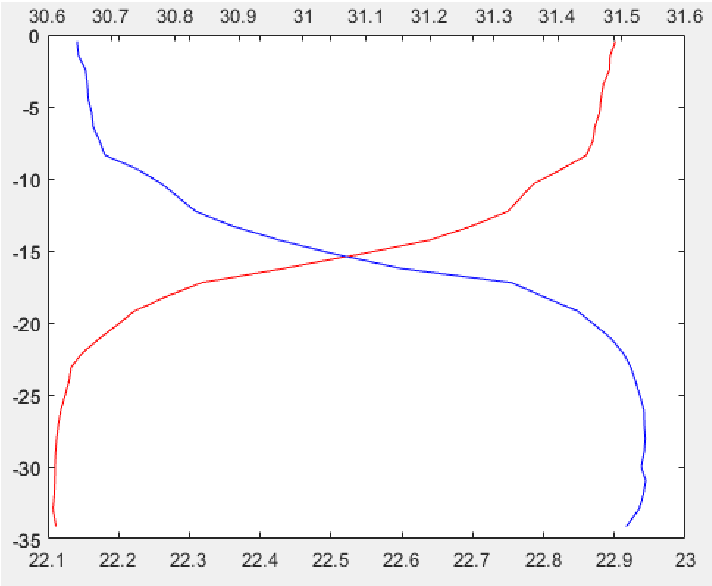


Рисунок 2.3 – Построение профиля распределения солености и температуры на одном графике

5. Подпишите оси графика, чтобы получить график вида:

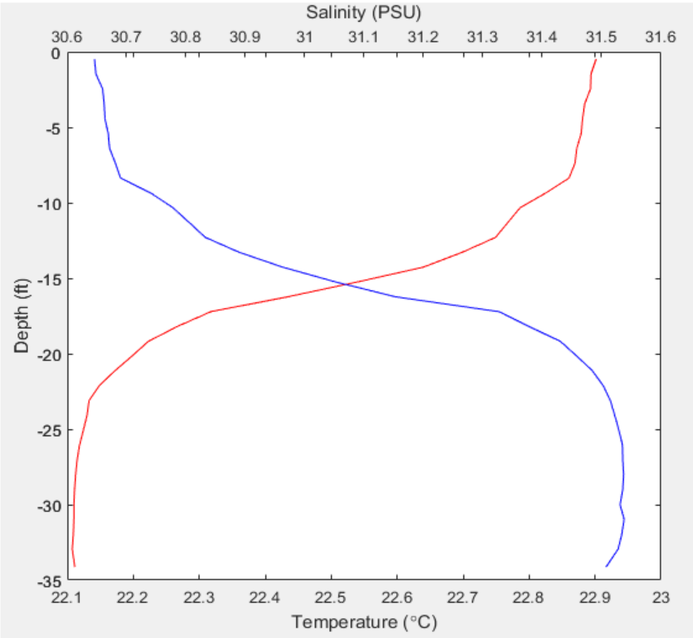


Рисунок 2.4 – Внесение подписей осей графика профиля распределения солености и температуры

6. Сейчас у графика в верхней шкале оси x дублируются отметки, их можно выключить, использовав команду `box off`:
   `set(gca, 'box','off')`
7. Теперь построим сетку на графике, используя команду `grid on`. Поскольку у нас разные величины, то и сетка для двух графиков будет разная, и вы получите график вида:

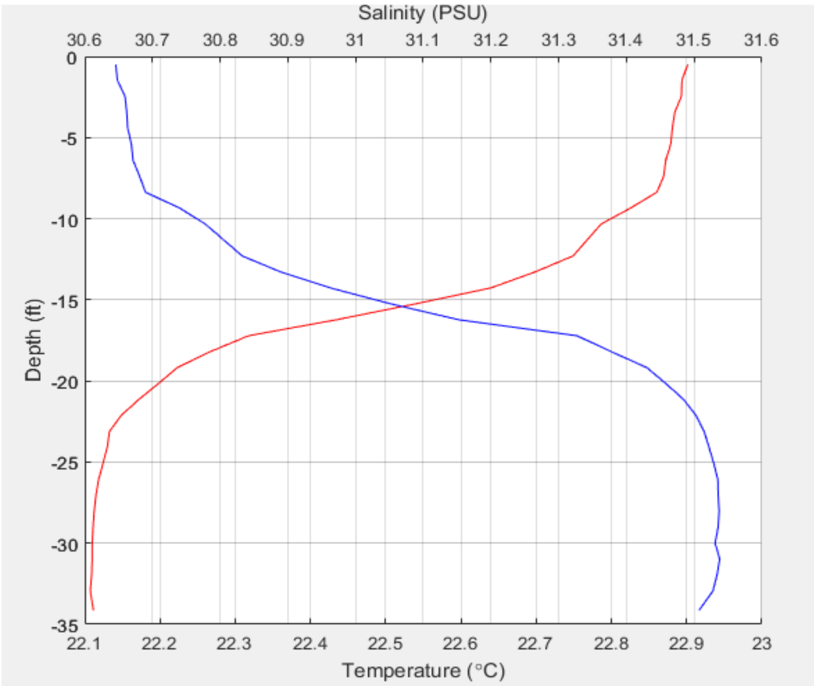


Рисунок 2.5 – Внесение сетки на графике

Для того чтобы отобразить единую сетку для обоих графиков, необходимо настроить ее так, чтобы шкалы были разделены на одинаковое количество частей, для этого допишем еще несколько установок в команду `set` и `axes`. Подберите такие значения A, B, C и D, чтобы получилась равномерная сетка:

`set(gca, 'box','off','XLim',[A B]);` - установка для оси температуры

`axes('pos',gp,'color','none','XAxisLocation','top','XLim',[C D]);` - установка для оси солености.

8. Последним этапом поменяйте вид графика на точечный. В итоге вы должны получить график следующего вида:

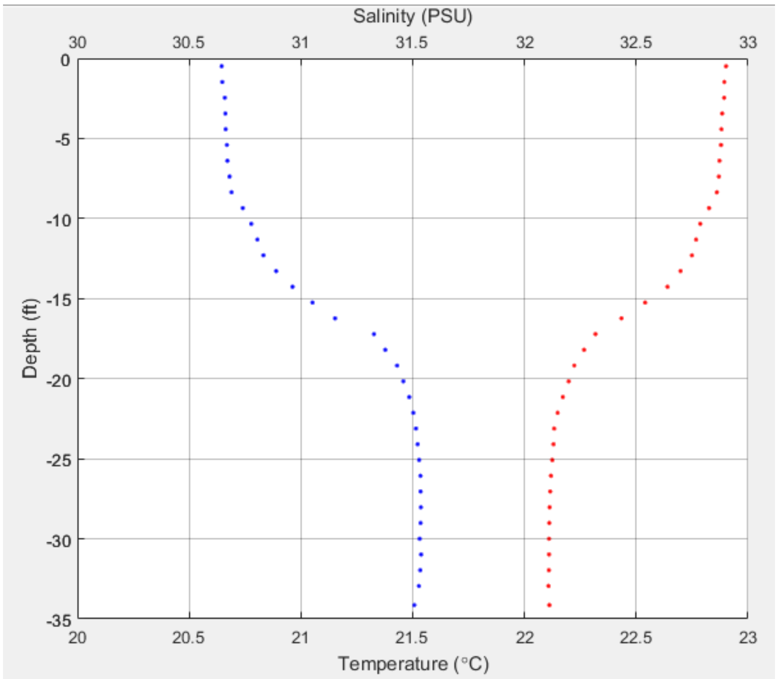


Рисунок 2.6 – Итоговый вид профиля распределения солености и температуры

Кейс 3. Исследование цвета океана

В этом практическом задании рассмотрим данные дистанционного зондирования, обработаем данные о цвете океана.

Задачи:

1) Загрузить данные о цвете океана с сайта.

2) Импортировать данные.

3) Отобразить данные определенной области океана.

4) Построить график среднегодовых значений данных в пределах выбранной зоны.

5) Создать цикл для обработки нескольких файлов.

Задание будет выполняться по вариантам, представленные в Таблице 3.1.

Таблица 3.1 – Варианты для Кейса 3

Номер варианта	Год исследования
1	2018
2	2017
3	2016
4	2015
5	2014

ЧАСТЬ 1. СОЗДАНИЕ КАРТЫ РАСПРЕДЕЛЕНИЯ ХЛОРОФИЛЛА В МИРОВОМ ОКЕАНЕ

Для начала давайте попробуем скачать данные с сайта NASA, где хранится информация о многолетних наблюдениях за океаном: https://oceandata.sci.gsfc.nasa.gov/. Данные в архивах NASA находятся в открытом доступе, но для того, чтобы скачать их, необходимо зарегистрироваться на сайте и создать свой аккаунт Earthdata. Поскольку данных действительно много, то все файлы содержат достаточно длинные и зачастую сложные к понимаю названия, однако на сайте есть расшифровки и руководство к использованию материалов по ссылке: https://oceancolor.gsfc.nasa.gov/docs/technical/#UG.

В данном задании не будем подробно останавливаться на характеристике данных, которые можно найти на сайте, скажем только, что среди них можно найти, например, данные по концентрации хлорофилла, отражательную способность, концентрацию кальцита, температуру поверхности океана, соленость воды и многое другое. В этом кейсе исследуем концентрацию хлорофилла.

По ссылке http://oceancolor.gsfc.nasa.gov/cgi/l3 можно найти необходимые нам данные. Здесь представлена информация о разных характеристиках океана с разных спутников. Так как эти спутники получали данные в разное время в различных регионах планеты, то в зависимости от цели и области исследования нужно выбрать спутники, отвечающие конкретному запросу.

Для нашего практического занятия выберем данные о концентрации хлорофилла Aqua Modis, OCI Algorithm. Далее можно выбрать, за какой период хотим получить данные – выберем за год – Annual composite и разрешение – 4 км. В итоге окно должно выглядеть так, как показано на Рисунке 3.1.

Рисунок 3.1 – Вид окна для выгрузки данных для Кейса 3

Выберем данные за необходимый вам год (согласно вашему варианту) в формате SMI (Standard Mapped Image). В итоге вы скачаете файл с названием A20180012018365.L3m_YR_CHL_chlor_a_4km и расширением nc. Данный формат файла соответствует NetCDF (Network Common Data Form) — формату файлов для хранения научных данных, широко употребляемому в гидрометеорологии. Для работы с файлами такого формата существует несколько команд.

Для начала определим, какие вообще данные и переменные есть в файле. Сделать это можно с помощью команды `ncdisp`:

```
ncdisp  A20180012018365.L3m_YR_CHL_chlor_a_4km.nc;
```

Эта команда откроет вам все данные про содержимое файла, но нас будет интересовать пока только размерность файла и описание переменных (Рисунок 3.2).

```
Dimensions:
           lat          = 4320
           lon          = 8640
           rgb          = 3
           eightbitcolor = 256
Variables:
    chlor_a
           Size:       8640x4320
           Dimensions: lon,lat
           Datatype:   single
           Attributes:
                       long_name     = 'Chlorophyll Concentration, OCI Algorithm'
```

Рисунок 3.2 – Представление данных о размерности файла и описании переменных (Кейс 3)

Присвоить данные из файла переменным позволяет команда `ncread`:

```
chlor_a=ncread('A20180012018365.L3m_YR_CHL_chlor_a_4km.nc','c
hlor_a');
```

```
lon=ncread('A20180012018365.L3m_YR_CHL_chlor_a_4km.nc','lon'
);
```

```
lat=ncread('A20180012018365.L3m_YR_CHL_chlor_a_4km.nc','lat'
);
```

Попробуем изобразить данные о концентрации хлорофилла графически, используя команду `imagesc`. Эта команда позволяет отобразить данные массива в диапазоне цветов.

Что вы видите на изображении?

Изображение получилось недостаточно четким, так как характеристики отображения используются по умолчанию. Используя команду `imagesc`, можно задать цветовую гамму отображения, например, с помощью еще одной команды – `clims`. Эта команда задает двухэлементный вектор `[cmin cmax]`, элементы массива меньше либо равные `cmin` обозначаются первым цветом диапазона, элементы массива больше либо равные `cmax` обозначаются последним цветом диапазона, все элементы между двумя этими значениями распределяются по всему диапазону цветов. Код при этом выглядит следующим образом:

```
clims=[ 0 0.3];
imagesc(chlor_a, clims);
```

Задание 1. Задайте значения `clims` **0 0.5; 0 2; 0 5; 0.3 0.5. Как при этом меняется изображение?**

Теперь на изображении показывается распределение и количество хлорофилла, собранное MODIS Aqua и усредненное за 2018 год. Добавим на карту реальные значения широты и долготы:

```
imagesc(lon, lat, chlor_a, clims);
```

Сейчас карта мира перевернута. Есть несколько способов перевернуть ее в привычный всем формат, в этом упражнении используем функцию `rot90` и перевернем массив на 90 градусов:

```
imagesc(lon, lat,rot90(chlor_a), clims);
```

Получилось ли у нас действительное отображения карты? На самом деле – нет. Нами были построены значения широты и долготы, но значения широты являются обратными (отрицательная широта в северном полушарии, в отличие от южного полушария). Чтобы решить эту проблему, можно встроить функцию `flipud`, чтобы перевернуть матрицу `chlor_a` сверху вниз, а также установить необходимую ориентацию оси `y`, используя функцию `set`:

```
imagesc(lon, lat, flipud(rot90(chlor_a)), clims);
set(gca,'YDir', 'normal');
```

Функция `set` является достаточно универсальной функцией настройки изображений в MatLab и устанавливает какую-либо характеристику (в нашем случае `'normal'`) какому-либо элементу изображения (в нашем случае `'YDir'` – направление оси `y`) какому-либо объекту (в нашем случае `gca` – последние используемые оси во всей программе).

Чтобы наша карта выглядела более реалистично и было понятно, что она означает – добавим расшифровку:

```
h=colorbar
```

Задание 2. Добавьте название нашей карте – год исследования и подпись к расшифровке карты – мг*м(-3).

ЧАСТЬ 2. СОЗДАНИЕ КАРТЫ РАСПРЕДЕЛЕНИЯ ХЛОРОФИЛЛА В ОПРЕДЕЛЕННЫХ ГРАНИЦАХ

В этом разделе построим карту распределения хлорофилла не во всем мире, а только в определенном участке – Тасмановом море. Нам будет необходимо ограничить зону построения координатами – с 145° до 175° восточной долготы и с 31° до 51° южной широты:

```
subAX=[145 175 -51 -30];
```

Задание 1. Создайте два вектора `xidx` **и** `yidx`**, содержащих координаты искомого участка. Вы можете выполнить это задание несколькими способами, но рекомендуем использовать для этого функцию** `find`**.**

Задание 2. Создайте массив `sub_chl`**, содержащий значения концентрации хлорофилла в выбранной области.**

Задание 3. Постройте карту выбранного участка.

ЧАСТЬ 3. СОЗДАНИЕ ЦИКЛА

Представьте, что для вашего исследования необходимо построить карты распределения концентрации хлорофилла за несколько лет. Вам не нужно повторять одну и ту же процедуру несколько раз, так как MatLab позволяет вам создать скрипт и вызывать его одной командой. Давайте разберемся, каким образом это можно сделать:

1. Нажмите кнопку “New Script” в меню File, перед вами откроется новое окно.
2. Начиная прописывать новую программу, всегда лучше стереть данные всех ваших предыдущих шагов, чтобы избежать путаницы. Для этого можно использовать команду `clear`.

3. В вашей рабочей папке находится некое количество файлов, которые вам нужно обработать. Вы можете, конечно, вручную запросить данные и вписать их названия в программу, но также вы можете облегчить себе жизнь и использовать для этого код. Можно использовать команду `dir`, которая перечислит файлы или папки, удовлетворяющие описанному условию. Создадим текстовый вектор `flist`, который будет содержать названия всех файлов в папке:
   `flist=dir('местоположение файлов\*.nc')`
   формулировка «`\*.nc`» позволит найти все файлы разрешения .nc.
   Теперь можно открыть вектор `flist` и посмотреть все названия файлов и их количество.
4. Создадим переменную, которая будет содержать количество файлов, а также прописанный путь, где наша программа будет искать необходимые файлы:
   `nfiles=length(flist);`

   `dirpath=['местоположение файлов']`
5. Теперь создадим цикл, который будет обрабатывать изображения вне зависимости от их количества. В MatLab циклы строятся при помощи команды `for` и заканчиваются командой `end`.
   Построим цикл для всех наших изображений:

```
for N=1:nfiles – задает количество файлов в папке
    filename=[dirpath flist(N).name]; - записывает имя файла
    chlor_a=ncread(filename, 'chlor_a');
    lon=ncread(filename, 'lon');

    lat=ncread(filename, 'lat');
    subAX=[145 175 -51 -31];
    clims=[0 0.3];
    xidx=find(lon>=subAX(1) & lon<=subAX(2));
    yidx=find(lat>=subAX(3) & lat<=subAX(4));
    sub_chl=chlor_a(xidx,yidx);
    year=flist(N).name(2:5);
    sub_lonx=lon(xidx);
    sub_laty=lat(yidx);
end
```

Что делают последние три команды цикла?

Задание. Постройте карты концентрации хлорофилла, полученные из всех файлов, в одном окне. Используйте для этого команду `subplot`.

Кейс 4. Исследование температуры океана

В этом кейсе рассмотрим данные дистанционного зондирования, обработаем данные о цвете океана.

Задачи:

1) Загрузить данные о температуре океана с сайта.

2) Импортировать данные.

3) Отобразить данные определенной области океана.

4) Построить картографические проекции (разных видов) температуры океана всего мира и выделенного участка.

Задание будет выполняться по вариантам, указанных в Таблице 4.1.

Таблица 4.1 – Варианты для Кейса 4

Номер варианта	Год исследования
1	2018
2	2017
3	2016
4	2015
5	2014

ЧАСТЬ 1. СОЗДАНИЕ КАРТЫ РАСПРЕДЕЛЕНИЯ ТЕМПЕРАТУРЫ В МИРОВОМ ОКЕАНЕ

Для этого кейса выберем данные о температуре океана Aqua Modis Sea Surface Temperature (daytime). Далее можно выбрать, за какой период необходимо получить данные – выберем за год – Annual composite и разрешение – 4 км. В итоге окно должно выглядеть так, как показано на Рисунке 4.1.

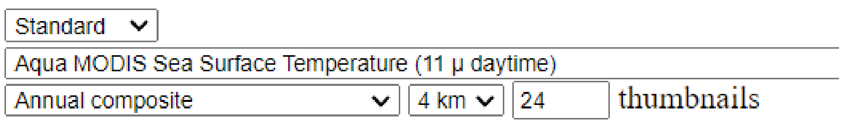


Рисунок 4.1 – Вид окна для выгрузки данных для Кейса 4

Выберем данные за необходимый вам год (согласно вашему варианту) в формате SMI (Standard Mapped Image). В итоге вы скачаете файл с названием: A20180012018365.L3m_YR_O_SST_sst_4km и расширением nc. Данный формат файла соответствует NetCDF (Network Common Data Form) — формату файлов для хранения научных данных, широко употребляемому в гидрометеорологии. Для работы с файлами такого формата существует несколько команд.

Для начала определим, какие вообще данные и переменные есть в файле. Сделать это можно с помощью команды `ncdisp`:

```
ncdisp AQUA_MODIS.20180101_20181231.L3m.YR.SST.sst.4km.nc;
```

Эта команда откроет вам все данные про содержимое файла, но нас будет интересовать пока только размерность файла и описание переменных:

```
Variables:
    sst
           Size:       8640x4320
           Dimensions: lon,lat
           Datatype:   int16
           Attributes:
                       long_name      = 'Sea Surface Temperature'
```

Рисунок 4.2 – Представление данных о размерности файла и описании переменных (Кейс 4)

Присвоить данные из файла переменным позволяет команда `ncread`:

```
sst =
ncread('AQUA_MODIS.20180101_20181231.L3m.YR.SST.sst.4km.nc',
'sst');
lon=ncread('AQUA_MODIS.20180101_20181231.L3m.YR.SST.sst.4km.
nc','lon');
lat=ncread('AQUA_MODIS.20180101_20181231.L3m.YR.SST.sst.4km.
nc','lat');
```

Полученную переменную `sst` можно визуализировать, например, через функцию `imagesc`:

```
imagesc(sst)
```

Сейчас карта мира перевернута. Есть несколько способов перевернуть ее в привычный всем формат, в этом упражнении перевернем массив с помощью команды `permute`:

```
sst = permute(sst, [2 1]);
```

Эта команда позволяет вручную поменять состав матрицы. В таком виде эта команда позволяет повернуть матрицу относительно главной диагонали.

Какие размеры у матриц сейчас?

В этом задании необходимо строить карты, которые требуют, чтобы матрицы были одинакового размера. Для каждой ячейки матрицы `sst` должны существовать такие же ячейки в матрицах `lat` и `lon`, содержащие координаты. Для этого можно использовать команду `ndgrid`:

```
[lat,lon]=ndgrid(lat,lon);
```

Теперь нам нужен «макет» карты, которую нужно построить. В MatLab сделать это можно разными способами, воспользуемся командой `worldmap`:

```
worldmap('World')
```

В качестве картографической проекции для карты мира по умолчанию выбирается проекция Робинсона (`'robinson'`). Нанесем на эту подложку наши данные:

`geoshow(lat,lon,sst,'displaytype','texture')`- команда, позволяющая отразить широту и долготу в выбранной проекции. Аргумент `'texture'` наиболее похож на команду `imagesc`. По итогу получим изображение вида:

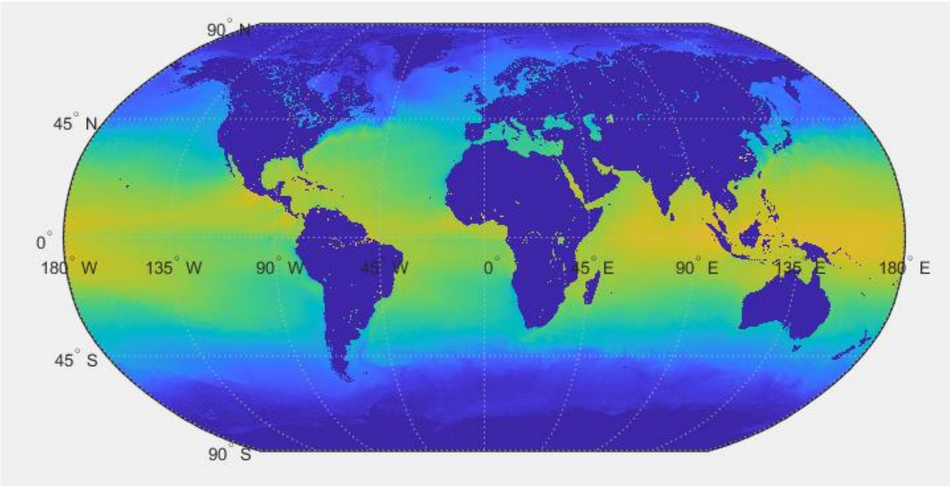


Рисунок 4.3 – Карта, построенная с помощью команды `worldmap`

Чтобы наша карта выглядела более реалистично и было понятно, что она означает – добавим расшифровку:

```
h=colorbar
```

Задание. Добавьте название нашей карте – год исследования и подпись к расшифровке карты – мг*м(-3)

ЧАСТЬ 2. ОГРАНИЧЕНИЕ РАЙОНА ОТОБРАЖЕНИЯ

В этом упражнении ограничить область отображения по широте и долготе можно сразу в команде `worldmap`:

```
worldmap([40 80],[-40 20])
```

Обратите внимание, что картографическая проекция автоматически поменялась на равнопромежуточную коническую (`'eqdconic'`).

Теперь установим границы цветовой шкалы, чтобы отображение температуры было более наглядным, и изменим цвет континентов на серый, указав три значения RGB:

```
caxis([-2 25])
geoshow('landareas.shp','FaceColor',[0.5 0.5 0.5])
```

Полученную карту можно раскритиковать за то, что невозможно отличить значения с низкой температурой от `NaN` (пропусков в данных). Самый простой способ избежать этого - выбрать другой тип отображения данных на карте (заменим `'texturemap'` на `'surface'`).

Как именно изменилась карта? В чем отличие двух видов отображения?

Всего в `geoshow` существует пять типов отображения: `'surface'`, `'contour'`, `'mesh'`, `'texturemap'`, `'image'`. Рассмотрим, как отображают наши данные команды `'contour'` и `'mesh'`.

Параметр `'mesh'` представит наши данные на карте в виде сетки. Если вы поменяете вид отображения для всех наших данных, то сетки особо не будет видно, так как данных слишком много. Поэтому для этого построения будем брать только каждое десятое значение.

Задание 2. Пропишите команду `geoshow` **с массивами** `lat,lon,sst` **таким образом, чтобы учитывалось только каждое десятое значение.**

Параметр `'contour'` представит наши данные на карте в виде изолиний. Для построения с этим параметром сразу обозначим шаг прорисовки изолиний (`'LevelStep',1`), иначе по умолчанию MatLab построит слишком мало изолиний. Также сразу сделаем изолинии черными и заполним пространство между ними:

```
geoshow
(…'contour','Fill','on','LineColor','black','LevelStep',1)
```

Последним этапом поменяем формат проекции на проекцию Меркатора – один из самых популярных форматов в океанологии. Тонкая настройка карты выполняется при помощи функции `setm`. Необходимо помнить, что задавать проекцию следует в самом начале, сразу после `worldmap`:

```
setm(gca,'MapProjection','mercator')
```

Задание 3. Импортируйте данные о концентрации хлорофилла из второго файла. Постройте изолинии концентраций хлорофилла в Балтийском море. Ваше окно выбора информации должно выглядеть следующим образом (рис. 4.4):

Рисунок 4.4 - Вид окна для выгрузки данных для Кейса 4 для построения изолиний хлорофилла

Кейс 5. Изменение климата и анализ экстремальных значений

В этом упражнении проанализируем месячные максимумы температуры и осадков, чтобы вычислить период их повторяемости в течение 100 лет. Начнем с быстрого анализа, чтобы получить исходный результат, а затем проведем более глубокий анализ для соотнесения модельных значений с эмпирическими данными. В этом упражнении поработаем сначала с климатическими данными США (NOAA), а дополнительное задание будет выполняться на основе климатических данных городов России.

Задачи:

1) Анализ климатических данных.

2) Выявление экстремальных событий с использованием обобщённого распределения экстремального значения.

Задание будет выполняться по вариантам, указанных в Таблице 5.1.

Таблица 5.1 – Варианты для Кейса 5

Номер варианта	Штат исследования
1	Калифорния
2	Флорида
3	Небраска
4	Юта
5	Вашингтон

Часть 1. НАХОЖДЕНИЕ МАКСИМАЛЬНЫХ ЗНАЧЕНИЙ

Сначала добудем наши климатические данные с сайта NOAA [2]. Ссылка: https://www.ncei.noaa.gov/access/monitoring/climate-at-a-glance/statewide/time-series

Выберем данные о максимальной температуре (temperature) воздуха за все месяцы с 1922 года по 2022 год в штате согласно варианту. Меню выбора должно выглядеть, как на рисунке 5.1.

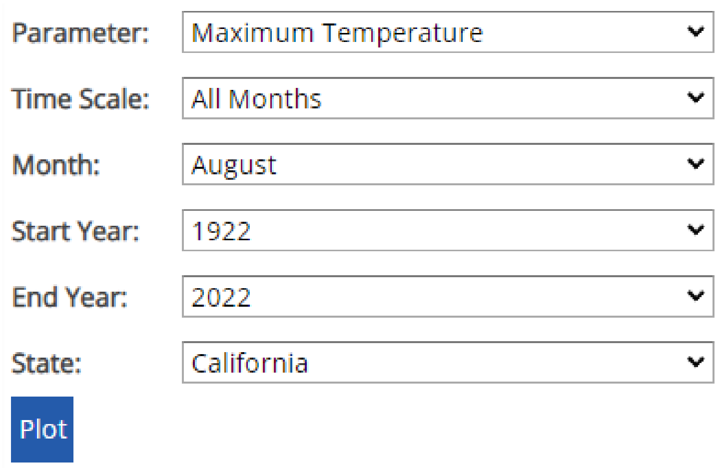


Рисунок 5.1 – Меню выбора

Скачайте данные в формате CSV. Таким же образом скачайте данные об осадках (precipitation).

Задание 1. Импортируйте данные о дате, температуре и осадках в MatLab до 12.2021 года.

Задание 2. Постройте графики температуры и осадков. Начните ваш код со следующей команды:

```
monthV = 1922+1/12:1/12:2022;
```

Что она делает?

Графики должны выглядеть примерно так, как указано на рисунке 5.2.

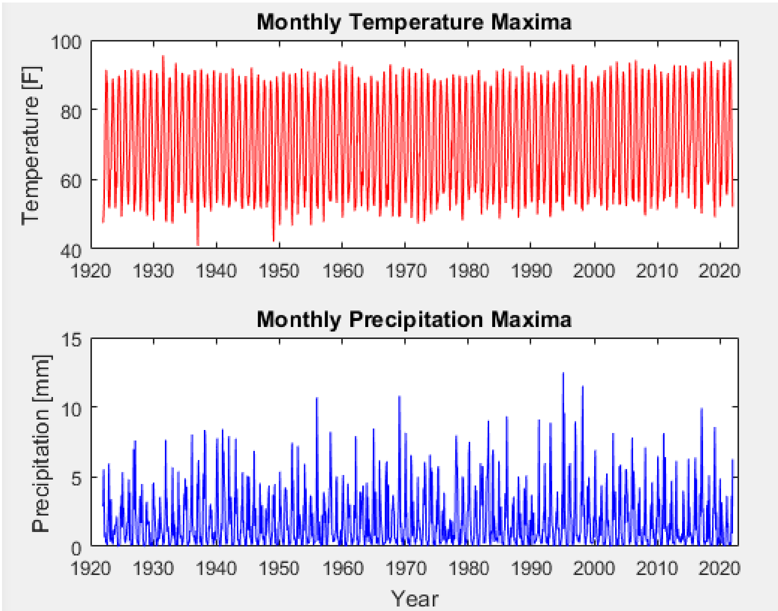


Рисунок 5.2 – Графики температуры и осадков

Если будет необходимо ограничить значение годов (ось x), используйте команду `xlim`.

Задание 3. Найдите максимальные значения температуры и осадков за каждый год. Назовите переменные `tempmax` и `prcpmax`. Для выполнения этого задания вы можете использовать цикл `for` или функцию `reshape`.

Задание 4. Постройте графики максимальных значений температуры и количества осадков. Графики должны выглядеть примерно так, как указано на рисунке 5.3.

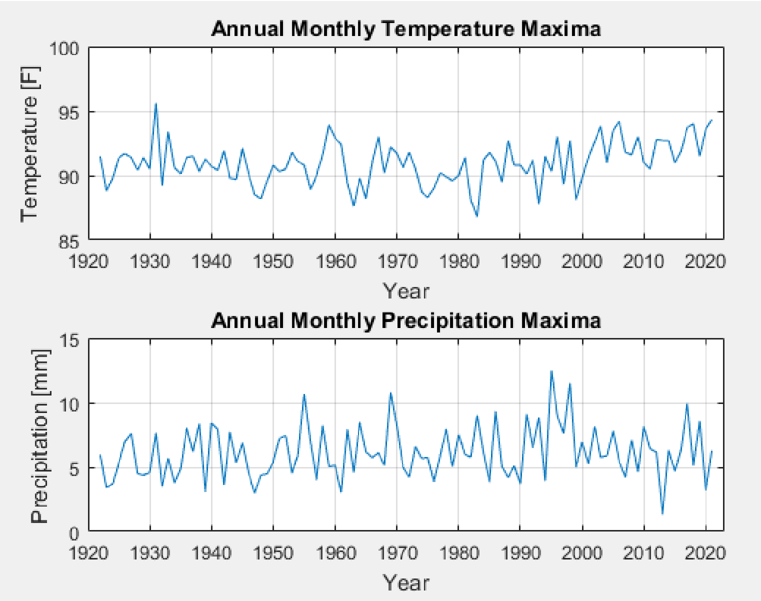


Рисунок 5.3 – Графики максимальных значений температуры и количества осадков

ЧАСТЬ 2. АНАЛИЗ ЭКСТРЕМАЛЬНЫХ ЗНАЧЕНИЙ

Теория экстремальных значений позволяет определить крайние отклонения выборки от средних значений. Этот анализ широко используется в науках о Земле, строительстве и финансовом деле. С ее помощью можно, например, оценить вероятность наводнений, предельных температур и финансового кризиса.

Обобщенное распределение экстремальных значений (`GEV`) представляет собой обобщение трех разных распределений (Гумбельта, Фрешета и Вейбулла) и используется для приближенного моделирования максимумов последовательностей [3]. Эта распределение определяется тремя параметрами: μ— параметр положения, σ — параметр масштаба, и ξ— параметр формы, и описывается формулой:

$$f(\xi,\mu,\sigma) = e^{-\left(1+\xi\left(\frac{\chi-\mu}{\sigma}\right)\right)^{-\frac{1}{\xi}}}\frac{1}{\sigma}\left(1+\xi\left(\frac{\chi-\mu}{\sigma}\right)\right)^{-\left(\frac{1}{\xi}\right)-1}$$

В MatLab это распределение уже «вшито», найти три параметра можно с помощью функции `gevfit`, которая вычисляет параметры с максимальной вероятностью. Найдем три параметра распределений температуры и осадков:

```
parmhatP    = gevfit(prcpmax);
parmhatT    = gevfit(tempmax);
```

`parmhat (1)` — параметр формы, `parmhat (2)` — параметр масштаба, а `parmhat (3)` — параметр местоположения.

Теперь можно проанализировать уровень и период возврата экстремального события. Период возврата – среднее время между экстремальными событиями [4]. Например, если у нас есть данные, что в определенном месте землетрясение силой, скажем, 7.0 баллов по шкале Рихтера происходит в среднем каждые 100 лет, то будет считаться, что это 100-летнее событие, обычно обозначаемое как «событие раз в столетие». В любой год вероятность землетрясения такой силы равна 1/100 = 1%.

Конечно, это не означает, что землетрясение такой силы будет происходить ровно раз в 100 лет. 1% — это ожидаемое событие. Землетрясение такой силы может не произойти ни разу в следующие 100 лет, а может произойти дважды в следующие 100 лет. Вероятность того, что оно произойдет хотя бы раз в 100 лет, равна $1 - (1/100)^{100} = 63.4\%$.

Чтобы измерить устойчивость к экстремальному событию за период r лет, проверим, является ли $F(x) > 1 - 1/r$, где x — предполагаемое максимальное значение некоторой переменной (например, температура или уровень землетрясений по шкале Рихтера) в данный период времени (скажем, в год), а F(x) — кумулятивная функция распределения в точке x. Уровень возврата соответствует значению экстремального значения.

MatLab позволяет автоматически вычислить период и уровень возврата с помощью функции `gevinv`:

```
returnP = 1:100; - установим период возврата
returnL = 1-1./returnP;
returnLevel_T = gevinv(returnL,parmhatT(1),parmhatT(2),parmhatT(3)); - вычислим уровень возврата
returnLevel_P = gevinv(returnL,parmhatP(1),parmhatP(2),parmhatP(3));
```

Задание 1. Постройте графики зависимости уровня возврата от периода. Графики должны выглядеть примерно так, как указано на рисунке 5.4.

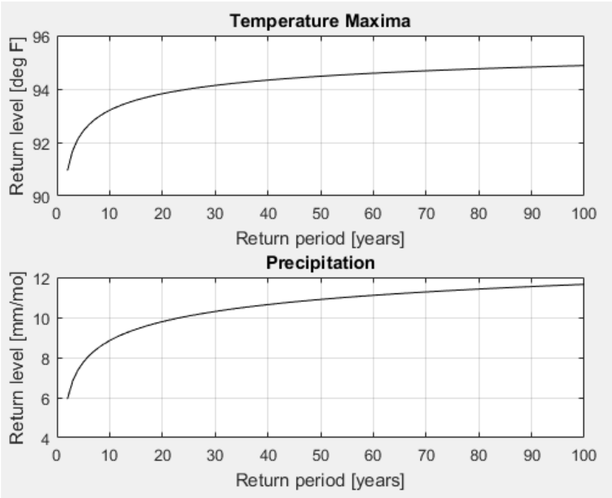


Рисунок 5.4 – Графики зависимости уровня возврата от периода

Какие экстремальные значение осадков и температуры будут наблюдаться примерно раз в 50 лет? Насколько точна эта модель?

ЧАСТЬ 3. АНАЛИЗ КЛИМАТИЧЕСКИХ ДАННЫХ В РОССИИ

Климатические данные регионов и городов России можно получить на сервере ВНИИГМИ-МЦД по ссылке: http://aisori-m.meteo.ru/waisori/index.xhtml?idata=8. После авторизации на сайте массивы данных должны выглядеть так, как представлено на рисунке 5.5.

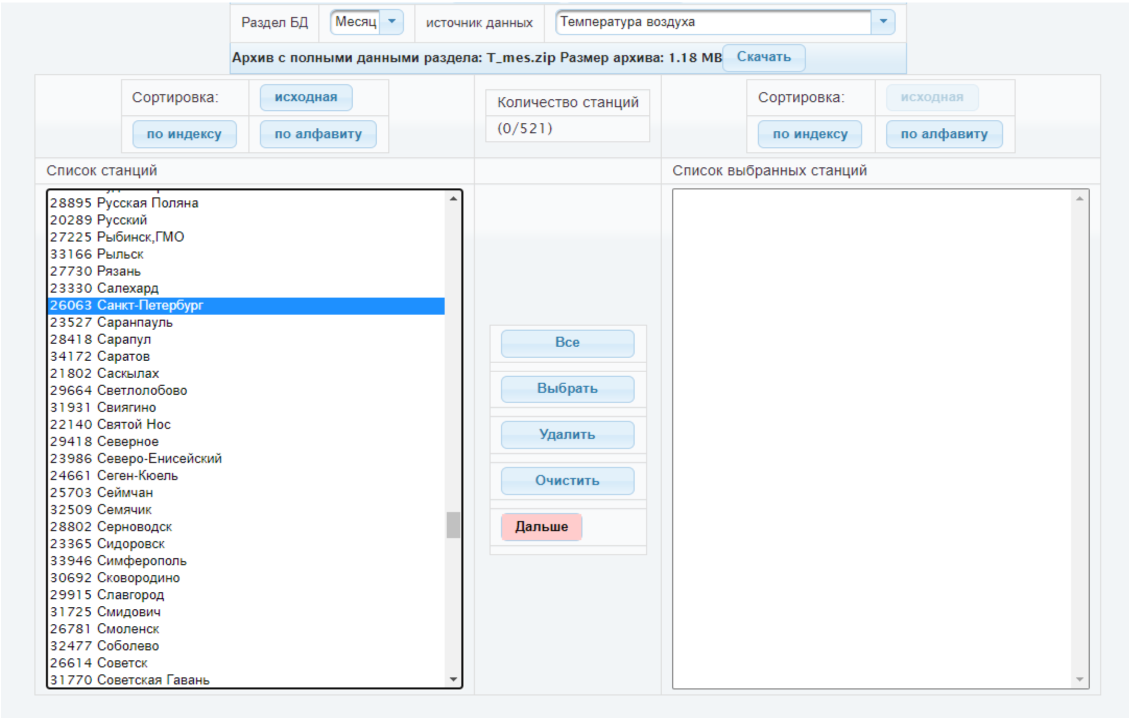


Рисунок 5.5 – Массивы данных на сервере ВНИИГМИ-МЦД

Задание. Загрузите данные о температуре воздуха в Санкт-Петербурге. Постройте графики максимальных значений температуры за каждый год с 1950 до 2020. Постройте графики зависимости уровня возврата от периода.

Кейс 6. Анализ звуков морских млекопитающих

В этом практическом задании поработаем в MatLab с аудиофайлами, а именно с записью голосов морских млекопитающих. Эти виды во многом полагаются на звуки, поэтому наличие антропогенных шумов в морской среде может значительно влиять на их жизнь. В этом задании проанализируем звуки, записанные гидрофоном MARS в Калифорнии [6]. Диапазон чувствительности гидрофона 10–100 000 Гц.

Задачи:

1) Визуализация аудио с помощью спектрограмм.

2) Анализ данных спектрограмм.

MatLab может обрабатывать аудиосигналы и преобразовывать их в массивы данных. Начнем нашу работу с записи голосов косаток (файл 458854__mbari-mars__orcas-killer-whales). Прочитать данные из аудио позволяет команда `audioread`:

```
[y, Fs]=audioread('458854__mbari-mars__orcas-killer-whales.wav');
```

где y содержит данные о значениях звуковых сигналов, а `Fs` – частота.

Какая частота у нашего файла?

Задание 1. Задайте переменную time как вектор времени, в котором каждый элемент соответствует времени каждого звукового сигнала. Постройте график зависимости звуковых сигналов от времени (в секундах). График должен выглядеть примерно, как на рисунке 6.1 (ваши данные будут отличаться от этого примера).

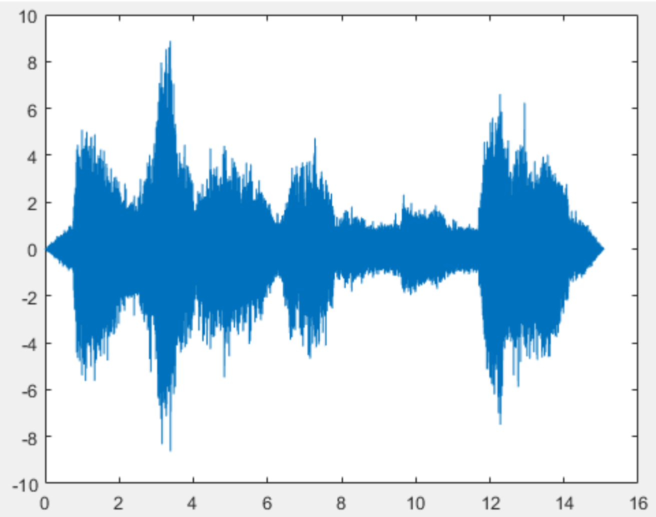


Рисунок 6.1 – График зависимости звуковых сигналов от времени (в секундах)

В MatLab встроен аудиоплеер, так что вы можете запустить нашу аудиозапись прямо в программе, используя команду `audioplayer`:

```
player = audioplayer(y,Fs); - загрузка аудио в плеер
play(player); - запуск аудио
```

Вы можете использовать команды `pause`, `resume` и `stop` чтобы управлять аудио.

Задание 2. Вы можете замедлить воспроизведение аудио, изменяя его частоту. Запустите аудио с частотой 1600.

Теперь попробуем построить спектрограмму нашего аудио, то есть визуальное отображение спектра частот сигнала. Такие спектрограммы еще называются сонографами. Спектрограмма создается путем применения быстрого преобразования Фурье (БПФ) к звуку, записанному в электронном формате. БПФ разделяет частоты и амплитуды составляющих волн, и их можно изобразить визуально. Амплитуды, показанные разными цветами, представляют собой различные частоты (вертикальная ось) в разное время (горизонтальная ось). Как правило, для чтения спектрограмм используются «широкополосные» спектрограммы, потому что они дают нам больше информации о том, что происходит со звуком.

Построить спектрограмму можно с помощью команды `spectrogram`. Есть разные варианты использования этой команды, начнем с самого простого формата отображения:

```
s=spectrogram(y);
spectrogram(y,'yaxis')
```

Получим изображение, как на рисунке 6.2.

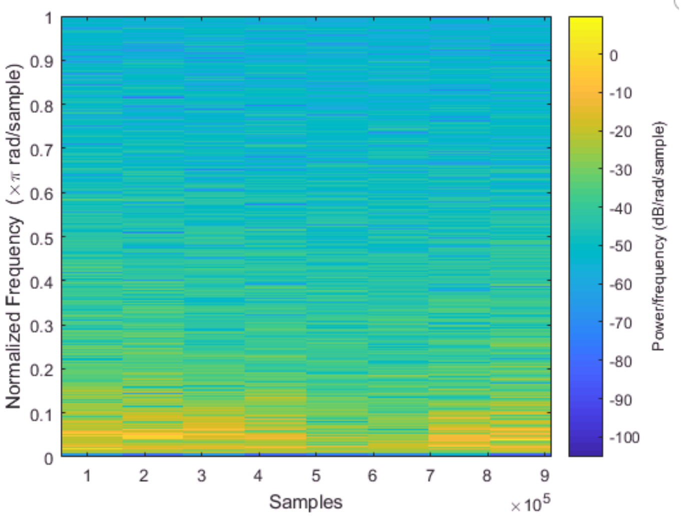


Рисунок 6.2 – График спектрограммы, построенный с помощью команды `spectrogram`

Это 2D отображение частот нашего сигнала. Для наглядности и лучшего понимания спектрограммы лучше представить ее в 3D-формате:

```
window=hamming(1000)
noverlap=256;
nfft=1024;
[S,F,T,P]=spectrogram(y,window,noverlap,nfft,Fs,'yaxis')
;
```

Здесь `window` делит наш сигнал на определенное количество сегментов, `noverlap` – количество перекрывающихся выборок, `nfft` - количество точек преобразования Фурье, `F` – частота наших данных, `T` – время.

Отобразим нашу спектрограмму:

```
surf(T,F,10*log10(P),'edgecolor', 'none');
```

Получим следующее изображение, представленное на рисунке 6.3.

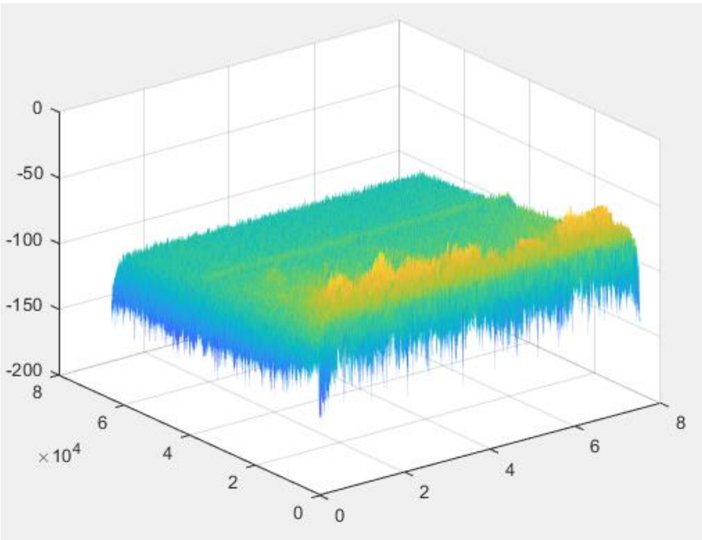


Рисунок 6.3 – График спектрограммы в 3D

Давайте рассмотрим нашу спектрограмму как отображение интенсивности звука как функции частоты (ось y) и времени (ось x):

```
axis tight;
view(0,90);
colormap(hot);
set(gca, 'clim', [-80 -30], 'ylim', [0, 20000]);
```

Задание 3. Добавьте спектрограмме подписи осей и легенду обозначения цветов.
Задание 4. Постройте спектрограмму записи мотора лодки (файл 79__23_07_13_H3_zodiac.wav**).**

Кейс 7. Анализ видового разнообразия рыб в Австралии

В этом упражнении проанализируем данные о видовом разнообразии рыб у берегов Австралии, полученные в рамках программы Reef Life Survey (RLS). RLS - это некоммерческая гражданская научная программа, в рамках которой обученные аквалангисты-добровольцы проводят стандартизированные исследования биоразнообразия скалистых и коралловых рифов по всему миру. Проанализируем данные, полученные с 2012 по 2017 годы.

Задачи:

1) Поиск уникальных значений в массиве.
2) Создание циклов.
3) Расчет индекса Симпсона.

Задание 1. Выгрузите данные из файла 'Reef_Life_Survey_(RLS)#_Global_reef_fish_dataset.csv' **в переменную** `RLSData`. Если теперь вы откроете массив `RLSData`, то увидите, что внутри у нас есть описание регионов '`Ecoregion`' Австралии, таксоны '`Taxon`' и общее количество особей определенного таксона '`Total`'.

Задание 2. Выгрузите количество исследованных регионов в переменную `ecoreg` **с использованием команды** `unique`**.**

Сколько получилось всего регионов? Сколько уникальных таксонов присутствует в этом массиве?

Видовое богатство – это количество различных видов, присутствующих в экосистеме, ландшафте или регионе. Подсчитаем видовое богатство и отобразим его на графике. Для этого нам нужно будет подсчитать количество уникальных видов для каждого региона. Воспользуемся командой `strcmp`, которая сравнивает заданную величину с массивом данных и создает логический массив из нулей (если значения не совпадают) и единиц (если совпадают):

```
idx=strcmp(ecoreg(1),RLSData.Ecoregion);
```

Задание 3. Создайте массив `RLSsub`**, состоящий из всех данных для первого региона. Используйте для этого логический массив** `idx`.

Задание 4. Создайте массив `splist`**, состоящий из уникальных видов для первого региона.**

Зададим переменную, которая будет содержать количество уникальных видов:

```
s=size(splist);
sr=s(1)
```

Задание 5. Создайте цикл, который выполнит предыдущие команды для всех регионов и сохранит количество видов в массив `richness=[];`

Задание 6. Постройте гистограмму (командой `bar`**) видового богатства следующего вида, как на рисунке 7.1 (ваши данные будут отличаться от этого примера).**

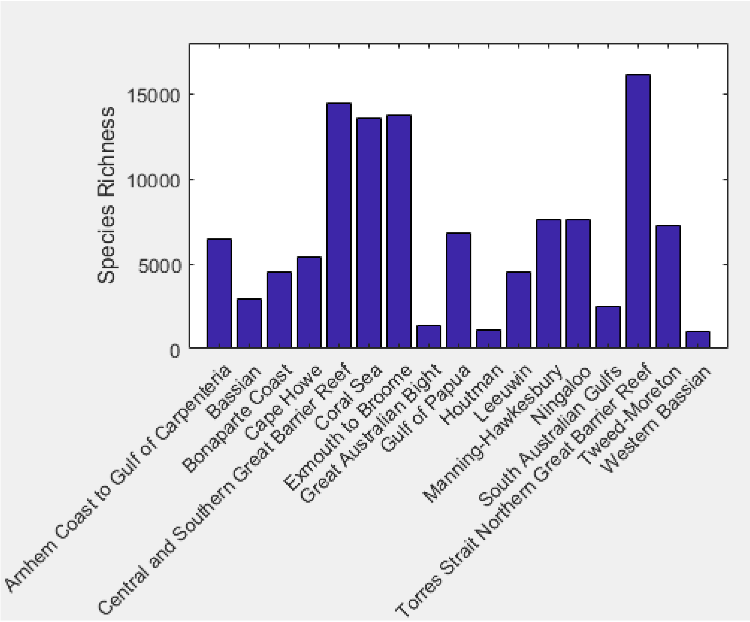


Рисунок 7.1 – Гистограмма видового богатства

Теперь вычислим индекс разнообразия Симпсона – индекс, описывающий вероятность принадлежности любых двух особей, случайно отобранных из неопределенно большого сообщества, к разным видам по формуле:

$$D = \frac{\sum n(n-1)}{N(N-1)},$$

где n – число особей определенного вида, а N – общее число особей. Если индекс равен нулю, то наблюдается бесконечное разнообразие, если единице – то разнообразие отсутствует, особи принадлежат одному виду. По мере увеличения D разнообразие уменьшается. Многие авторы считают, что наилучшая мера – это «индекс полидоминантности»: S= 1/D, его как раз и посчитаем. Начнем с расчета индекса Симпсона для первого региона.

Задание 7:

- **Снова задайте массив** `RLSsub`**, состоящий из всех данных для первого региона. Используйте для этого логический массив** `idx`.

- **Посчитайте общее количество всех организмов всех видов (N). Необходимые данные находятся в столбце** ‘`Total`’, **используйте команду** `sum`**.**
- **Создайте цикл чтобы просуммировать общее количество организмов всех видов (n). Необходимые данные находятся в столбце** ‘`Taxon`’.
- **Посчитайте сам индекс S= 1/D**

Задание 8. Создайте цикл, который посчитает индекс разнообразия Симпсона в вектор `Sall`.

Задание 9. Постройте гистограмму со значениями полученного индекса.

Кейс 8. Калибровка гидрологической модели

Задачи:
1) Создание массивов данных с различными распределениями.
2) Создание условий и циклов.
3) Создание отдельных функций и скриптов.
4) Использование функций оптимизации параметров.

ЧАСТЬ 1. ПОСТРОЕНИЕ ГИДРОЛОГИЧЕСКОЙ МОДЕЛИ

В этом практическом занятии рассмотрим модель дождевых стоков [7]. Представим, что бассейн реки — это обычное ведро с водой. Вода попадает в ведро из крана (это осадки) и вытекает из ведра из щели внизу и перелива сверху (рисунок 8.1).

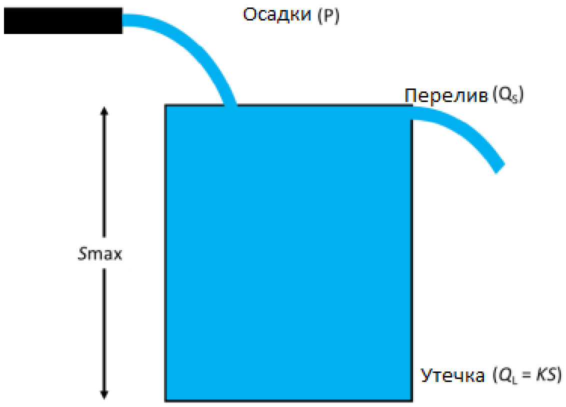


Рисунок 8.1 – Визуальное представление упрощенной модели дождевых стоков

Остальные движения воды считаем незначительными. В этой модели утечка из щели является меженным стоком, то есть наименьшим стоком, обычно наблюдающимся в межень, а перелив – полный сток за период выпадения осадков. Параметрами этой модели являются `Smax` и `K` – максимальный объем ведра и скорость вытекания воды из щели.

Задание 1. С помощью команды `gamrnd` **создайте случайное распределение осадков в течение 100 дней. Данная команда создаст гамма-распределение осадков. Укажите 10 и 2 в качестве параметров формы и масштаба. Назовите переменную** `precip`.

Будем считать, что эти данные являются реальными данными об осадках в нашем регионе. Уровень осадков может быть измерен различными способами, включая дождемер, радар и дистанционное зондирование. Будем считать, что были получены данные от дождемера. Так как любые эмпирические данные обычно содержат тот или иной уровень погрешности, то добавим немного шума нашим данным. Предположим, что погрешность имеет распределение Гаусса со средним, равным 0 мм, и стандартным отклонением, равным 1 мм.

Задание 2. С помощью команды `normrnd` **создайте случайное распределение погрешности осадков в течение 100 дней (**`precip_noise`**). Итоговое значение осадков – сумма двух получившихся векторов (**`observed_precip`**).**

Задание 3. Постройте два графика в одних осях – график реального значения осадков (`precip`**) и график наблюдаемых значений (**`observed_precip`**).**

Давайте зададим значения `Smax` и `K` для нашей модели. `K` может принимать любые значения от 0 до 1, а вот `Smax` может быть вообще любым. В этом упражнении предлагаем выбрать значение `Smax` между 60 и 100 мм.

Задание 4. Задайте значения `Smax` **и** `K`**.**

Теперь пропишем основные параметры нашей модели, посчитаем данные утечки (`leak`), перелива (`spill`), общего стока воды из нашего ведра (`runoff=spill+leak`) и объем ведра (`storage`).

Поскольку нам необходимо пересчитывать нашу систему несколько раз, то создадим отдельную функцию в отдельном файле, которая будет автоматически выполнять заданные действия, и нам не придется копировать код несколько раз. С помощью функций можно задавать переменные, возвращать их значения из кода. Функции сохраняются в отдельном файле MatLab и вызываются по их названию. Подробнее про функции и примеры их использования вы можете почитать в справке.

Создадим функцию, которая будет выдавать значения общего стока и объема, используя в качестве входных данных количество осадков, максимальный объем системы и скорость вытекания воды:

```
function [runoff,storage] = bucketmodel(precip,Smax,K)
```

Далее зададим необходимые нам массивы:

```
n = length(precip);
storage = nan(n,1);
spill = nan(n,1);
leak = nan(n,1);
runoff = nan(n,1);
```

Теперь посчитаем объем воды и общий сток за первый день. Будем считать, что изначально наше ведро пустое и доступный уровень воды (`a_water`) равен количеству осадков за первый день.

Задание 5. Используя цикл условия (`if…else`**), заполните массивы** `storage`, `spill`, `leak`, `runoff`**. Если доступный уровень воды меньше или равен** `Smax`**, то перелива не происходит и переменная** `spill` **равна нулю, а утечка равна произведению** `K` **на доступный уровень воды. Если доступный уровень воды больше** `Smax`**, то перелив равен разнице между доступным уровнем воды и** `Smax`**, а утечка равна произведению** `Smax` **и** `K`**. Посчитайте общий сток за первый день как сумму утечки и перелива. Посчитайте объем ведра как разницу между уровнем доступной воды и общего стока.**

Задание 6. Создайте цикл `for`**, который заполнит массивы** `storage, spill, leak, runoff` **на все дни. Начаться он должен так:**

```
for t = 2:n
    a_water = storage(t-1) + precip(t);
```

На этом код нашей функции заканчивается. Сохраните функцию отдельным файлом с соответствующим названием – `bucketmodel`.

Вернемся к нашему основному коду. Вызовите нашу функцию, чтобы получить данные о стоке и объеме модели:

```
[runoff,storage] = bucketmodel(precip,Smax,K);
```

Теперь у нас есть данные о итоговом стоке из нашей реки, но они также получены без погрешности. Добавим погрешность к полученным данным:

Задание 8. Добавьте погрешность (`runoff_noise`**) с помощью команды** `normrnd` **и создайте случайное распределение погрешности осадков в течение 100 дней (среднее =0 и стандартное отклонение =2). Итоговое значение стока – сумма двух получившихся векторов (**`observed_runoff`**).**

Задание 9. Постройте два графика в одних осях – график реального значения стока (`runoff`**) и график наблюдаемых значений (**`observed_runoff`**).**

ЧАСТЬ 2. КАЛИБРОВКА ГИДРОЛОГИЧЕСКОЙ МОДЕЛИ

У нас сейчас есть реальные данные об осадках и стоке, а также данные, которые были получены с помощью наблюдений. В реальной жизни у нас есть только данные от наблюдений и реальные значения нам неизвестны, поэтому в этом разделе попробуем откалибровать нашу модель таким образом, чтобы минимизировать погрешность наблюдений.

Можно определить различные показатели для калибровки моделей, обычно эти параметры называются целевыми функциями. Одной из таких функция является среднеквадратичное отклонение (СКО). Настроим параметры модели таким образом, чтобы минимизировать СКО.

Давайте поменяем входные данные о скорости вытекания воды и максимального объема системы и посчитаем смоделированный сток и объем с помощью нашей функции. Поскольку обычно реальные значения осадков неизвестны, то в качестве входных данных используем наблюдаемые нами значения:

```
K_man = 0.45;
Smax_man = 50;
[simulated_runoff_man,simulated_storage] =
bucketmodel(observed_precip,Smax_man,K_man);
```

Теперь посчитаем СКО значений общего стока:

```
simulation_error = observed_runoff-simulated_runoff_man;
squared_sim_error = simulation_error.^2;
RMSE_man = sqrt(mean(squared_sim_error))
```

Задание 10. Постройте графики `observed_runoff` **и** `simulated_runoff_man` **в одном окне. Поменяйте значения** `K_man` **и** `Smax_man` **и посмотрите, как при этом будет меняться значение СКО.**

Кроме ручной калибровки системы с помощью входных параметров, можно применить также автоматическую калибровку с помощью инструментов оптимизации. Поскольку наша цель — минимизировать значение СКО, используем алгоритм оптимизации для автоматического изменения параметров и минимизации значения СКО. Первым шагом является определение целевой функции, которая может быть автоматически рассчитана алгоритмом оптимизации:

```
RMSE_function_observed = @(x) sqrt(mean((observed_runoff
- bucketmodel(observed_precip,x(1),x(2))).^2));
```

Используем встроенную функцию оптимизации MatLab для настройки параметров нашей модели с функцией `fminsearch`.

Задание 11. Посмотрите в справке назначение функции `fminsearch` **и как она задается.**

Зададим изначальные параметры алгоритма оптимизации и найдем оптимальные параметры максимального объема и скорости вытекания воды:

```
initial_guess = [50,0.4];
[optimum_param,RMSE_auto] =
fminsearch(RMSE_function_observed,initial_guess);
Smax_auto = optimum_param(1);
K_auto = optimum_param(2);
```

Задание 12. С помощью функции `bucketmodel` **получите данные о оптимизированном общем стоке** `simulated_runoff_auto` **(входными данными функции будут** `Smax_auto` **и** `K_auto`**). Постройте графики** `simulated_runoff_auto` **и** `runoff` **в одном окне.**

Теперь оптимизированные данные общего стока должны больше совпадать с реальными значениями без шумов.

Наконец, давайте узнаем реальные параметры нашей модели, используя реальные данные `runoff` и `precip`.

Задание 13. Повторите действия, прописанные после задания 10, с реальными данными, а не наблюдаемыми. Назовите выходные переменные `Smax_auto_ideal` **и** `K_auto_ideal`.

Построим таблицу со всеми найденными нами параметрами системы:

```
Ks = [K;K_man;K_auto;K_auto_ideal];
smaxs = [Smax;Smax_man;Smax_auto;Smax_auto_ideal];
RMSEs = [nan;RMSE_man;RMSE_auto;RMSE_ideal];
T =
table(Ks,smaxs,RMSEs,'VariableNames',{'K','Smax','RMSE'}
,'RowNames',{'True';'Manual';'Automatic';'Ideal'})
```

Какие идеальные параметры вашей системы?

Кейс 9. Диффузия жидкостей

Задачи:

1) Построение графиков.
2) Ввод математических формул.

Это практическое задание посвящено диффузии одной жидкости в другой жидкости. Например, если добавить каплю красного красителя в тарелку с водой, то краситель будет медленно диффундировать, а красное пятно со временем станет больше (рисунок 9.1).

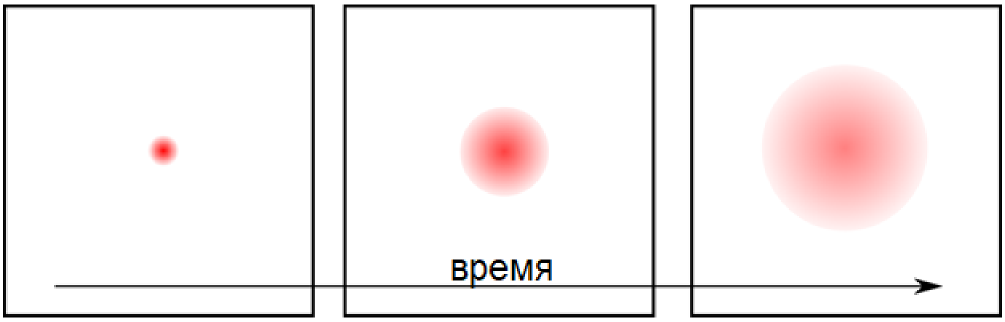


Рисунок 9.1 – Диффузия капли красного красителя в воде

Рассмотрим классический случай диффузии —определенная масса растворенного вещества (M) целиком (вся капля единовременно) помещается в одну точку пространства (x0) в момент времени (t0). Если диффузия происходит только в одном измерении (1D), то концентрация (C) определяется по формуле:

$$C(x,t) = \frac{M}{\sqrt{4\pi D(t-t_0)}}\exp\left(-\frac{(x-x_0)^2}{4D(t-t_0)}\right).$$

Коэффициент диффузии (D) определяет скорость, с которой происходит распространение.

Если диффузия происходит в двух измерениях (2D) (как на рисунке выше) и капля падает в точке (x0, y0) в момент времени t0, то решение для концентрации равно:

$$C(x,y,t) = \frac{M}{\sqrt{4\pi D_x(t-t_0)}\sqrt{4\pi D_y(t-t_0)}}\exp\left(-\frac{(x-x_0)^2}{4D_x(t-t_0)}\right)\exp\left(-\frac{(y-y_0)^2}{4D_y(t-t_0)}\right).$$

В этом случае есть вероятность, что коэффициент диффузии в направлении x (Dx) и коэффициент диффузии в направлении y (Dy) могут быть разными.

Рассмотрим оба случая более подробно:

Задание 1. Капля массой 1 грамм распространяется в одном измерении, начиная с точки x=0 и t=0. Коэффициент диффузии D = $1{\cdot}10^{-5}$ м^2/с. Постройте распределение концентрации капли в пространстве в течение одного часа с момента попадания капли в воду.

Задание 2. Та же самая капля, но теперь она распространяется в двух измерениях из точки x=y=0. Коэффициенты диффузии равны Dx = $1{\cdot}10^{-5}$ м^2/с и Dy = $2{\cdot}10^{-5}$ м^2/с. Постройте трехмерный график распределения концентрации капли в пространстве в течение одного часа с момента попадания капли в воду.

[Подсказка: используйте функцию `meshgrid`**, чтобы создать двумерную сетку, на которой можно оценить экспоненциальную функцию].**

Кейс 10. Уравнения состояний идеального и реальных газов

Задачи:
1) Построение графиков.
2) Ввод математических формул.

Это практическое задание посвящено уравнениям состояния идеального и реальных газов и построению изобар.

Уравнение состояния идеального газа выглядит следующим образом:

$$PV = nRT,$$

где P - давление газа (Па), V - занимаемый им объем (м^3), n - количество молей газа, R - универсальная газовая постоянная (8.31 Дж/(моль·К)), T - абсолютная температура (К).

Задание 1. Постройте график зависимости температуры от объема идеального газа по формуле, указанной выше, с n = 1 и давлением P = 1.0, 0.5, 2.0 и 10.0 Па. Все кривые должны быть на одном и том же графике и разных цветов.

Для отображения состояния реальных газов используем уравнение Ван-дер-Ваальса:

$$P = \frac{nRT}{V-nb} - \frac{an^2}{V^2},$$

где a и b – константы, при этом a учитывает силы притяжения между молекулами (давление на стенку уменьшается, так как есть силы, втягивающие молекулы приграничного слоя внутрь), а b учитывает суммарный объём молекул газа.

Задание 2. Постройте график зависимости температуры от объема для всех реальных газов из таблицы ниже с n = 1 и давлением P = 1.0 Па. Все кривые должны быть на одном графике и разных цветов. Также на этом же графике должна быть изобара идеального газа из предыдущего задания. График должен быть подписан и иметь легенду.

Таблица 10.1 – Данные для построения графика зависимости температуры от объема для газов

Газ	*a*	*b*
He	0.0341	0.0237
H_2	0.2461	0.0267
N_2	1.39	0.0391
O_2	1.36	0.0318
CO_2	3.59	0.0427
NH_3	4.17	0.0371

Задание 3. В комментариях в тексте программы напишите, какой из перечисленных газов ближе всего к идеальному?

Кейс 11. Анализ силы ветра

Задачи:

1) Построение графиков, включая разные виды диаграмм.
2) Ввод математических формул.

Это практическое задание посвящено анализу силы ветра для разных ландшафтов: луга, пустыни, городской ландшафт, аэродром.

Сила ветра над землей может быть выражена как функция неровности поверхности и скорости ветра, измеренная на любой высоте в пределах приземного слоя над землей. Она может быть выражена как

$$U_z = \frac{U_*}{k} ln \frac{Z}{Z_0},$$

где: U_* - скорость поперечной волны ветра (м/с), U_z - скорость ветра (м/с) на высоте Z, Z - высота (м), на которой измерялась скорость ветра, Z_0 – аэродинамическая неровность (м), k – 0.4 (постоянная).

Данные о U_z и Z_0 для четырех разных ландшафтов представлены в таблице 11.1.

Таблица 11.1 – Данные по типам ландшафтов

	Луг	Луг	Пустыня	Пустыня	Город	Город	Аэродром	Аэродром
U_z	10	3	20	10	10	5	10	20
Z_0	0.01	0.01	0.001	0.001	0.01	0.01	0.0001	0.0001

Измерения скорости ветра производятся с метеостанций, датчики ветра установлены на высоте 10 м.

Задание 1. Посчитайте значения U_* для всех данных из таблицы. Постройте график с точками полученных значений. Точки для разных ландшафтов должны выглядеть по-разному. Пример графика представлен на рисунке 11.1.

Задание 2. Рассчитайте значения U_* при диапазоне U_z от 0 м/с до 20 м/с и Z_0=0.01. Постройте график U_* от U_z.

Задание 3. Рассчитайте значения U_* при диапазоне Z_0 от 0.0001 до 1 м и U_z=10 м/с. Постройте график U_* от Z_0.

Задание 4. Постройте снова график U_* от Z_0 используя функцию `semilogx`.

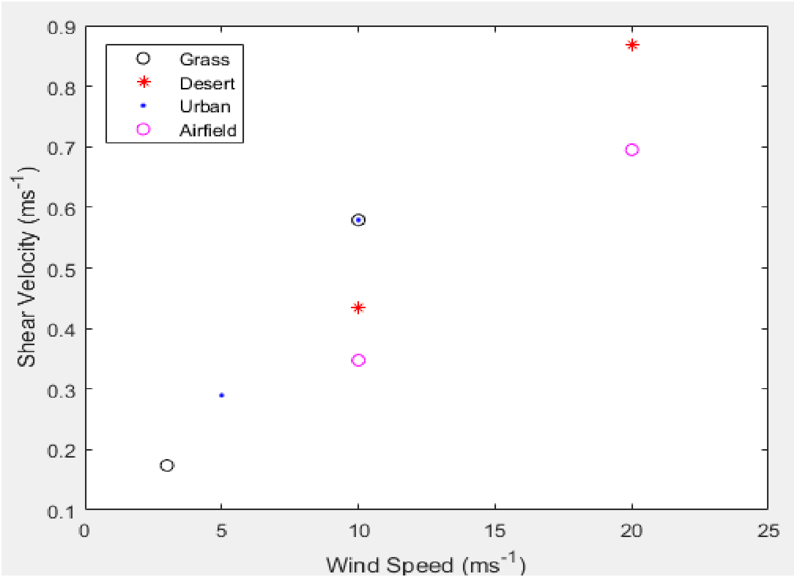


Рисунок 11.1 – Построение графика для разных ландшафтов

Задание 5. Используя функцию `meshgrid` **и** `[C,h] = contourf`, **постройте трехмерный график усиления силы ветра как функции от Z_0 и U_z. Добавьте графику шкалу изменения цвета (**`colorbar`**). Ограничьте ось y значениями от -7 до -1.**

Закончите программу следующим кодом:

```
z0_labels = num2str(exp(-7:1:-1)','%1.4f');
set(gca,'YTickLabel',z0_labels);
clabel(C,h);
```

Итоговый график должен выглядеть так, как на Рисунке 11.2.

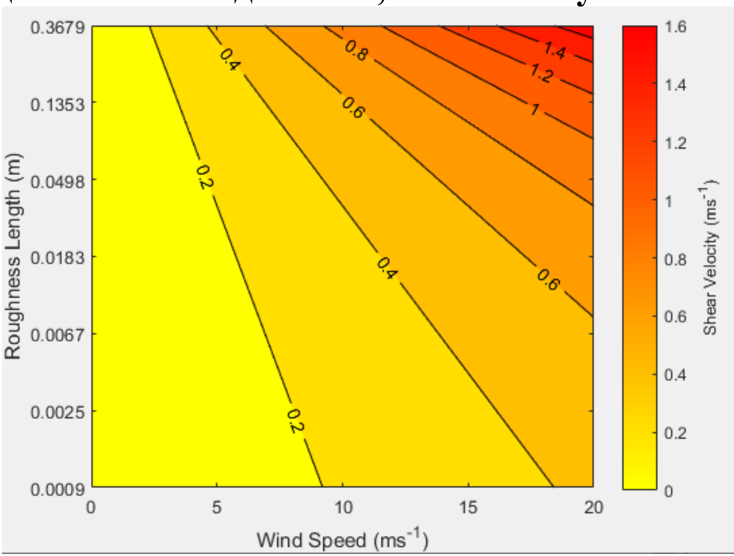


Рисунок 11.2 – Пример итогового графика силы ветра

Итоговый кейс. Моделирование водохранилища

В этом большом итоговом практическом задании предпримем первые шаги в моделировании водохранилища, исходя из климатических данных региона его расположения. Вода попадает в водохранилище из реки. Часть воды испаряется, еще одна часть потребляется для водоснабжения и орошения, также из водохранилища спускается минимально возможное количество воды для сохранения водной экосистемы ниже по течению [8]. Схематичное представление циркуляции воды в водохранилище продемонстрировано на рисунке 12.1.

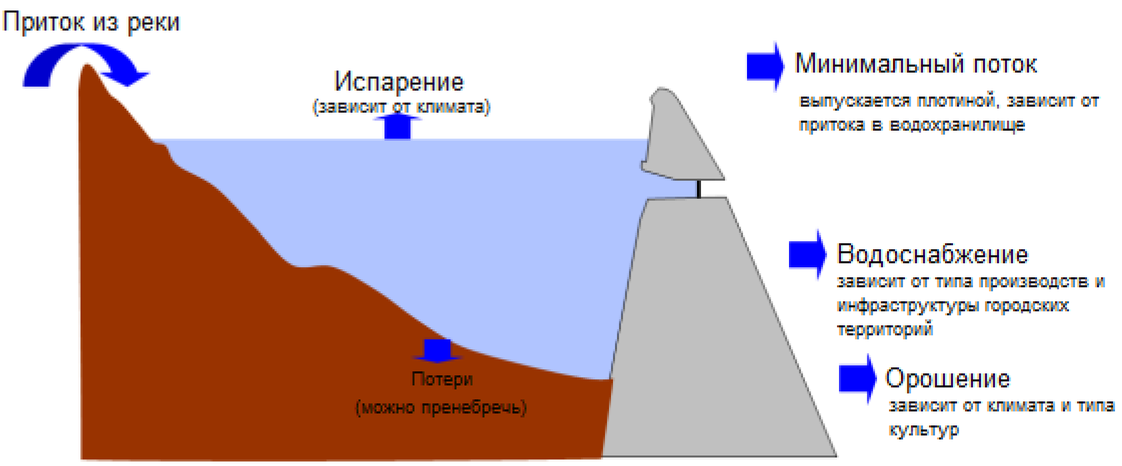


Рисунок 12.1 – Схема циркуляции воды в водохранилище

Задачи:

1. Рассчитать минимальный поток.
2. Рассчитать средний приток из реки для каждого месяца.
3. Рассчитать средний годовой приток.
4. Рассчитать количество осадков.
5. Рассчитать объем испарений.

В качестве входных данных для этого задания у нас есть несколько файлов:

1. `mean_daily_Q.txt` – содержит среднесуточный расход воды водохранилища за период с 01.01.1970 по 31.12.1999 в $м^3/с$.
2. `daily_P.txt` – содержит количество осадков в мм за каждый день за период с 01.01.1984 по 31.12.2008
3. `Max_min_T.txt` – содержит минимумы и максимумы температуры воздуха за каждый день за период с 01.01.1984 по 31.12.2008

Обратите внимание, что в массивах нет данных за 29 февраля.

ЧАСТЬ 1. ГИДРОЛОГИЧЕСКАЯ И КЛИМАТОЛОГИЧЕСКАЯ ХАРАКТЕРИСТИКА УЧАСТКА

Задание 1. Расход воды:

1. Постройте график среднесуточного расхода воды водохранилища за весь период.
2. Постройте кривую продолжительности расхода воды. Графики должны выглядеть примерно таким образом, как на рисунке 12.2.

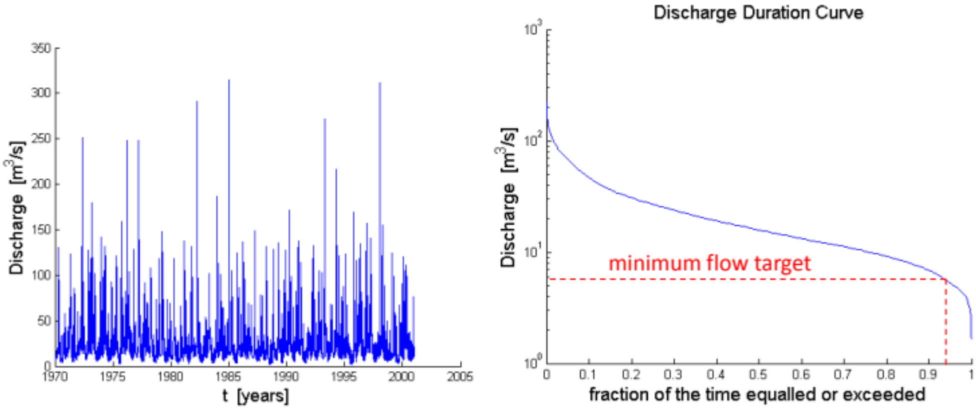


Рисунок 12.2 – Графики среднесуточного расхода воды водохранилища (слева) и кривой продолжительности расхода воды (справа)

3. Минимальный поток составляет объем, который остается спустя 95% времени. Определите его (Q_{347}).

Задание 2. Приток из реки:

1. Вычислите и постройте график среднего расхода для каждого месяца.
2. Вычислите и постройте график стандартного отклонения от расхода для каждого месяца.
3. Вычислите среднегодовое значение притока (общий объем воды, протекающий в среднем за год - одно значение в м^3).

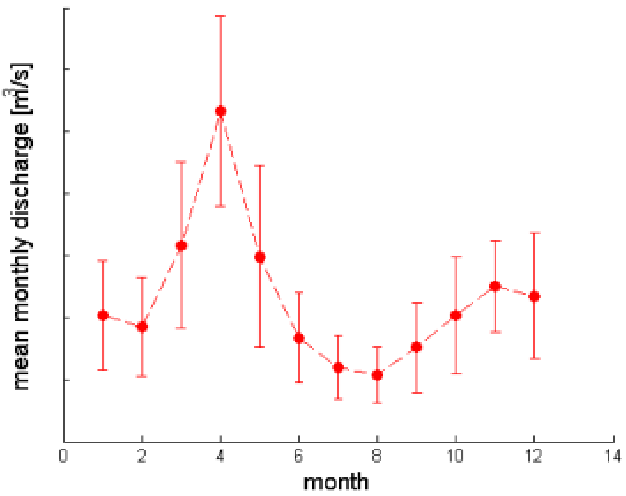


Рисунок 12.3 – Визуализация графиков среднего расхода для каждого месяца и стандартного отклонения от расхода

Задание 3. Осадки:

1. Вычислите и постройте график дневного объема осадков.
2. Вычислите и постройте график средней интенсивности осадков за каждый месяц в мм/день.

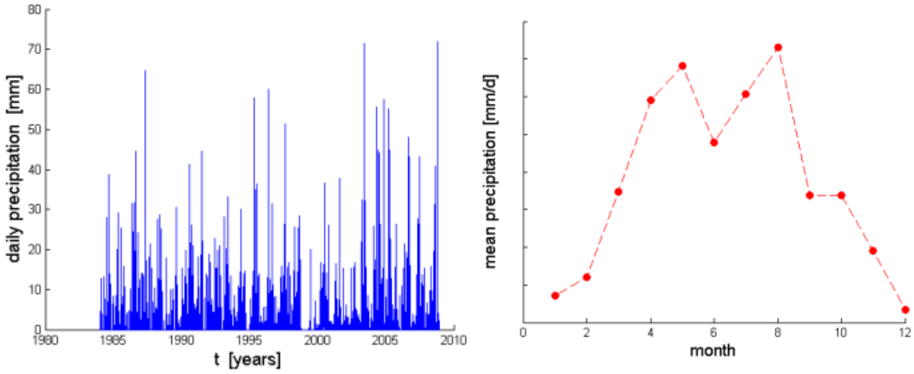


Рисунок 12.4 – Визуализация графиков дневного объема осадков (слева) и средней интенсивности осадков (справа)

Задание 4. Температура:

1. Вычислите и постройте график средней температуры за день (среднее между минимумом и максимумом).
2. Постройте график средней температуры за месяц.

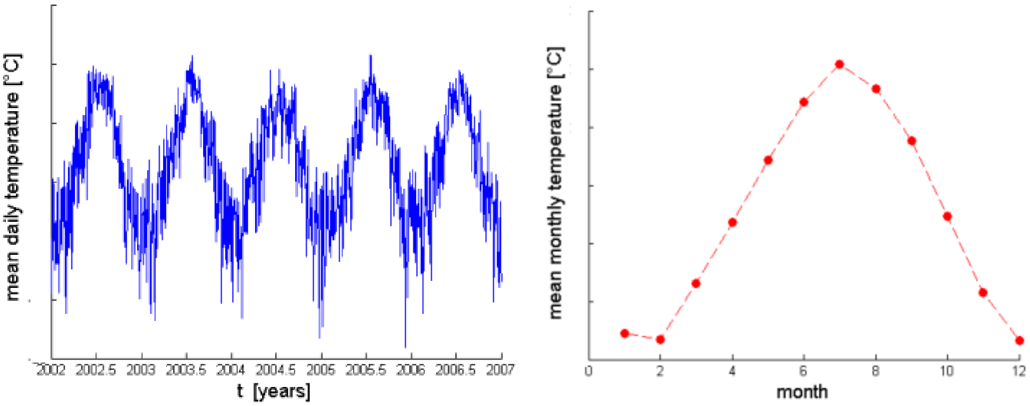


Рисунок 12.5 – Визуализация графиков средней температуры за день (слева) и средней температуры за месяц (справа)

Все задания этого раздела можно выполнить разными способами, но, возможно, вам могут понадобиться следующие команды:

- `sort`
- `semilogy`
- `sum`
- `cumsum`
- `size`
- `zeros`
- `reshape`
- `mean`
- `std`
- `errorbar`
- `sort`
- `find`
- `reshape`

ЧАСТЬ 2. АНАЛИЗ ВОДНОГО БАЛАНСА

Во втором разделе необходимо посчитать водный баланс, то есть оценить расход воды из водохранилища на разные нужды. Для начала оценим потенциальное испарение из водохранилища и определим годовой объем воды, доступный для орошения.

Испарение оценим с помощью уравнения Торнтвейта (1948), которое является широко используемым эмпирическим методом оценки потенциального испарения. Уравнение требует в качестве входных данных только среднюю ежемесячную температуру воздуха и среднюю продолжительность светового дня за каждый месяц, которая может быть рассчитана по широте. В этом задании используем его следующую версию:

$$ET_0(m) = 16\frac{N_m}{12}\left(\frac{10T_m}{I}\right)^a,$$

где $ET_0(m)$ - среднее потенциальное испарение за месяц в мм, N_m - среднее количество светлых часов в месяц m, T_m - средняя температура в месяц m, I – тепловой индекс, вычисляемый по формуле:

$$I = \sum_1^{12} \left(\frac{T_m}{5}\right)^{1.514},$$

a – экспериментальная степень:

$$a = 6.75 \cdot 10^{-7} I^3 - 7.71 \cdot 10^{-5} I^2 + 1.79 \cdot 10^{-2} I + 0.49$$

Количество светлых часов в день можно вычислить по формуле:

$$N_D = \frac{24\omega_s}{\pi}$$

$$\omega_s = cos^{-1}(-\tan(\varepsilon)\tan(\delta))$$

$$\delta = 0.409 \sin\left(2\pi\frac{D}{365} - 1.39\right),$$

где ε - широта, D – количество дней с первого января текущего года.

Среднее количество светлых часов в месяц вы можете получить через количество светлых часов в день и количество дней в каждом месяце.

Немного еще входных данных для расчетов:

Площадь водохранилища составляет 5.5 км^2;

Водозабор на муниципальные нужды - 280000 м^3 в день;

Широта: 37°.

Задание:

1. Посчитайте потенциальное испарение за каждый месяц и постройте график.
2. Посчитайте объем воды, который тратится на минимальный поток в год в м^3.
3. Посчитайте объем воды, который тратится на муниципальные нужды в год в м^3.
4. Посчитайте объем воды, который тратится на испарение за год в м^3, учитывая площадь водохранилища, по формуле:
$$ET_L(m) = 2ET_0(m) \cdot S_L$$
5. Найдите разницу между объемом годового притока в водохранилище и потреблением воды. Предполагая, что объем притока в год равен всему доступному объему воды для всех нужд – найдите объем воды, который можно использовать для орошения.
6. Постройте круговую гистограмму водного баланса (рисунок 12.6).

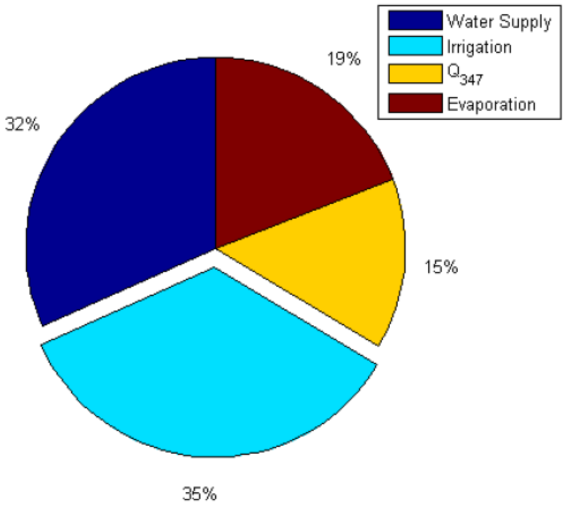


Рисунок 12.6 – Круговая гистограмма водного баланса

ЧАСТЬ 3. МОДЕЛИРОВАНИЕ ОБЪЕМА ВОДОХРАНИЛИЩА

В этой части упражнения посчитаем доступную площадь посевов в зависимости от доступного объема воды для орошения и вычислим объем водохранилища.

Задание:

1. Для начала посчитаем, сколько воды необходимо посевам. Загрузите файл Kc.txt, где находятся данные о дневных коэффициентах расхода влаги культур за один год.
2. Посчитайте среднее значение коэффициентов за месяц (K_C) и постройте график (рисунок 12.7).

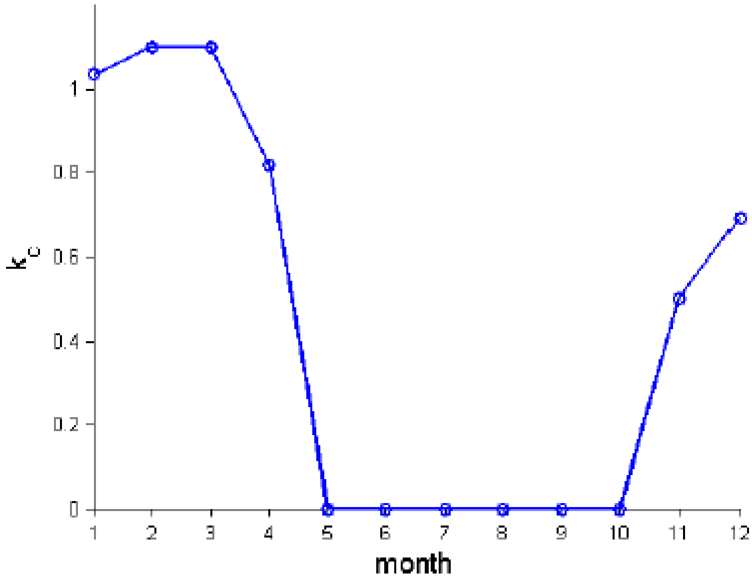


Рисунок 12.7 – График среднего значения коэффициентов за месяц

3. Посчитайте среднее за месяц значение объема воды, необходимой для посевов:
$$ET_C(m) = 2ET_0(m) \cdot K_C.$$
4. Вторым шагом посчитаем эффективный объем осадков, который примем за половину от объема всех осадков.
5. Постройте на одном графике среднее за месяц значение объема воды, необходимой для посевов, и эффективный объем осадков (рисунок 12.8).

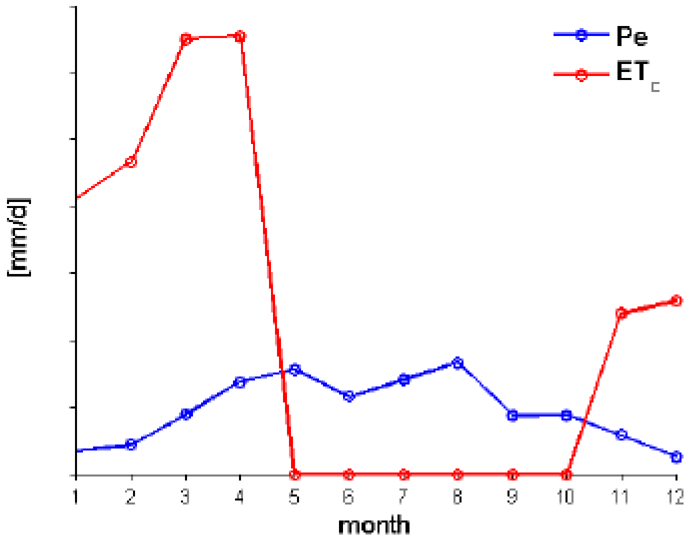


Рисунок 12.8 – Построение графиков среднего за месяц значения объема воды и эффективного объема осадков

6. Посчитайте среднемесячный расход воды на орошение (через эффективность орошения) на единицу площади орошения в мм/день.
   Эффективность орошения = необходимый объем воды/расход воды на орошение = 0.5
7. Посчитайте расход воды на орошение за год на единицу площади в метрах. Зная годовой расход воды, посчитайте максимально возможную площадь посевов.
8. Определите среднемесячные потоки воды и постройте график (рисунок 12.9).

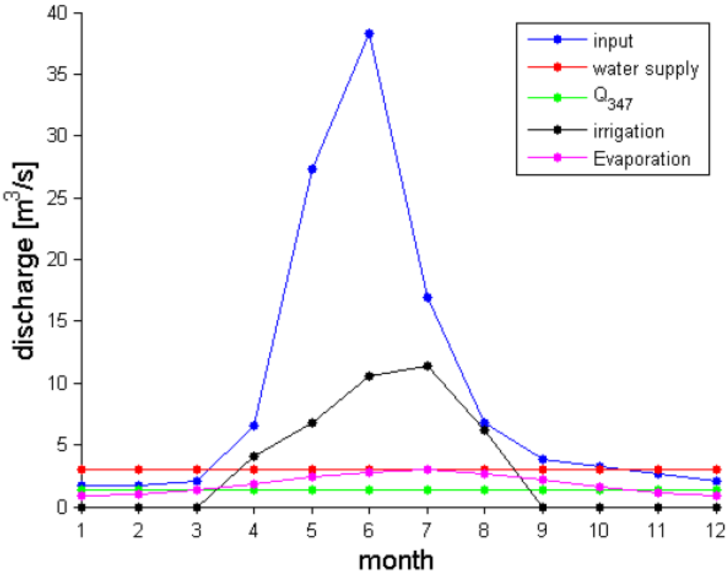


Рисунок 12.9 – График среднемесячных потоков воды

9. Постройте график состояния водохранилища по месяцам в течение года. Для этого вычислите диапазон состояний:

$$V(t) - V(0) = \int_0^t Q_{in}(t')dt' - \int_0^t Q_{out}(t')dt'$$

Объем водохранилища является максимальной разницей в диапазоне состояний (рисунок 12.10).

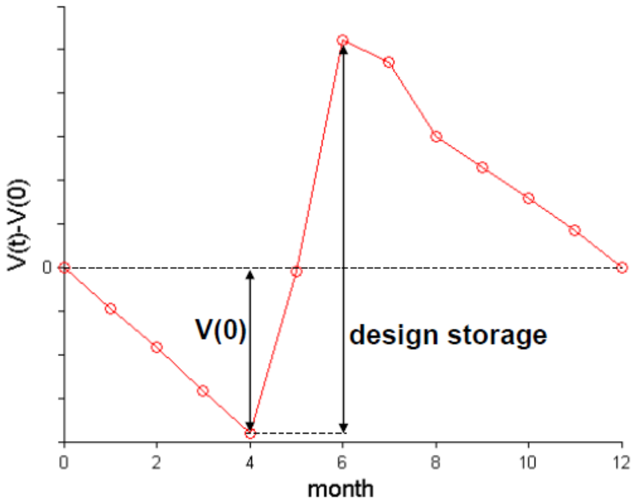


Рисунок 12.10 – График состояния водохранилища по месяцам в течение года

Литература

1. Как построить вертикальные профили солёности и температуры воды на одном графике в MATLAB // KOLDUNOV TIPS блог братьев Колдуновых URL: http://koldunov.ru/kak-postroit-vertikalnye-profili-soljonosti-i-temperatury-vody-na-odnom-grafike-v-matlab/ (дата обращения: 23.01.2023).
2. Climate at a Glance: Global Time Series // NOAA National Centers for Environmental information URL: https://www.ncei.noaa.gov/access/monitoring/climate-at-a-glance/ (дата обращения: 23.01.2023).
3. Laurens Haan, Ana Ferreira. Extreme Value Theory. An Introduction. - 1 изд. - Springer New York, NY, 2006. - 418 с.
4. Fawcett, L. and Walshaw, D. Estimating return levels from serially dependent extremes/Environmetrics, 23,2012 – pp. 272-283 DOI:10.1002/env.2133
5. Булыгина О.Н., Разуваев В.Н., Трофименко Л.Т., Швец Н.В. «Описание массива данных среднемесячной температуры воздуха на станциях России» Свидетельство о государственной регистрации базы данных № 2014621485 URL: http://meteo.ru/data/156-temperature#описание-массива-данных
6. Ryan et al. New Passive Acoustic Monitoring in Monterey Bay National Marine Sanctuary/ IEEE Xplore, 2016 DOI: 10.1109/OCEANS.2016.7761363
7. Yapo, P.O., Gupta, H.V. and Sorooshian, S. Automatic calibration of conceptual rainfall-runoff models: sensitivity to calibration data/ Journal of Hydrology, 181(1-4), 1996 - pp.23-48.
8. Hoboken, N.J. Water resources engineering. Wiley, 2nd ed.; 2011

Маюрова Александра Сергеевна
Быковская Елена Александровна
Тюрикова Екатерина Павловна
Тимофеева Ирина Валерьевна

Методические указания по дисциплине «Ситуационное моделирование»

Учебно-методическое пособие

В авторской редакции

Редакционно-издательский отдел Университета ИТМО

Зав. РИО Н.Ф. Гусарова

Подписано к печати

Заказ №

Тираж

Отпечатано на ризографе

Редакционно-издательский отдел

Университета ИТМО

197101, Санкт-Петербург, Кронверский пр., 49