

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»
Физико-механический институт, Высшая школа теоретической механики и математической физики

(наименование учебного подразделения)

**Отчет о прохождении учебной
(научно-исследовательская работа) практики**
(вид и тип практики)

Качевская Ольга Андреевна
(Ф.И.О. обучающегося)

3 курс бакалавриата группа 5030103/00301
(номер курса обучения и учебной группы)

01.03.03 Механика и математическое моделирование,
01.03.03_03 Математическое моделирование процессов нефтегазодобычи
(Направление подготовки (код и наименование))

Место прохождения практики: ООО «Газпромнефть НТЦ»
(указывается наименование профильной организации или наименование структурного подразделения)

ФГАОУ ВО «СПбПУ», фактический адрес)

Сроки практики: с 10.07.2023 по 21.07.2023

Руководитель практики от ФГАОУ ВО «СПбПУ»: Курта Иван Валентинович, к.т.н.
доцент Высшей школы теоретической механики и математической физики, ФизМех
(Ф.И.О., уч. степень, должность)

Руководитель практики от профильной организации: Печко Константин
Анатольевич, главный специалист НОЦ «Газпромнефть-Политех»
(Ф.И.О., должность)

Оценка: отлично

Руководитель практики
от ФГАОУ ВО «СПбПУ»: / Курта И.В./

Руководитель практики
от профильной организации: / Печко К.А./

Обучающийся: / Качевская О.А./

Дата: 20.06.23

(наименование учебного подразделения)

ИНДИВИДУАЛЬНЫЙ ПЛАН (ЗАДАНИЕ И ГРАФИК) ПРОВЕДЕНИЯ ПРАКТИКИ

Ф.И.О. обучающегося Качевская Ольга Андреевна

Направление подготовки (код/наименование) 01.03.03 Механика и математическое моделирование

Профиль (код/наименование) 01.03.03_03 Математическое моделирование процессов нефтегазодобычи

Вид практики производственная

Тип практики научно-исследовательская работа

Место прохождения практики ООО «Газпромнефть НТЦ»

Руководитель практики от ФГАОУ ВО «СПбПУ»: Курта Иван Валентинович, к.т.н.

доцент Высшей школы теоретической механики и математической физики, ФизМех
(Ф.И.О., уч. степень, должность)

Руководитель практики от профильной организации: Печко Константин Анатольевич,
главный специалист НОЦ «Газпромнефть-Политех»

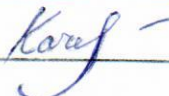
(Ф.И.О., должность)

Рабочий график проведения практики

Сроки практики: с 10.07.2023 по 21.07.2023

№ п/п	Этапы (периоды) практики	Вид работ	Сроки прохождения этапа (периода) практики
1	Организационный этап	Введение в теорию интегрированных моделей, постановка целей и задач, инструктаж по технике безопасности, выдача сопроводительных документов по практике	10.07.2023
2	Основной этап	Получение данных и построение интегрированной модели	10.07.2023– 21.07.2023
3	Заключительный этап	Защита отчета по практике	21.07.2023

Обучающийся



/Качевская О.А./

Руководитель практики
от ФГАОУ ВО «СПбПУ»



/ Курта И.В./

Согласовано:

Руководитель практики
от профильной организации:



/ Печко К.А./

Содержание

Введение.....	4
1. Теоретическая справка	5
2. Создание простейшей интегрированной модели.....	6
2.1 Постановка задачи.....	6
2.2 Схема модели.....	7
2.3 Создание интегрированной модели.....	8
2.4 Выводы по задаче.....	11
Заключение	12
Список использованной литературы.....	13

Введение

Сегодня все передовые нефтегазовые компании применяют интегрированное моделирование для выполнения как оперативных, так и стратегических расчетов по наиболее перспективным месторождениям. Интегрированные модели могут быть разнообразными, однако в данной работе будет рассмотрена интегрированная модель актива (ИМА). В состав ИМА входят такие модели как модель пласта, скважин, а также наземной сети сбора и транспорта углеводородов.

Основу интегрированной модели всегда составляет модель пласта, в частности постоянно действующая геолого-технологическая модель (ПДГТМ). Данный узел является основополагающим, наиболее трудоёмким, с наибольшими геологическими неопределенностями. Модель пласта позволяет решать множество задач. Поэтому именно подход к созданию и адаптации ГДМ имеет ключевую роль при прогнозировании добычи углеводородов (1). Также следует обратить внимание и на модели скважин. Сами скважины по своему назначению подразделяются на несколько видов: опорные, параметрические, структурные, поисковые, разведочные, эксплуатационные, нагнетательные, наблюдательные и специального назначения. Помимо этого, необходимо рассмотреть модель сети сбора и транспорта углеводородом. Под системой сбора и транспорта продукции нефтяных скважин понимают систему оборудования и трубопроводов, построенных для сбора продукции скважин и доставки ее до центрального пункта подготовки нефти (ЦППН). Наиболее типовые системы сбора и транспорта продукции нефтяных скважин: высоконапорная, лучевая и др.

В данной работе будет реализована простейшая интегрированная модель с визуализацией и описанием всех параметров.

1. Теоретическая справка

Для решения поставленных задач воспользуемся языком программирования Python и такими библиотеками и модулями, как:

- json – модуль для кодирования и декодирования данных, основанный на синтаксисе литералов объекта JavaScript.
- pandas – библиотека, используемая для аналитики данных, статистики и Data science (2).
- plotly – графическая библиотека, используема для создания интерактивной графики публикационного качества (3).
- dash – модуль plotly для анализа и визуализации данных.

2. Создание простейшей интегрированной модели

2.1 Постановка задачи

Имеются данные, описывающие интегрированную модель в формате json. В них присутствуют 6 типов элементов, распределенных по 2 массивам: nodes (узел(junction), сток(sink), пласт(source)) и pipes(труба(tube), коннектор(connector), скважина(well)). Для каждого элемента есть набор параметров, который описывает определенный элемент.

```
{
  "gasRateTonneToD": 0.0,
  "id": 112,
  "name": "1112 thp",
  "type": "JUNCTION",
  "oilRateTonneToD": 0.0,
  "waterRateTonneToD": 0.0,
  "on": false
},
{
  "id": 113,
  "inletPressureAtma": 10.0,
  "name": "DNS",
  "type": "SINK"
},
{
  "id": 114,
  "name": "T vr K9",
  "type": "JUNCTION"
},
{
  "id": 115,
  "name": "T vr K7",
  "type": "JUNCTION"
},
```

Рисунок 1. Данные в формате json, описывающие интегрированную модель.

2.2 Схема модели

Из массива данных видно, что все элементы могут быть связаны между собой по определенным параметрам. Также имеется информация о том, в каком примерно порядке должны располагаться элементы. Тогда получим приблизительную схему модели.

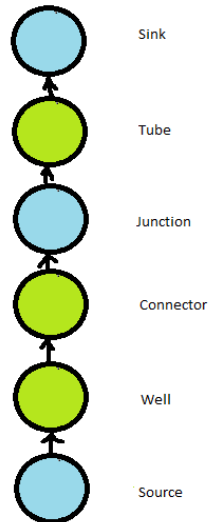


Рисунок 1. Схема модели

Элементы в ней связаны благодаря таким параметрам, как `id`, `type`, `outletNodeId` и `inletNodeId`.

2.3 Создание интегрированной модели.

Для начала импортируем необходимые модули и библиотеки, а также подгрузим базу данных.

```
import json
import pandas as pd
from plotly import *
import dash
from dash.dependencies import Input, Output
import dash_core_components as dcc
import dash_html_components as html
import dash_cytoscape as cyto

app = dash.Dash(__name__)
server = app.server

table = "network_actualized_for_forecast (1) (2).json"

with open(table, 'r', encoding='utf-8') as f:
    table = json.load(f)
```

Рисунок 2. Импортируемые библиотеки, модули и база данных.

После напомним класс функций для добавления элементов в схему, а также для отображения информации об элементах модели.

```
def add_element():
    id_source = int(input('Введите id пласта от 126 до 238\n'))

    if (126 <= id_source <= 238):
        for pipes1 in table["pipes"]:
            if id_source == pipes1["id"] and pipes1["type"] == "WELL":
                #print(f"SOURCE ID: {pipes['id']} --> WELL ID: {pipes['id']}")
                a = pipes1['id']

        for pipes1 in table["pipes"]:
            for pipes2 in table["pipes"]:
                if (pipes1["id"] == id_source and pipes1["type"] == "WELL" and (pipes2["type"] == "CONNECTOR" and pipes1["outletNodeId"] == pipes2["inletNodeId"]):
                    #print(f"SOURCE ID: {pipes['id']} --> WELL ID: {pipes['id']} --> CONNECTOR ID: {pipes1['id']}")
                    b = pipes2['id']

        for pipes2 in table["pipes"]:
            for pipes3 in table["pipes"]:
                if (pipes2["id"] == b and pipes2["type"] == "CONNECTOR" and (pipes3["type"] == "TUBE" and pipes2["outletNodeId"] == pipes3["inletNodeId"]):
                    #print(f"SOURCE ID: {pipes['id']} --> WELL ID: {pipes['id']} --> CONNECTOR ID: {pipes1['id']}")
                    c = pipes3['id']
                    #print(f"SOURCE ID: {a} --> WELL ID: {a} --> CONNECTOR ID: {b} --> JUNCTION ID: {b} --> TUBE ID: {c} --> SINK ID: 113")
```

Рисунок 3. Функция добавления элементов.


```
def get_element():
    id = int(input("Specify id"))
    type = str(input("Specify type(Caps Lock)"))

    for Node in table["nodes"]:
        if (8<=id<=113):
            if id == Node["id"] and type == Node["type"] and Node["type"] == "JUNCTION":
                print(f'id:{Node["id"]}\n type:{Node["type"]}\n name:{Node["name"]}\n on:{Node["on"]}\n gasRateTonneToD:{Node["gasRateTonneToD"]}\n oilRateTonneToD:{Node["oilRateTonneToD"]}\n waterRateTonneToD:{Node["waterRateTonneToD"]}\n inletPressureAtma:{Node["inletPressureAtma"]}\n\n')
            elif id == Node["id"] and Node["type"] == "SINK":
                print(f'id:{Node["id"]}\n type:{Node["type"]}\n name:{Node["name"]}\n inletPressureAtma:{Node["inletPressureAtma"]}\n\n')

        if (114<=id<=125):
            if id == Node["id"] and type == Node["type"] and Node["type"] == "JUNCTION":
                print(f'id:{Node["id"]}\n type:{Node["type"]}\n name:{Node["name"]}\n\n')

        if (126<=id<=238):
            if id == Node["id"] and type == Node["type"] and Node["type"] == "SOURCE":
                print(f'id:{Node["id"]}\n type:{Node["type"]}\n name:{Node["name"]}\n on:{Node["on"]}\n gasRateTonneToD:{Node["gasRateTonneToD"]}\n oilRateTonneToD:{Node["oilRateTonneToD"]}\n waterRateTonneToD:{Node["waterRateTonneToD"]}\n\n')

    for Pipes in table["pipes"]:
        if (8<=id<=6):
            if id == Pipes["id"] and type == Pipes["type"] and Pipes["type"] == "TUBE":
                print(f'id:{Pipes["id"]}\n type:{Pipes["type"]}\n name:{Pipes["name"]}\n inletNodeId:{Pipes["inletNodeId"]}\n outletNodeId:{Pipes["outletNodeId"]}\n\n')
            elif id == Pipes["id"] and type == Pipes["type"] and Pipes["type"] == "TUBE":
                print(f'id:{Pipes["id"]}\n type:{Pipes["type"]}\n name:{Pipes["name"]}\n inletNodeId:{Pipes["inletNodeId"]}\n outletNodeId:{Pipes["outletNodeId"]}\n\n')

        if (7<=id<=11):
            if id == Pipes["id"] and type == Pipes["type"] and Pipes["type"] == "TUBE":
                print(f'id:{Pipes["id"]}\n type:{Pipes["type"]}\n name:{Pipes["name"]}\n inletNodeId:{Pipes["inletNodeId"]}\n outletNodeId:{Pipes["outletNodeId"]}\n\n')
            elif id == Pipes["id"] and type == Pipes["type"] and Pipes["type"] == "TUBE":
                print(f'id:{Pipes["id"]}\n type:{Pipes["type"]}\n name:{Pipes["name"]}\n inletNodeId:{Pipes["inletNodeId"]}\n outletNodeId:{Pipes["outletNodeId"]}\n\n')

        if (12<=id<=124):
            if id == Pipes["id"] and type == Pipes["type"] and Pipes["type"] == "CONNECTOR":
                print(f'id:{Pipes["id"]}\n type:{Pipes["type"]}\n name:{Pipes["name"]}\n inletNodeId:{Pipes["inletNodeId"]}\n outletNodeId:{Pipes["outletNodeId"]}\n\n')
            elif id == Pipes["id"] and type == Pipes["type"] and Pipes["type"] == "CONNECTOR":
                print(f'id:{Pipes["id"]}\n type:{Pipes["type"]}\n name:{Pipes["name"]}\n inletNodeId:{Pipes["inletNodeId"]}\n outletNodeId:{Pipes["outletNodeId"]}\n\n')

        if (125<=id<=237):
            if id == Pipes["id"] and type == Pipes["type"] and Pipes["type"] == "WELL":
                print(f'id:{Pipes["id"]}\n type:{Pipes["type"]}\n name:{Pipes["name"]}\n inletNodeId:{Pipes["inletNodeId"]}\n outletNodeId:{Pipes["outletNodeId"]}\n\n')
            elif id == Pipes["id"] and type == Pipes["type"] and Pipes["type"] == "WELL":
                print(f'id:{Pipes["id"]}\n type:{Pipes["type"]}\n name:{Pipes["name"]}\n inletNodeId:{Pipes["inletNodeId"]}\n outletNodeId:{Pipes["outletNodeId"]}\n\n')
```

Рисунок 5. Функция отображения информации элементов.

В итоге мы получаем последовательность из id элементов, которые связаны между собой по различным параметрам. Далее с помощью библиотеки pandas сделаем из id таблицу, а после напишем код визуализации и получим простейшую интегрированную модель.



```

None
id:126
type:SOURCE
name:1000
on:False
gasRateTonneToD:0.0
oilRateTonneToD:0.0
waterRateTonneToD:0.0

None
id:126
type:WELL
name:1001
inletNodeId:127
outletNodeId:1
VFPModelNumber:1001

None
id:13
type:CONNECTOR
name:13
inletNodeId:1
outletNodeId:120
  
```

```
None
id:13
type:JUNCTION
name:1013 thp
on:False
gasRateTonneToD:0.0
oilRateTonneToD:0.0
waterRateTonneToD:0.0

None
id:6
type:TUBE
name:Kust 9-T vr K9
inletNodeId:120
outletNodeId:114
innerDiameterMm:159
profileHorDistanceMSpaceHeightM:['0 112.33', '2923 116.1']
roughnessMm:1.524e-05

None
id:113
type:SINK
name:DNS
inletPressureAtma:10.0

None
```

2.4Вывод

В результате анализа данных, составления схемы, написания кода по формированию данных, а также их визуализации мы получили достоверную простейшую интегрированную модель инфраструктуры.

Заключение

В данной практической работе было получено общее представление, что такое интегрированная модель, а также о ее применении. Были получены базовые знания и навыки, необходимые для построения простейших интегрированных моделей, отражающих расположение пласта, наземной и подземной инфраструктуры.

Для решения поставленной задачи были использованы язык программирования Python, а также различные библиотеки и модули, благодаря которым мы получили достоверный результат. Таким образом, можно сделать вывод, что применение языков программирования в задачах из нефтегазовой области целесообразно.

Список использованной литературы

1. «Создание и эксплуатация интегрированной модели, учитывающей особенности газоконденсатной залежи» В. Г. Зипир, А. Г. Менгалиев, А.Г.Рясный, М.Н. Курбатова, О.М. Рудая
https://earchive.tpu.ru/bitstream/11683/73806/1/bulletin_tpu-2022-v333-i10-15.pdf
2. «Руководство пользователя по применению библиотеки pandas»
https://pandas.pydata.org/docs/user_guide/index.html#user-guide
3. «Руководство пользователя по применению библиотеки plotly»
<https://plotly.com/python/>