

**Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
Санкт-Петербургский горный университет**

**Кафедра информатики и компьютерных технологий**

**ИНФОРМАТИКА**  
**МОДЕЛИРОВАНИЕ СВОЙСТВ НЕФТИ**  
**КАК ДИСПЕРСНОЙ СИСТЕМЫ**

*Методические указания к курсовой работе  
для студентов специальности 21.03.01*

**САНКТ-ПЕТЕРБУРГ**  
**2020**

УДК 519.876.5 (073)

**ИНФОРМАТИКА. Моделирование свойств нефти как дисперсной системы:** Методические указания к курсовой работе / Санкт-Петербургский горный университет. Сост.: *С.Б. Крыльцов, М.А. Коробицына*. СПб, 2020. 26 с.

В методических указаниях приведено задание на выполнение курсовой работы по моделированию движения нефтесодержащих продуктов по трубопроводу, выполняемому с помощью программирования на языке Python с использованием пакета matplotlib для трёхмерной визуализации. В методических указаниях также приведены основы моделирования системы частиц в ограниченном объеме, описание каждого элемента системы сопровождается листингами с кодом.

Научный редактор доц. *А.Б. Маховиков*

Рецензент канд. техн. наук *К.В. Столяров* (компания “Telum Inc.”)

I

## ВВЕДЕНИЕ

Подготовка студентов по специальности 21.03.01 «Нефтегазовое дело» предполагает получение всесторонних знаний о нефтесодержащей продукции – начиная от её добычи, и заканчивая переработкой на объектах топливно-энергетического комплекса. Ввиду удалённости мест добычи нефти от потребителей нефтесодержащей продукции одной из актуальных задач в нефтегазовом деле является транспортировка нефтесодержащей продукции по нефтепроводам.

При перемещении нефтесодержащей продукции на большие расстояния актуальным становится представление нефтепродуктов в виде дисперсной системы.

Дисперсная система в общем случае представляет собой образование из двух или более тел, между которыми не происходит химических реакций. Нефтесодержащие продукты представлены системой из веществ – дисперсных фаз, которые мелко распределены в дисперсионной среде.

Течение дисперсных систем описывается законами гидродинамики сплошных сред, а также основывается на расслоении, массопереносе и другим физическим явлениям, приводящим к структурному изменению дисперсных фаз.

Описание таких систем является сложной задачей, при решении которой чаще всего применяется ряд допущений. Дополнительная сложность возникает при описании дисперсной системы нефтепродуктов, перемещающихся в турбулентных течениях, где поведение частиц целиком определяется их содержанием в единице объема и размерами.

Несмотря на наличие широкого спектра программных продуктов, предназначенных для численного моделирования процессов теплообмена, массопереноса и взаимодействия частиц, зачастую необходимо понимать, как происходит сам процесс моделирования. Язык программирования Python при использовании библиотек для визуализации данных позволяет достаточно просто разработать своё приложение для моделирования поведения дисперсной системы.

## ОГЛАВЛЕНИЕ

Введение.....	3
Рекомендуемое программное обеспечение.....	5
Моделирование простейшей дисперсной системы.....	7
Создание класса дисперсной частицы.....	7
Линейное движение частицы по заданному объёму.....	8
Моделирование системы частиц.....	11
Добавление внешнего воздействия на частицы.....	14
Взаимодействие частиц.....	15
Задание на курсовую работу.....	18
Варианты к курсовой работе.....	19
Заключение.....	24
Библиографический список.....	25

## РЕКОМЕНДУЕМОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Для выполнения курсового проектирования в соответствии с данными методическими указаниями минимально необходимое программное обеспечение включает в себя: интерпретатор языка Python версии 3.6 и выше, интерактивную вычислительную среду на основе веб-интерфейса Jupyter Notebook для Python3, программную библиотеку для обработки и анализа данных pandas для Python3. Для упрощения процесса установки необходимых библиотек для Python целесообразно установить Anaconda – свободно распространяемый набор программного обеспечения для статистической обработки данных, машинного обучения, формирования аналитики с использованием средств языков программирования Python и R. Для установки Anaconda под операционной системой (ОС) Windows необходимо загрузить и выполнить инсталлятор с официального сайта <https://www.anaconda.com/> выбрав версию релиза для индивидуального использования – «Individual Edition» – под 32- либо 64-разрядную архитектуру, в зависимости от архитектуры ОС.

При установке Anaconda рекомендуется использовать параметры, представленные на рисунке 1, во избежание конфликтов с другими версиями интерпретатора Python, установленными в системе.

В методических указаниях используются следующие версии программного обеспечения:

1. Anaconda 3 (версия 2020.02).
2. Excel 2019 (версия 16.0.6742.2048).

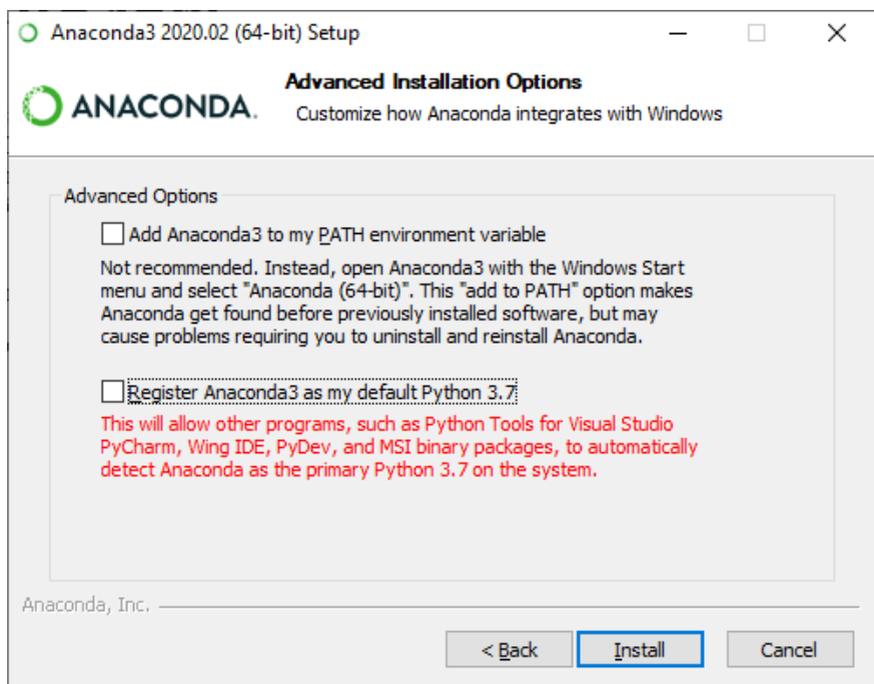


Рис. 1 – Рекомендуемые параметры для установки Anaconda 3

## **МОДЕЛИРОВАНИЕ ПРОСТЕЙШЕЙ ДИСПЕРСНОЙ СИСТЕМЫ**

Дисперсная система представлена частицами, распределёнными в некотором объёме. Их взаимодействие при перемещении по заданному объёму рассчитывается для набора временных промежутков в каждой единице объёма.

Приложение для моделирования простейшей дисперсной системы целесообразно разрабатывать в соответствии с объектно-ориентированным подходом, в котором можно выделить класс частиц и класс системы частиц.

Объекты класса частиц рекомендуется делать максимально самодостаточными, таким образом, что при передаче им параметров трубопровода, в момент инициализации они занимают случайное положение внутри объёма трубопровода и получают начальные параметры, приведённые в вариантах к курсовой работе.

Объект класса системы частиц же в свою очередь содержит в себе объекты частиц в виде списка, и на каждом этапе расчёта системы изменяет положение частиц в соответствии с законами их взаимодействия как между собой, так и с элементами трубопровода и дисперсионной средой.

### **СОЗДАНИЕ КЛАССА ДИСПЕРСНОЙ ЧАСТИЦЫ**

Для моделирования дисперсной системы целесообразно создать класс частиц, объекты которого будут иметь поля, регистрирующие текущее положение частицы в 3d пространстве, массу частицы, а также вектор скорости перемещения частицы. Вектор скорости частицы может изменяться в зависимости от воздействия внешних факторов, а также взаимодействия частиц между собой либо взаимодействия со стенками трубопровода и препятствиями.

```

1 class Particle:
2     def __init__(self):
3         self.x = randint(1, 100)
4         self.y = randint(1, 100)
5         self.z = randint(1, 100)
6         self.dx = randint(1, 10)
7         self.dy = randint(1, 10)
8         self.dz = randint(1, 10)
9         self.xs = []
10        self.ys = []
11        self.zs = []

```

Рис. 2 – Создание класса частицы

## ЛИНЕЙНОЕ ДВИЖЕНИЕ ЧАСТИЦЫ ПО ЗАДАННОМУ ОБЪЁМУ

Предположим, что созданная частица должна перемещаться по заданному кубическому объёму, не имея возможности его покинуть. Тогда на каждом этапе обчёта системы необходимо изменять текущие координаты частицы в соответствии с её вектором движения.

Напишем функцию для изменения координаты в зависимости от достижения пределов заданного объёма, которая в качестве аргументов принимает текущее положение частицы по одной из осей, её изменение положения на следующем этапе обчёта системы и максимальную координату по данной оси:

```

1 def move(value: int, d_value: int, value_max):
2     return value + d_value if value + d_value <=
   value_max else value_max

```

Рис. 3 – Функция для изменения координаты частицы в пределах заданного объёма

Используя созданную функцию добавим метод `move()` класса `Particle`, изменяющий положение частицы в трёхмерном пространстве, и сохраняющий новые координаты в списки с историей перемещения частицы:

```
1     def move(self):
2         self.x = move(self.x, self.dx, 100)
3         self.xs.append(self.x)
4         self.y = move(self.y, self.dy, 100)
5         self.ys.append(self.y)
6         self.z = move(self.z, self.dz, 100)
7         self.zs.append(self.z)
```

Рис. 4 – Метод для изменения координат частицы в пределах заданного объёма

Теперь, когда каждая созданная частица случайным образом располагается внутри заданного объёма, а также имеет случайный вектор перемещения по осям `xuz`, становится возможным осуществить визуализацию её перемещения средствами модуля `matplotlib`.

Для начала необходимо импортировать модуль визуализации добавлением в начало файла кода, приведённого на рисунке 5.

```
1 import matplotlib.pyplot as plt
```

Рис. 5 – Добавление средства визуализации из модуля `matplotlib`

Работа с модулем `ruplot` заключается в создании объекта `figure`, на котором будет отрисовываться диаграмма движения частицы в 3d пространстве. Для этого необходимо в конец файла с исходным кодом добавить код из листинга на рисунке 6.

```
1 fig = plt.figure()
2 ax = fig.gca(projection='3d')
3 ax.set_xlabel('X')
```

```
4 ax.set_ylabel('Y')
5 ax.set_zlabel('Z')
6
7 p1 = Particle()
8 for t in range(10):
9     p1.move()
10
11 ax.plot(p1.xs, p1.ys, p1.zs, 'bo-')
12 plt.show()
```

Рис. 6 – Код для отрисовки 10 положений частицы внутри объёма куба размером 100 условных единиц

В листинге на рисунке 6 создаётся объект Figure, содержащий объект визуализации диаграмм и графиков Plot, а также объект ax, содержащий информацию об осях графика. Для объекта ax задётся трёхмерная система координат и добавляются подписи к осям.

Plot в качестве аргументов принимает массив значений x, y, z, которые необходимо отрисовать на графике. Также для Plot возможно задать опциональные параметры отрисовки линий. Значение 'bo-' означает, что линии будут отрисованы синим цветом ('b' – blue), точки отрисовки будут выделены кружочком ('o') и соединены прямыми линиями ('-').

Результат выполнения кода из рисунка 6 приведён на рисунке 7. Как видно из получившегося графика, частица достигая границ куба размером 100 x 100 x 100 условных единиц продолжает движение вдоль его стенок.

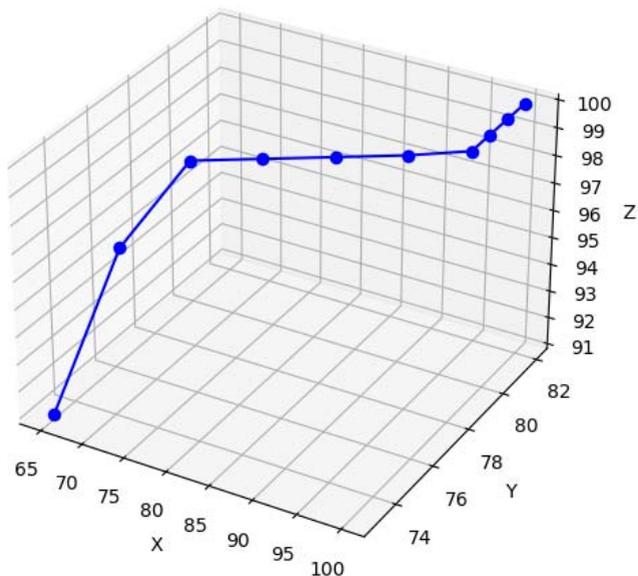


Рисунок 7 – Визуализация движения частицы внутри объёма куба размером 100 x 100 x 100 условных единиц

## МОДЕЛИРОВАНИЕ СИСТЕМЫ ЧАСТИЦ

Система частиц в данной курсовой работе представлена объектом, содержащим частицы и обесчитывающим их положения на каждом этапе моделирования.

Элементарная система частиц может быть описана классом `ParticlesSystem`, код которого приведён в листинге на рисунке 8. В качестве входного параметра объект класса `ParticlesSystem` принимает количество частиц в системе, которые создаются в момент инициализации объекта.

```

1 class ParticlesSystem:
2     def __init__(self, particles_num: int):
3         self.particles = []
4         for i in range(particles_num):
5             self.particles.append(Particle())
6
7     def process(self):
8         for particle in self.particles:
9             particle.move()
10
11    def render(self):
12        fig = plt.figure()
13        ax = fig.gca(projection='3d')
14        ax.set_xlabel('X')
15        ax.set_ylabel('Y')
16        ax.set_zlabel('Z')
17        colors = ['r', 'g', 'b']
18        n = 0
19        for particle in self.particles:
20            ax.plot(particle.xs, particle.ys,
21                  particle.zs, colors[n] + 'o-')
22            n = n + 1 if n < 2 else 0
23        plt.show()

```

Рис. 8 – Код для отрисовки 10 положений частицы внутри объёма куба размером 100 x 100 x 100 условных единиц

Так как каждая частица при инициализации получает случайные координаты из объёма куба, класс `ParticlesSystem` ответственен только за перемещение всех частиц внутри объёма, для чего при каждом вызове метода `process()` для всех частиц системы вызывается метод `move()`.

Также объект класса `ParticlesSystem` включает в себя метод `render()`, который отрисовывает траекторию движения всех частиц чередуя для них цвета из списка с красным, зелёным и синим цветом.

Система из 10 частиц для 50 этапов моделирования может быть визуализирована с помощью кода, приведённого на рисунке 9.

```
1 ps = ParticlesSystem(10)
2 for i in range(50):
3     ps.process()
4 ps.render()
```

Рис. 9 – Код для визуализации перемещения 10 частиц системы на 50 этапах моделирования

Результаты выполнения кода из рисунка 9 приведены на рисунке 10.

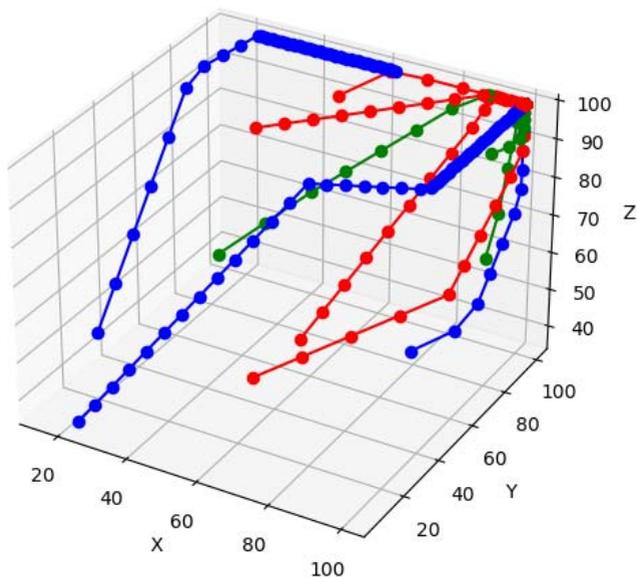


Рис. 10 – Визуализация движения 10 частиц внутри объёма куба размером 100 x 100 x 100 условных единиц

## ДОБАВЛЕНИЕ ВНЕШНЕГО ВОЗДЕЙСТВИЯ НА ЧАСТИЦЫ

При перемещении частиц внутри трубопровода на них оказывается влияние дисперсионной среды. Данное влияние может быть смоделировано изменением составляющих перемещения каждой частицы на следующем этапе моделирования.

В качестве примера добавим синусоидальный закон изменения составляющей  $dx$  для каждой частицы. Для этого необходимо добавить к системе частиц поле `current_step`, которое содержит в себе текущий этап моделирования, а метод `process()` необходимо изменить в соответствии с кодом, приведённым в листинге на рисунке 11.

```
1     def process(self):
2         for particle in self.particles:
3             particle.dx +=
4             int(sin(self.current_step/2)*4)
5             particle.move()
6             self.current_step += 1
```

Рис. 11 – Код для визуализации перемещения 10 частиц системы на 50 этапах моделирования

Для демонстрации процесса визуализации синусоида рассчитывается исходя из значения текущего этапа моделирования, поделённого на 2, а амплитуда перемещения составляет 4 условных единицы.

Результаты выполнения кода, приведённого на рисунке 11 приведены на рисунке 12.

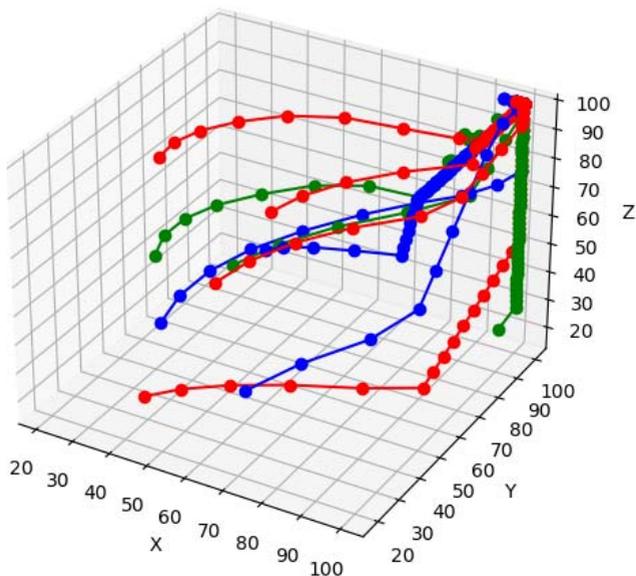


Рис. 12 – Визуализация движения 10 частиц внутри объёма куба размером 100 x 100 x 100 условных единиц с моделированием внешнего воздействия на частицы, приводящего к синусоидальному изменению их координат по оси x

## ВЗАИМОДЕЙСТВИЕ ЧАСТИЦ

Для описания взаимодействия между собой необходимо изменить метод `process()` объекта класса `ParticlesSystem` таким образом, чтобы на каждом этапе моделирования, он для каждой частицы также учитывал положение других частиц.

В качестве примера рассмотрим столкновение частиц, при котором частицы обмениваются составляющими  $dx$ ,  $dy$ ,  $dz$  с противоположным знаком. Для этого изменим метод `process()` в соответствии с кодом, приведённым в листинге на рисунке 13.

```

def process(self):
    for particle in self.particles:
        particle.dx +=
1 int(sin(self.current_step/2)*2)
2         particle.move()
3         for other_particle in self.particles:
4             if particle != other_particle:
5                 if particle.x == other_particle.x
6 and particle.y == other_particle.y and particle.z ==
7 other_particle.z:
8                     particle.dx, other_particle.dx
9 = -other_particle.dx, -particle.dx
10                    particle.dy, other_particle.dy
11 = -other_particle.dy, -particle.dy
                    particle.dz, other_particle.dz
                = other_particle.dz, -particle.dz
                self.current_step += 1

```

Рис. 13 – Код для визуализации перемещения 10 частиц системы на 50 этапах моделирования

Результаты выполнения метода `render()` после изменения метода `process()` в соответствии с кодом, приведённым на рисунке 13, представлены на рисунке 14. Как видно из рисунка, теперь при соударении частиц в точке с координатами, близкими к 100, 100, 100, частицы отскакивают друг от друга и двигаются в обратную сторону.

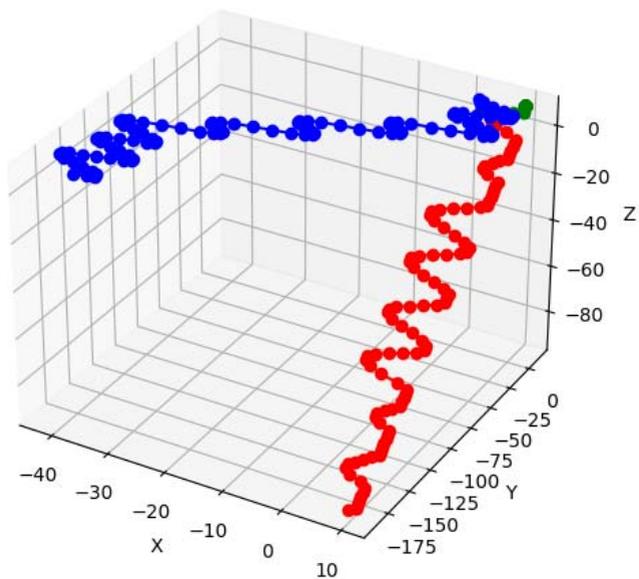


Рис. 14 – Визуализация движения 10 частиц внутри объёма куба размером 100 x 100 x 100 условных единиц с моделированием внешнего воздействия на частицы, приводящего к синусоидальному изменению их координат по оси x

## ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

В курсовой работе необходимо средствами Python произвести моделирование взаимодействия частиц внутри трубопровода, представляющего из себя цилиндр с заданными параметрами.

Частицы должны взаимодействовать в соответствии с законом сохранения импульса как физические тела с нулевыми размерами. При соударении со стенками трубопровода либо с другими частицами выделяется энергия, зависящая от массы частицы нефтепродукта.

На частицы оказывает влияние дисперсионная система, заставляя их двигаться по спиральной траектории, параметры которой приведены в вариантах к заданию.

Внутри трубопровода должно быть расположено препятствие с формой и толщиной, приведёнными в вариантах к заданию, с которым частицы также должны взаимодействовать.

## ВАРИАНТЫ К КУРСОВОЙ РАБОТЕ

### Вариант 1

Длина трубопровода (усл. ед.): 1000.

Диаметр трубопровода (усл. ед.): 50.

Количество частиц: 100.

Собственная скорость частиц (усл. ед./этап): 10.

Диаметр спирали воздействия дисперсионной среды (% от диаметра трубопровода): 20.

Случайное изменение перемещения частиц вследствие воздействия дисперсионной среды (% от  $dx$ ,  $dy$ ,  $dz$ ): 10.

Диапазон условных масс частиц (усл. ед.): 5-15.

Расположение препятствия по оси  $x$  от начала трубопровода (% от длины трубопровода): 20.

Форма препятствия: кольцо.

Толщина препятствия (усл. ед.): 100.

Площадь перекрытия сечения трубопровода препятствием (% от площади сечения трубопровода): 50.

### Вариант 2

Длина трубопровода (усл. ед.): 1500.

Диаметр трубопровода (усл. ед.): 100.

Количество частиц: 200.

Собственная скорость частиц (усл. ед./этап): 15.

Диаметр спирали воздействия дисперсионной среды (% от диаметра трубопровода): 10.

Случайное изменение перемещения частиц вследствие воздействия дисперсионной среды (% от  $dx$ ,  $dy$ ,  $dz$ ): 15.

Диапазон условных масс частиц (усл. ед.): 15-25.

Расположение препятствия по оси  $x$  от начала трубопровода (% от длины трубопровода): 70.

Форма препятствия: параллелепипед по центру.

Толщина препятствия (усл. ед.): 500.

Площадь перекрытия сечения трубопровода препятствием (% от площади сечения трубопровода): 30.

### **Вариант 3**

Длина трубопровода (усл. ед.): 200.

Диаметр трубопровода (усл. ед.): 10.

Количество частиц: 250.

Собственная скорость частиц (усл. ед./этап): 5.

Диаметр спирали воздействия дисперсионной среды (% от диаметра трубопровода): 50.

Случайное изменение перемещения частиц вследствие воздействия дисперсионной среды (% от  $dx$ ,  $dy$ ,  $dz$ ): 5.

Диапазон условных масс частиц (усл. ед.): 5-10.

Расположение препятствия по оси  $x$  от начала трубопровода (% от длины трубопровода): 50.

Форма препятствия: параллелепипед по центру.

Толщина препятствия (усл. ед.): 50.

Площадь перекрытия сечения трубопровода препятствием (% от площади сечения трубопровода): 70.

### **Вариант 4**

Длина трубопровода (усл. ед.): 2000.

Диаметр трубопровода (усл. ед.): 300.

Количество частиц: 150.

Собственная скорость частиц (усл. ед./этап): 20.

Диаметр спирали воздействия дисперсионной среды (% от диаметра трубопровода): 75.

Случайное изменение перемещения частиц вследствие воздействия дисперсионной среды (% от  $dx$ ,  $dy$ ,  $dz$ ): 15.

Диапазон условных масс частиц (усл. ед.): 25-30.

Расположение препятствия по оси  $x$  от начала трубопровода (% от длины трубопровода): 30.

Форма препятствия: параллелепипед смещённый к краю.

Толщина препятствия (усл. ед.): 700.

Площадь перекрытия сечения трубопровода препятствием (% от площади сечения трубопровода): 20.

### **Вариант 5**

Длина трубопровода (усл. ед.): 1400.

Диаметр трубопровода (усл. ед.): 120.

Количество частиц: 400.

Собственная скорость частиц (усл. ед./этап): 30.

Диаметр спирали воздействия дисперсионной среды (% от диаметра трубопровода): 40.

Случайное изменение перемещения частиц вследствие воздействия дисперсионной среды (% от  $dx$ ,  $dy$ ,  $dz$ ): 10.

Диапазон условных масс частиц (усл. ед.): 5-15.

Расположение препятствия по оси  $x$  от начала трубопровода (% от длины трубопровода): 40.

Форма препятствия: цилиндр по центру.

Толщина препятствия (усл. ед.): 200.

Площадь перекрытия сечения трубопровода препятствием (% от площади сечения трубопровода): 80.

### **Вариант 6**

Длина трубопровода (усл. ед.): 800.

Диаметр трубопровода (усл. ед.): 60.

Количество частиц: 200.

Собственная скорость частиц (усл. ед./этап): 5.

Диаметр спирали воздействия дисперсионной среды (% от диаметра трубопровода): 80.

Случайное изменение перемещения частиц вследствие воздействия дисперсионной среды (% от  $dx$ ,  $dy$ ,  $dz$ ): 20.

Диапазон условных масс частиц (усл. ед.): 10-15.

Расположение препятствия по оси  $x$  от начала трубопровода (% от длины трубопровода): 70.

Форма препятствия: цилиндр, смещённый к краю.

Толщина препятствия (усл. ед.): 100.

Площадь перекрытия сечения трубопровода препятствием (% от площади сечения трубопровода): 60.

### **Вариант 7**

Длина трубопровода (усл. ед.): 600.

Диаметр трубопровода (усл. ед.): 80.

Количество частиц: 300.

Собственная скорость частиц (усл. ед./этап): 30.

Диаметр спирали воздействия дисперсионной среды (% от диаметра трубопровода): 40.

Случайное изменение перемещения частиц вследствие воздействия дисперсионной среды (% от  $dx$ ,  $dy$ ,  $dz$ ): 10.

Диапазон условных масс частиц (усл. ед.): 20-35.

Расположение препятствия по оси  $x$  от начала трубопровода (% от длины трубопровода): 10.

Форма препятствия: кольцо.

Толщина препятствия (усл. ед.): 50.

Площадь перекрытия сечения трубопровода препятствием (% от площади сечения трубопровода): 40.

### **Вариант 8**

Длина трубопровода (усл. ед.): 900.

Диаметр трубопровода (усл. ед.): 100.

Количество частиц: 150.

Собственная скорость частиц (усл. ед./этап): 40.

Диаметр спирали воздействия дисперсионной среды (% от диаметра трубопровода): 10.

Случайное изменение перемещения частиц вследствие воздействия дисперсионной среды (% от  $dx$ ,  $dy$ ,  $dz$ ): 15.

Диапазон условных масс частиц (усл. ед.): 10-20.

Расположение препятствия по оси  $x$  от начала трубопровода (% от длины трубопровода): 60.

Форма препятствия: параллелепипед по центру.

Толщина препятствия (усл. ед.): 300.

Площадь перекрытия сечения трубопровода препятствием (% от площади сечения трубопровода): 60.

### **Вариант 9**

Длина трубопровода (усл. ед.): 700.

Диаметр трубопровода (усл. ед.): 150.

Количество частиц: 500.

Собственная скорость частиц (усл. ед./этап): 30.

Диаметр спирали воздействия дисперсионной среды (% от диаметра трубопровода): 80.

Случайное изменение перемещения частиц вследствие воздействия дисперсионной среды (% от  $dx$ ,  $dy$ ,  $dz$ ): 20.

Диапазон условных масс частиц (усл. ед.): 15-25.

Расположение препятствия по оси  $x$  от начала трубопровода (% от длины трубопровода): 40.

Форма препятствия: параллелепипед по центру.

Толщина препятствия (усл. ед.): 200.

Площадь перекрытия сечения трубопровода препятствием (% от площади сечения трубопровода): 20.

### **Вариант 10**

Длина трубопровода (усл. ед.): 1100.

Диаметр трубопровода (усл. ед.): 110.

Количество частиц: 600.

Собственная скорость частиц (усл. ед./этап): 50.

Диаметр спирали воздействия дисперсионной среды (% от диаметра трубопровода): 40.

Случайное изменение перемещения частиц вследствие воздействия дисперсионной среды (% от  $dx$ ,  $dy$ ,  $dz$ ): 30.

Диапазон условных масс частиц (усл. ед.): 5-10.

Расположение препятствия по оси  $x$  от начала трубопровода (% от длины трубопровода): 50.

Форма препятствия: параллелепипед по центру.

Толщина препятствия (усл. ед.): 300.

Площадь перекрытия сечения трубопровода препятствием (% от площади сечения трубопровода): 70.

## ЗАКЛЮЧЕНИЕ

В теоретической части методических указаний по выполнению курсовой работы приведена последовательность действий, сопровождаемая примерами кода, которая позволяет в начальном приближении моделировать дисперсную систему в замкнутом кубическом объёме.

Для успешного выполнения курсовой работы рекомендуется тщательно ознакомиться с документацией к библиотеке `matplotlib`, что позволит визуализировать достаточно сложные трёхмерные объёмы.

Визуальное оформление, а также способ реализации геометрии трубопровода с препятствием остаются на усмотрение студента. Выполнение кода, написанного в ходе работы, должно приводить к отрисовке трубопровода и траекторий частиц в заданном объёме. Код должен выполняться без ошибок.

Для моделирования дисперсной системы рекомендуется придерживаться объектно-ориентированного подхода, однако расчёт взаимодействия частиц может включать элементы функционального программирования.

Оформление отчёта по курсовой работе должно соответствовать действующим государственным стандартам. Содержание пояснительной записки к курсовой работе должно в полной мере описывать структуру разработанной программы, алгоритм её работы в виде блок-схемы с пояснениями, а также поэтапное описание процесса разработки программы с выделением возникших проблем и их решений.

Иллюстрации в пояснительной записке должны быть напрямую связаны с её содержанием. Код необходимо оформлять в виде текстовых листингов (как это показано в методических указаниях). При написании пояснительной записки не следует заимствовать фрагменты из учебников и интернета. Достаточно в тексте оформить ссылки на них.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Marchisio D.L., Fox R.O. Multiphase reacting flows: modelling and simulation. CISM, Udine. 2007. 228 с.
2. Закожурников Ю.А. Транспортировка нефти, нефтепродуктов и газа. – 2010. 432 с.
3. Python documentation [Электронный ресурс]. – URL: <https://docs.python.org/3/> (дата обращения 10.02.2020).
4. Matplotlib documentation [Электронный ресурс]. – URL: <https://matplotlib.org/contents.html> (дата обращения 10.02.2020).
5. Библиотека Matplotlib [Электронный ресурс]. – URL: <https://devpractice.ru/files/books/python/Matplotlib.book.pdf> (дата обращения 10.02.2020).
6. ГОСТ 7.32-2017 Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления. М.: Стандартинформ, 2018. 33 с.
7. ГОСТ 2.105-95. Единая система конструкторской документации. Общие требования к текстовым документам. М.: Стандартинформ, 2007. 32 с.
8. ГОСТ Р 7.0.5-2008. Библиографическая ссылка. Общие требования и правила составления. М.: Стандартинформ, 2008. 23 с.

**ИНФОРМАТИКА**  
**МОДЕЛИРОВАНИЕ СВОЙСТВ НЕФТИ**  
**КАК ДИСПЕРСНОЙ СИСТЕМЫ**

*Методические указания по выполнению курсовой работы  
для студентов специальности 21.03.01*

Сост.: *С.Б. Крыльцов, М.А. Коробицына*

Печатается с оригинал-макета, подготовленного кафедрой  
информатики и компьютерных технологий

Ответственный за выпуск *С.Б. Крыльцов.*  
Лицензия ИД № 06517 от 09.01.2002

Подписано к печати 26.10.2020. Формат 60×84/16.  
Усл. печ. л. 1,5. Усл.кр.-отт. 1,5. Уч.-изд.л. 1,2. Тираж 75 экз. Заказ 750.

Санкт-Петербургский горный университет  
РИЦ Санкт-Петербургского горного университета  
Адрес университета и РИЦ: 199106 Санкт-Петербург, 21-я линия, 2