

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра математической
кибернетики и компьютерных наук

**РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ ДЛЯ РАСЧЕТА И ВИЗУАЛИЗАЦИИ
АМПЛИТУДНО-ЧАСТОТНЫХ И ФАЗОЧАСТОТНЫХ
ХАРАКТЕРИСТИК МЕХАНИЧЕСКОЙ СИСТЕМЫ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студентки 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные
технологии
факультета КНиИТ
Быковой Анастасии Дмитриевной

Научный руководитель
доцент, к. ф.-м. н.


10.06.2017

Ю. Н. Кондратова

Заведующий кафедрой
к. ф.-м. н.


10.06.17

С. В. Миронов

Саратов 2017

ВВЕДЕНИЕ

В большинстве современных конструкций, используемых в различных отраслях техники, в том числе и на транспорте, справедлива модель, которая состоит из двух цилиндрических оболочек, вложенных друг в друга, между которыми расположена жидкость. Примерами использования такой модели можно считать жидкостные реактивные двигатели (ЖРД), телескопические шасси, плунжерную пару с полым плунжером и т.д. В таких моделях жидкость между оболочками может служить как для демпфирования собственных колебаний оболочек, так и для охлаждения этих оболочек. Такие системы «упругая оболочка — охлаждающая жидкость — упругая оболочка» выступают как единое целое. В них возможен резонанс, при котором прогибы внутренней и внешней оболочек будут максимальны, а в слое охлаждающей жидкости возникает кавитация на частотах соответствующих резонансным. Выявление резонансных частот для систем с заданными параметрами является одной из важнейших задач современной науки и техники. В связи с этим особенно актуальным становится возможность корректировки параметров системы «упругая оболочка — охлаждающая жидкость — упругая оболочка» на этапе проектирования с помощью программного комплекса.

Цель бакалаврской работы. Целью работы является создание приложения, с помощью которого появляется возможность компьютерного моделирования процесса взаимодействия упругих соосных цилиндрических оболочек с вязкой несжимаемой жидкостью, находящейся между ними и получение амплитудно-частотных и фазочастотных характеристик при различных входных параметрах. Математическая модель для создания программного комплекса была сформулирована в работах [1,2].

При наличии созданной в бакалаврской работе программы можно еще на этапе разработки системы, содержащей вышеописанную конструкцию, рассчитать резонансные частоты. Путем корректировки параметров системы: материалов, размеров можно сдвинуть резонансные частоты в ту или иную сторону. Построение графика по амплитудно-частотным и фазочастотным характеристикам позволяет наглядно проследить изменения, происходящие в системе.

Для достижения поставленной цели в бакалаврской работе необходимо решить следующие задачи:

- рассмотреть математическую модель системы, состоящей из двух вло-

женных соосных цилиндрических оболочек, соединенных вязкой несжимаемой жидкостью;

- провести анализ технических средств разработки;
- разработать программу для компьютерного моделирования процесса взаимодействия упругих соосных цилиндрических оболочек с вязкой несжимаемой жидкостью, находящейся между ними;
- разработать программный комплекс визуализации проведенных расчетов;
- провести анализ полученных результатов при различных входных параметрах.

Основные результаты:

- рассмотрены возможности языка Python;
- разработан программный комплекс визуализации расчетов;
- проведено тестирование программного комплекса на входных параметрах при последовательном и параллельном запуске вычислений.

Структура бакалаврской работы состоит из 3 глав, введения, заключения, списка использованной литературы, рисунков, страниц, 1 приложения.

В первой главе поставлена задача дипломного проекта и описана математическая модель гидроупругости тонкостенных конструкций. Во второй главе на основе собранной информации проведен анализ технических средств разработки программного комплекса. В третьей главе представлен разработанный программный комплекс компьютерного моделирования для системы «упругая оболочка - охлаждающая жидкость - упругая оболочка» и проведено тестирование приложения при различных входных параметрах.

1 Основное содержание работы

Рассматривается механическая система, состоящая из двух соосных упругих цилиндрических оболочек конечной длины с жестким защемлением по торцам, сдавливающих слой вязкой несжимаемой жидкости. Модель представлена на рисунке 1.

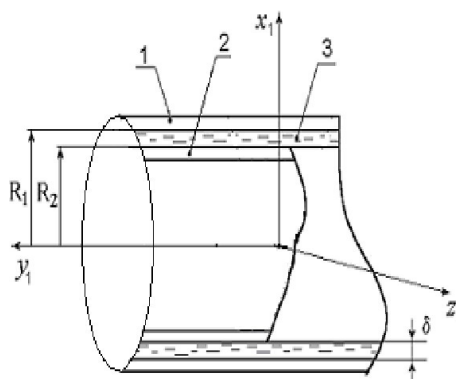


Рисунок 1 – Физическая модель жидкостного реактивного двигателя.

Предполагается, что внешняя оболочка 1 — упругая замкнутая цилиндрическая оболочка с внутренним радиусом R_1 и с жестким защемлением по торцам. Внутренняя оболочка 2, с внешним радиусом R_2 — также упругая замкнутая цилиндрическая оболочка, жестко защемленная по торцам. Зазор между стенками оболочек 2 и 1 полностью заполнен жидкостью 3. Наружная поверхность внешней оболочки и поверхность внутренней оболочки образуют цилиндр в цилиндре длиной $2l$. Радиальный зазор цилиндрической щели $\delta = R_1 - R_2 \ll R_2$. Торцевые зазоры рабочей камеры, с абсолютно жесткими стенками, имеют протяженность a , значительно большую радиального зазора ($a \gg \delta$). На систему действуют гармонически меняющееся по времени на торцах давление и переносная сила инерции. Перемещение внутренней оболочки относительно внешней на защемленных торцах отсутствует. Механическая система считается термостабилизированной.

Математическая модель рассматриваемой механической системы представляет собой связанную систему уравнений, включающую нелинейные уравнения в частных производных Навье-Стокса и уравнение неразрывности, уравнения в частных производных для описания динамики внутренней и внешней упругих цилиндрических оболочек, полученные исходя из гипотез Кирхгофа-Лява, и соответствующие граничные условия.

Решение этих задач представлено в виде гармонических функций по времени и бесконечных рядов по координате с использованием процедуры Бубнова-Галеркина. На основе построенной математической модели найдены амплитудно-частотные и фазочастотные характеристики для внутренней и внешней оболочек в условиях воздействия внешнего источника вибрации применительно к силовым цилиндрам и двигателям внутреннего сгорания.

На основании формул, взятых из [1], в бакалаврской работе проводилось компьютерное моделирование процессов взаимодействия упругих соосных цилиндрических оболочек с вязкой несжимаемой жидкостью, находящейся между ними. Для создания оконного приложения в бакалаврской работе используется математическая модель, результатом которой являются амплитудно-частотные и фазочастотные характеристики. Получившийся программный продукт должен удовлетворять следующим требованиям:

- при вводе любых допустимых параметров вычислять амплитудно-частотные и фазочастотные характеристики;
- иметь возможность выбора файла со значениями параметров;
- построение графика и сохранения его в память компьютера;
- выполнение вычислений с параллельной обработкой данных.

В данной бакалаврской работе использовались следующие средства:

1. Python — высокоуровневый язык программирования общего назначения: от простых скриптов на веб-сервере до программ для управления сложными процессами на производстве. Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных;
2. Matplotlib — это библиотека двумерной графики для языка программирования Python, с помощью которой можно создавать высококачественные рисунки различных форматов. Matplotlib является гибким, легко конфигурируемым пакетом, который вместе с NumPy, SciPy и Python предоставляет возможности, подобные MATLAB;
3. PyQt4 — это набор инструментов для разработки GUI приложений. Он

представляет из себя смесь языка программирования Python и библиотеки Qt;

4. Multiprocessing — модуль Python для реализации параллельных процессов, использующий особые приёмы для клонирования процессов. Модуль реализует базовые примитивы синхронизации: мьютексы, семафоры, условные переменные.

Под пользовательским интерфейсом (ПИ) программы будет пониматься совокупность элементов, позволяющих пользователю программы управлять ее работой и получать требуемые результаты. Фактически, пользовательский интерфейс — это канал, по которому осуществляется взаимодействие пользователя и программы. Главными требованиями при реализации любого программного продукта является удобство и функциональность. В данной задаче необходима реализация чтения из файла, возможность изменения данных, а также их проверка, вывод результатов на экран, построение графиков и сохранение в файл. Таким образом, необходимо последовательно решать поставленные цели.

Для удобства пользователя необходимо создание двух вкладок: в первой будет пакетная обработка, а во второй будет работа со значениями и построение графиков.

Необходимые минимальные возможности приложения: добавление названия, изменение размеров экрана, добавление элементов.

Класс `QTabWidget` предоставляет набор виджетов со вкладками. Виджет с вкладками — это панель вкладок и «область страниц», которая используется для отображения страниц, связанных с каждой из вкладок. Было создано две вкладки с названиями «package» и «value». Результат создания простейшего приложения представлен на рисунке 2.

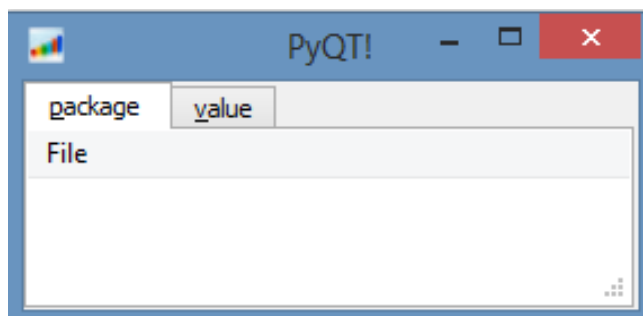


Рисунок 2 – Приложение с вкладками и меню.

Следующим этапом создания приложения является расположение полей,

надписей и кнопок внутри вкладок. PyQt предоставляет простейшие функции для создания и расположения кнопок, надписей, полей для данных. Для создания надписи необходимо использовать возможности пакета QtGui, в котором содержатся все графические компоненты. В указанном пакете содержится метод QLabel, который для переданной строки возвращает сущность «надпись». Надпись не будет выводиться на экране, пока ей не указать координаты расположения. Для этого необходимо вызвать метод move() для изменения положения виджета. Виджет располагается согласно координатам X и Y. Начало системы координат находится в левом верхнем углу окна. Координата X растёт справа налево, а Y сверху вниз.

```
1 QtGui.QLabel('NumberVariant', self).move(5, 50)
```

Для создания текстовых полей используется такой же принцип, что и для создания надписей. Отличия будут заключаться лишь в функциях, которые необходимо вызвать для получения желаемого результата. Также важным отличием создания текстового поля от создания надписи является то, что текстовое поле содержит какие-то данные, строку, которую внутри приложения скорее всего необходимо получить. Таким образом, внутри приложения необходимо связать сущность «надпись» с переменной, в которой будут храниться необходимые методы для получения данных или для установки своих значений в текстовое поле.

```
1 self.NumberVariant_ = QtGui.QLineEdit(self)
2 self.NumberVariant_.setGeometry(QtCore.QRect(115, 50, 60, 20))
```

PyQt предоставляет возможность отслеживать действия пользователя, например при клике на кнопку. Необходимо обратить внимание на создание кнопки btn. На кнопке будет изображаться надпись «choose file». Из названия следует, что при нажатии на кнопку надо будет выбрать файл с входными параметрами. Чтобы определить действие при нажатии на кнопку, вызывается метод clicked.connect класса QPushButton. Само действие определяется в методе showDialog класса Window.

```
1 btn = QtGui.QPushButton("choose file", self)
2 btn.clicked.connect(self.showDialog)
```

Следующим этапом на пути к созданию пользовательского приложения является создание возможности выбора файла.

```

1 def showDialog(self):
2     filename = QtGui.QFileDialog.getOpenFileName(self, 'Open file', '/home')
3     file=open(filename)
4     data = file.read()

```

Вычисление амплитудно-частотных и фазочастотных характеристик начинается не сразу, для этого выделена специальная кнопка «calculate» с вызовом метода `CalculateDataInValueTab`. Преимуществом этого является то, что пользователь может проверить корректность данных перед выполнением подсчета характеристик, а также изменить их на необходимое значение.

Метод `CalculateDataInValueTab` выполняет подсчет характеристик, а также проверяет данные на минимальную корректность: все значения должны быть положительными.

В начале составления программы был импортирован файл под названием `achfchModule`, именно этот файл содержит функцию `go`, которая выполняет подсчет амплитудно-частотных и фазочастотных характеристик. Данная функция возвращает 5 массивов: массив состоящий из проверяемых частот, а также массивы состоящие из значений фазочастотных характеристик внутренней и внешней оболочек, амплитудно-частотных характеристик внутренней и внешней оболочек. Выполнение данного метода завершится выводом сообщения в статусное поле.

Рисунок `Figure` является основой каждого изображения, которые создаётся в `Matplotlib`. Обычно после создания экземпляра рисунка (точнее — экземпляра класса `Matplotlib.figure.Figure`) о нём можно забыть. Тем не менее, именно на основе экземпляра `Figure` вызываются экземпляры `Axes`, которые несут основную нагрузку при создании научной графики.

Область рисования `Axes` — это центральный объект при создании научных графиков. Это тот объект, та область на которую наносятся координатные оси, линии графиков и столбцы диаграмм, легенды к ним и так далее. Взамен активного использования `Pyplot` предлагается использовать более объектно-ориентированный подход для вызова графических команд и изменениях их свойств. Структурно область рисования `Axes` необходима как элемент более низкого иерархического уровня нежели `Figure` (весь рисунок), на котором непосредственно будут располагаться различные графические элементы. `Axes` похож на `Frame` в веб-дизайне, то есть это область, на которой можно раз-

мещать графические примитивы. При этом в Matplotlib непосредственно на Figure нельзя рисовать (изменение цвета фона не считается за полноценное рисование) и соответственно нужен хотя бы один экземпляр Axes.

Для пользователя важно различать какая линия к какой зависимости относится. Для таких ситуаций в Matplotlib предусмотрена легенда графика. Создание легенды подразумевает два логических этапа:

- Формирование подписей к условным обозначениям;
- Отображение легенды на рисунке.

В конце произведена инициализация FigureCanvas. Построение графика фазочастотных характеристик внешней оболочки представлено на рисунке 3.

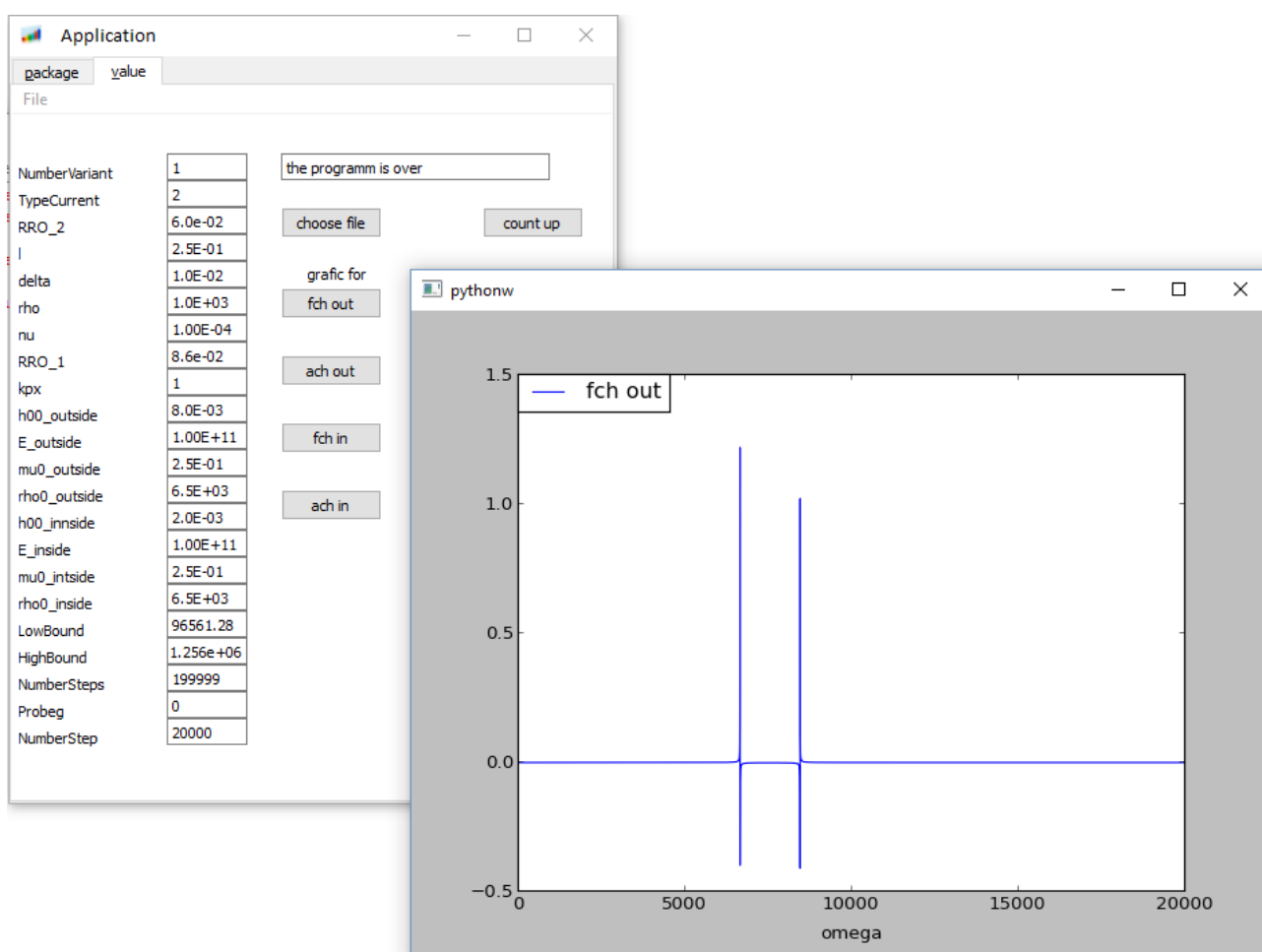


Рисунок 3 – График фазочастотных характеристик внешней оболочки.

Необходимо создать вкладку «package». Данная вкладка будет наделена меньшим функционалом, чем вкладка «value». Работа с данной вкладкой будет интуитивно понятна пользователю.

На вкладке будут располагаться: поля для входных и выходных данных,

поле для ввода количества шагов, поле статуса и кнопка запуска подсчета амплитудно-частотных и фазочастотных характеристик. По аналогии с созданием первой вкладки, необходимо создать метод, в котором и будет осуществляться отрисовка окна.

```
1 self.output_ = QtGui.QTextEdit(self)
2 self.output_.setGeometry(QtCore.QRect(90, 200, 300, 150))
```

В данном случае, единственным нововведением является создание поля QTextEdit, который отличается возможностью представления многострочных текстов. При нажатии на кнопку с надписью result запускается метод CalculateDataInPackageTab. Аналогично первой вкладке запускается проверка на корректность данных: если данные прошли проверку, то запускается процесс подсчета амплитудно-частотных характеристик. В самом файле achfchModule результаты записываются в файл, поэтому в данном случае просто считаются данные из этого файла, и с помощью метода setText помещаются в окно для выходных данных. Пример работы вкладки «package» представлен на рисунке 4.

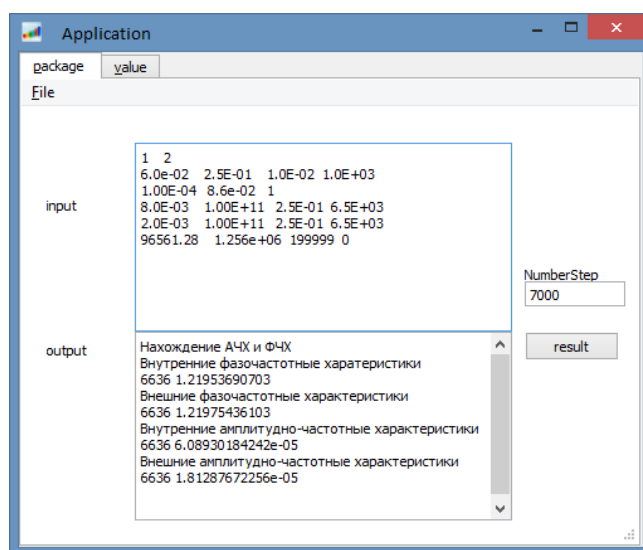


Рисунок 4 – Работа вкладки «package».

Для распараллеливания используется модуль Multiprocessing. Map — это удобная маленькая функция, а главное, простая для распараллеливания Python кода. Map заимствована из функциональных языков. Это функция, которая применяет другую функцию к последовательности, например:

```
1 allResult = pool.map(startCalculate, allData)
```

Map совершает итерацию по последовательности, применяет функцию и сохраняет результаты в список.

В `allData` будет составлять массив, где каждый элемент это набор элементов характеризующих параметры для вычисления амплитудно-частотных и фазочастотных характеристик. `allResult` будет представлять собой массив из результатов, где каждый элемент это набор `omega`, `fchin`, `fchout`, `achin`, `achout` (частота, фазочастотные, амплитудно-частотных характеристики для внутренней и внешней оболочки соответственно). В `startCalculate` происходит обращение к функции `go`, производящей необходимые вычисления.

Для доступа к map-параллельной функции, сперва нужно импортировать модули в которых она содержится и создать пул.

Объекты из пула принимают несколько параметров, но упоминания стоит только один: `processes`. Этот параметр устанавливает количество воркеров в пуле. Если оставить это поле пустым, то по умолчанию оно будет равно количеству ядер в процессоре.

```
1 from multiprocessing import Pool as ThreadPool
2 pool = ThreadPool(2)
```

Если используется многопроцессовый пул для ядро-раздельных задач, то больше ядер означает большую скорость. Если запустить слишком много потоков, время, затраченное на переключения между ними будет больше, чем на полезную работу, так что в этом случае стоит изменять параметры до тех пор, пока не найдется оптимальный вариант для задачи.

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы была рассмотрена математическая модель «упругая оболочка — охлаждающая жидкость — упругая оболочка». Были изучены основные преимущества и недостатки различных языков программирования, предназначенных для компьютерного моделирования процесса взаимодействия упругих соосных цилиндрических оболочек. Для создания приложения, моделирующего поведение рассматриваемой системы, была выбрана связка Python, Matplotlib, PyQt, Multiprocessing.

В работе было создано приложение с простым и удобным интерфейсом для подсчета амплитудно-частотных и фазочастотных характеристик. Созданное приложение позволяет анализировать результаты сразу после их обработки. Реализованное в работе распараллеливание вычислений позволяет экономнее расходовать ресурсы системы, обоснование чего приведено выше. Разработанное приложение было протестировано на нескольких примерах, описанных в работе. Графики, построенные в созданном программном комплексе, показывают смещение резонансных частот при изменении типоразмеров. В результате, с помощью данного приложения можно подобрать типоразмеры системы таким образом, чтобы резонансные частоты сместились из области допустимых частот.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Кондратова, Ю. Н. Моделирование процессов взаимодействия упругих соосных цилиндрических оболочек с вязкой несжимаемой жидкостью, находящейся между ними :дис. ... канд. физ.-мат. наук : 01.02.04 / Ю. Н. Кондратова. — Саратов, 2011.
- 2 Кондратов, Д. В. Математическое моделирование процессов упругогидродинамики в машино- и приборостроении :дис. ... доктор физ.-мат. наук : 05.13.18 / Д. В. Кондратов. — Саратов, 2009.
- 3 Прохоренок, Н. А. Python 3 и PyQt. Разработка приложений / Н. А. Прохоренок. — Санкт-Петербург: БХВ-Петербург, 2016.
- 4 Tosi, S. Matplotlib for Python Developers / S. Tosi. — Packt Publishing Ltd, 2014.
- 5 Matplotlib documentation [Электронный ресурс]. — URL: <http://matplotlib.org/> (Дата обращения 13.05.2017). Загл. с экр. Яз. англ.
- 6 PyQt [Электронный ресурс]. — URL: <https://habrahabr.ru/post/31684/> (Дата обращения 13.05.2017). Загл. с экр. Яз. англ.
- 7 Чан, У. Д. Python: создание приложений. Библиотека профессионала / У. Д. Чан. — Вильямс, 2016.
- 8 Лутц, М. Программирование на Python (Том 1) / М. Лутц. — Символ-Плюс, 2013.
- 9 Bird, S. Natural Language Processing with Python / S. Bird. — O'Reilly Media, 2012.
- 10 McGregor, D. M. Mastering matplotlib / D. M. McGregor. — Packt Publishing, 2015.
- 11 Figure api [Электронный ресурс]. — URL: https://matplotlib.org/api/figure_api.html (Дата обращения 25.05.2017). Загл. с экр. Яз. англ.

10.06.2017 