

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по проектной работе №1
по дисциплине «Компьютерная Математика»
Тема: Повороты графа в пространстве.

Студенты гр. 0383

Сергеев Д.В.,
Бояркин Н.А.

Преподаватель

Коптелов Я.Ю.

Санкт-Петербург

2022

Оглавление.	2
Задание.	3
Выбранный метод выполнения задачи.	3
Построение графа.	4
Выбор «правильных» рёбер.	4
Поворот графа.	4
Решение обратной задачи.	6
Тестирование	6
Выводы.	10
Список использованных источников.	11

Задание.

Пусть дана информация о структуре молекулы в формате SDF из базы данных PubChem, например, молекула аспирина.

Программа должна распарсить SDF (только атомы с координатами и связи с кратностями, игнорируя заряды и прочую дополнительную информацию), построить в памяти граф молекулы (вершины = атомы, рёбра = связи) и определить рёбра, которые одновременно:

- одинарные (т. е. соответствуют одинарным, а не двойным и не тройным связям)
- не принадлежат циклам
- не являются "висячими", т. е. продолжаются с обоих концов

В вышеупомянутой молекуле аспирина таких рёбер четыре (если смотреть на двумерную картинку, в которой не показаны водороды); в трёхмерной картинке с водородами таких рёбер можно насчитать пять.

Далее, программа должна повернуть молекулу в трёхмерном пространстве вокруг каждого из найденных рёбер на случайный угол так, чтобы вращалась не вся молекула, а лишь "половина", лежащая по одну сторону от ребра. Координаты атомов при этом, конечно, изменятся, изменится форма молекулы.

Далее, программа должна забыть случайные углы, проанализировать получившиеся координаты атомов и найти такие углы вращения вокруг найденных связей, чтобы атомы встали на исходные места.

Выбранный метод выполнения задачи.

Существуют несколько возможных способов решения поставленной задачи. Один из них: сгенерировать множество случайных наборов углов и из каждого из них градиентным спуском приблизить атомы к исходным положениям.

После распарсировки файла и получения графа, запомним исходные координаты. Далее выберем вершину с минимальным индексом и произведём

поиск “правильных” ребер, вокруг которых необходимо вращать исходную молекулу.

После этого сгенерируем случайные углы от 0 до 2π и поворачиваем граф вокруг каждого “правильного” ребра (будет являться осью вращения). Значение углов забываем. В итоге мы получаем изменённый граф.

Для решения обратной задачи воспользуемся градиентным спуском.

Сам алгоритм будет описан ниже.

Построение графа.

Для построения графа был использован модуль Chem из библиотеки rdkit. С помощью функции *Chem.MolFromMolFile()* мы распарсили данные из файла. Для получения координат в графе была реализована функция *get_coordinates(molecule)*. Для графического вывода была использована функция *Chem.Draw.IPythonConsole.drawMol3D()*.

Поиск «правильных» рёбер.

Для поиска “правильных” ребёр была реализована функция *find_bonds(molecule)*. От вершины с индексом 0 проходимся по всем ребрам и если они одновременно: одинарные, не принадлежат циклам и не являются "висячими", то добавляем их в список ребер, который впоследствии возвращаем.

Поворот графа

Один из способов описания поворотов в пространстве — матрицы поворота, которые представляют из себя ортогональные матрицы с единичным определителем. При умножении вектора на такую матрицу изменяются координаты, однако длина вектора остается прежней.

Для выполнения задачи необходимо уметь поворачивать точки в графе относительно одного из рёбер. Для этого необходимо рассчитать единичный вектор из вектора ребра и вычислить матрицу поворота, затем необходимо переместить вершины так, чтобы конечная точка ребра лежала в начале

системы координат, после чего необходимо применить к вершинам матрицу поворота. Таким образом находятся новые вершины после поворота на угол θ :

$(x_{new}, y_{new}, z_{new}, 1) = (x_{old}, y_{old}, z_{old}, 1) * T^{-1} * M_{gen} * T$, где T - матрица смещения начала координат, M_{gen} - расширенная до 4x4 матрица поворота M :

$$M(\hat{v}, \theta) = \begin{pmatrix} \cos \theta + (1 - \cos \theta)x^2 & (1 - \cos \theta)xy - (\sin \theta)z & (1 - \cos \theta)xz + (\sin \theta)y \\ (1 - \cos \theta)yx + (\sin \theta)z & \cos \theta + (1 - \cos \theta)y^2 & (1 - \cos \theta)yz - (\sin \theta)x \\ (1 - \cos \theta)zx - (\sin \theta)y & (1 - \cos \theta)zy + (\sin \theta)x & \cos \theta + (1 - \cos \theta)z^2 \end{pmatrix}$$

Где x, y, z – это произвольный единичный вектор, θ – угол поворота.

Для выполнения работы была реализована функция *rotate(molecule, A, B, V, degrees)*, где *molecule* - граф, *A* - какие вершины необходимо вращать, *B* - точка оси вращения, *V* - точка смещения координат, *degrees* - угол поворота.

Эта функция вызывается в *rotate_graph(m, bonds, degrees, is_reverse)*, где *m* - молекула, *bonds* - связи в молекуле, вокруг которых мы поворачиваем вершины, *degrees* - список углов (в градусах) - на какой угол поворачиваем рёбра, *is_reverse* - флаг обхода вершин. С помощью обхода в ширину находятся изменяемые вершины.

Решение обратной задачи.

Обратную задачу мы решаем с помощью градиентного спуска. Он осуществляется следующим образом:

1. Мы генерируем случайные углы

2. Итеративно выполняем:

2.1 Для каждого ребра совершаем обратный поворот.

2.2 От функции ошибки считаем производную по параметру текущий угол.

2.3 Далее обновляем каждый угол с заданным шагом:

$$\theta^{(k+1)} = \theta^{(k)} - h * \frac{df}{d\theta^{(k)}}$$

Сначала обновили последний угол. Обновили координаты, то есть исходные координаты повернули на этот угол. Аналогично делаем для остальных углов.

Алгоритм продолжает работу пока квадратичная разница старых координат и новых больше или равна заданного числа ϵ .

Тестирование программы.

После получения данных из файла, молекула аспирина представлена на рис. 1.

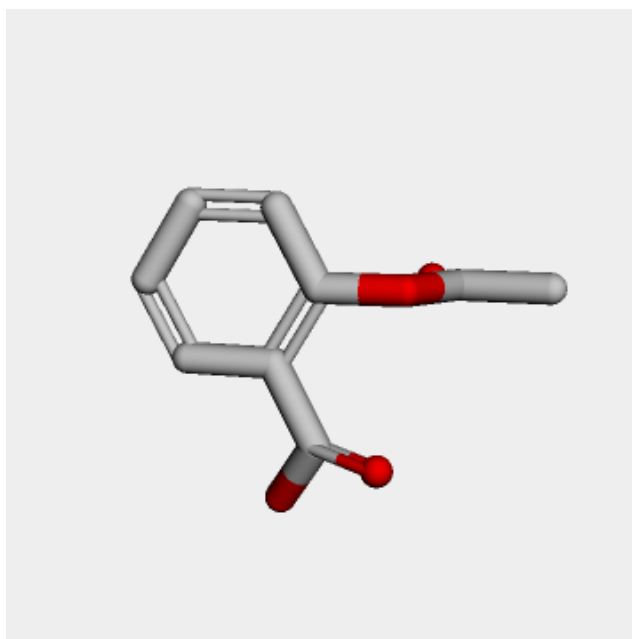


Рисунок 1. Изначальная молекула

Случайно генерируем углы, которые необходимо найти: [247.7, 92.57, 62.44]. Повернем граф на эти углы вокруг “правильных” ребер. Графическое отображение графа можно посмотреть на рис. 2. После поворота забываем сгенерированные углы.

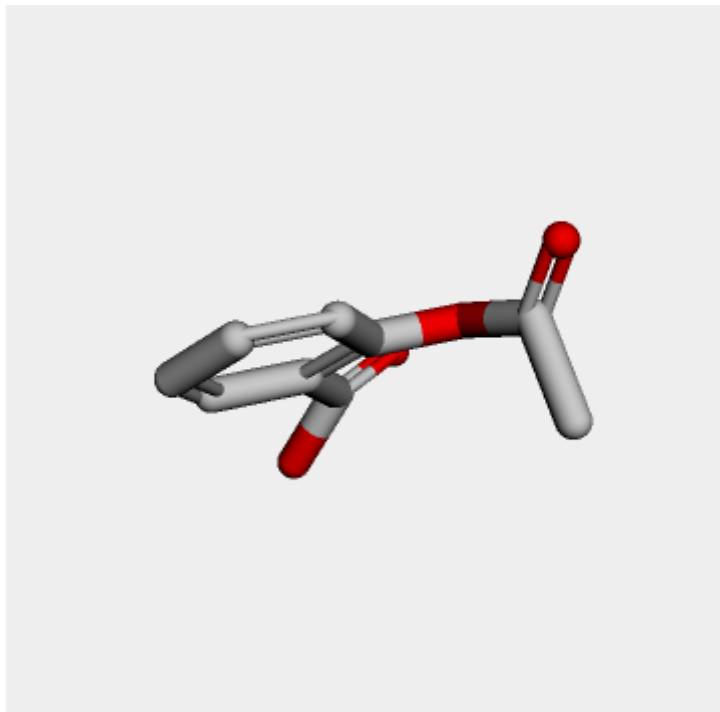


Рисунок 2. Молекула, повернутая на определенные углы вокруг “правильных” ребер

С помощью градиентного спуска найдем приблизительные углы, на которые повернули граф изначально. Найденные углы для нашего примера: [248.42, 88.17, 54.93]. Поворачиваем граф на обратные им углы, вокруг “правильных” ребер. Графическое отображение можно увидеть на рис. 3.

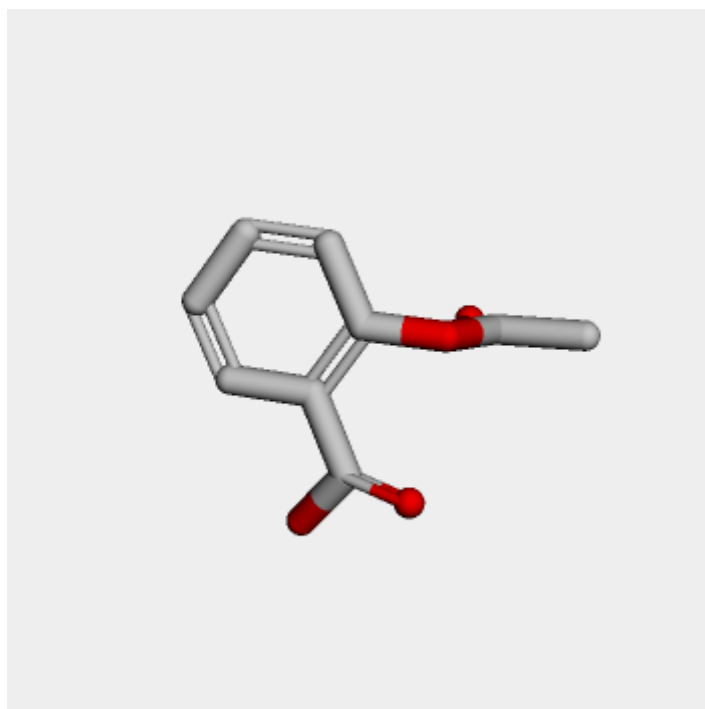


Рисунок 3. Повернутая молекула на найденные углы

Из рисунка 4 видно, что квадратичная ошибка (разница между начальными и найденными координатами в квадрате) стремится к 0 при $\varepsilon = 5$, следовательно алгоритм сходится к правильному решению. Аналогично на рисунке 5 при $\varepsilon = 0.5$.

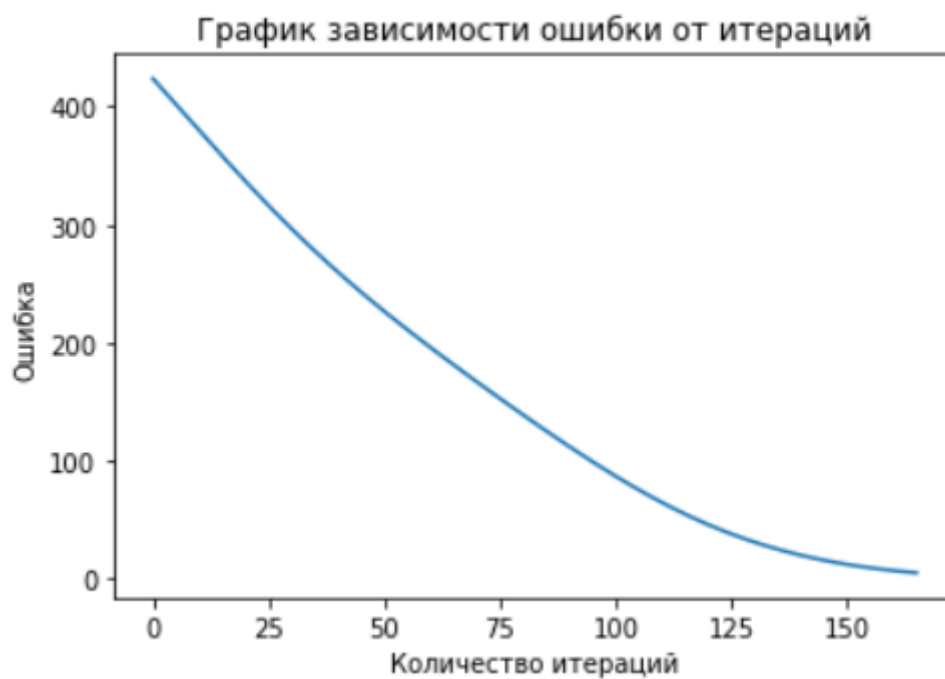


Рисунок 4. График поведения квадратичной ошибки при $\varepsilon = 5$

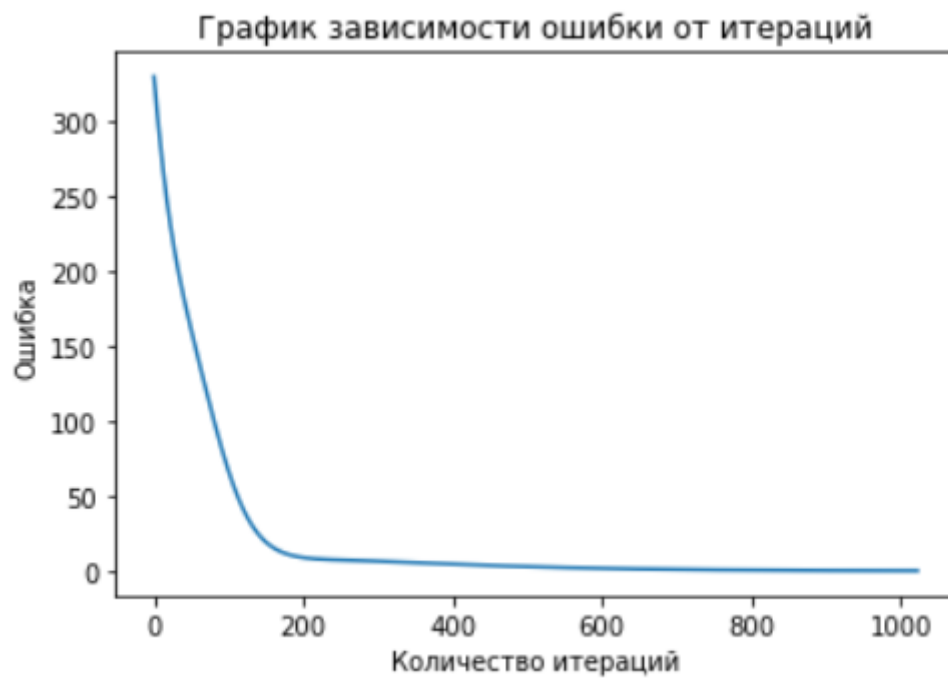


Рисунок 5. График поведения квадратичной ошибки при $\varepsilon = 0.5$

Выводы.

В ходе выполнения данной проектной работы были изучены: предобработка данных, работа со структурой молекулы и метод градиентного спуска. Придумано и реализовано решение обратного поворота молекулы. А также реализованы функции поворота векторов в пространстве с помощью матриц поворота.

Список использованных источников.

1. https://scask.ru/n_book_graph.php?id=18 - ПЕРЕНОС И ПОВОРОТЫ В ТРЕХМЕРНОМ ПРОСТРАНСТВЕ
2. <https://habr.com/ru/post/467185/> - Градиентный спуск по косточкам
3. https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D1%80%D0%B8%D1%86%D0%B0_%D0%BF%D0%BE%D0%B2%D0%BE%D1%80%D0%BE%D1%82%D0%B0 - Матрица поворота
4. https://scask.ru/a_book_mm3d.php?id=60 - ПОВОРОТ ВОКРУГ ПРОИЗВОЛЬНОЙ ОСИ В ПРОСТРАНСТВЕ