

Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования



**Московский государственный технический  
университет имени Н.Э.Баумана  
(МГТУ им. Н.Э.Баумана)**

---

КАФЕДРА  
“ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА”

КУРСОВАЯ РАБОТА  
по курсу «Алгоритмы Компьютерной Графики»  
на тему:

**“Построение молекулярных поверхностей средствами браузера”**

Выполнил:  
студент 3 курса группы ИУ9-51  
Климов С.В.

\_\_\_\_\_  
Руководитель курсовой  
работы:  
Дубанов А. В.

\_\_\_\_\_  
Москва, 2018

## **Оглавление**

<b>Введение.....</b>	<b>3</b>
<b>Постановка задачи.....</b>	<b>3</b>
<b>Определения.....</b>	<b>4</b>
<b>Форматы файлов.....</b>	<b>5</b>
<b>Существующие решения.....</b>	<b>5</b>
<b>Разработка методов построения молекулярной поверхности.....</b>	<b>6</b>
<b>Реализация.....</b>	<b>16</b>
<b>Тестирование.....</b>	<b>21</b>
<b>Результаты.....</b>	<b>23</b>
<b>Литература.....</b>	<b>24</b>

## **Введение**

Молекулярные поверхности являются зоной межмолекулярной коммуникации. Они служат удобным инструментом исследования молекул и проведения многих необходимых расчетов. Наиболее частое применение — визуализация свойств поверхности (заряд, липофильность).

Кроме того, как и в случае географической карты, двумерное изображение поверхности молекулы можно сделать информативным, нанеся на него различные физико-химические характеристики и/или важные особенности структуры. Одно из часто встречающихся применений — расчет и цветовая визуализация электростатического потенциала на молекулярной поверхности, определяющего зарядовые взаимодействия молекул между собой.

Еще один пример — картирование молекулярного «рельефа» на поверхности — расположение «ущелий», «холмов» и «равнин». В частности, эта информация, вместе с локализацией на карте позиций доноров и акцепторов водородной связи, определяет нахождение сайтов «связанной» воды — мест, где молекулы растворителя задерживаются особенно долго.

Такой анализ свойств поверхности используется специалистами в предметных областях (химиками, биотехнологами и другими) для решения ряда практических задач.

## **Постановка задачи**

В задачу данного курсового проекта входит разработка приложения для расчета молекулярной поверхности и дальнейшее её отображение средствами браузера.

## Определения

Стоит отметить пару важных определений, которые будут использоваться при разработке данного приложения.

Ангстрем (Å) — это единица длины, равная  $10^{-10}$  м или 0,1 нанометра .

Естественные науки часто используют ангстрем для выражения размеров атомов, молекул, микроскопических биологических структур и длин химических связей , расположения атомов в кристаллах. Ангстрем является международно признанным, но не является официальной частью Международной системы единиц (СИ). Ближайшая единица СИ — нанометр (  $10^{-9}$  м ).

Ван-дер-Ваальсов радиус атома – это радиус воображаемой сферы, представляющей из себя ближайшее расстояние, на которое может приблизиться какой-либо другой несвязанный атом[1,4].

Скажем пару слов о цветовой схеме моделей Кори — Полинга — Колтуна. Это популярная схема цветового обозначения атомов различных химических элементов в молекулярных моделях. Данная схема названа в честь молекулярной модели *СРК* , созданной Робертом Кори и Лайнусом Полингом и усовершенствованной Уолтером Колтуном. В ней используются белый цвет для водорода, чёрный для углерода, голубой для азота, красный для кислорода, тёмно-жёлтый для серы, пурпурный для фосфора, оттенки от светло- до тёмно-зелёного соответственно для галогенов (F, Cl, Br, I)[2,5].

## Форматы файлов

Наиболее популярными форматами файлов содержащих информацию о строении молекулы являются файлы .mol2 и .pdb. Так как разрабатываемое приложение написано на языке JavaScript, то для удобства в нем используется json-аналог этих файлов, в котором содержится вся необходимая информация о структуре молекулы. При необходимости использования файлов .mol и .pdb можно написать для них парсер, либо использовать уже существующее решение.

## Существующие решения

Существует несколько desktop-приложений предоставляющих возможность построения молекулярной поверхности. К таковым относятся PyMol[7] и Avogadro[8]. Существует также web-аналог - GLmol[6], который позволяет визуализировать поверхность молекулы(рис. 1)

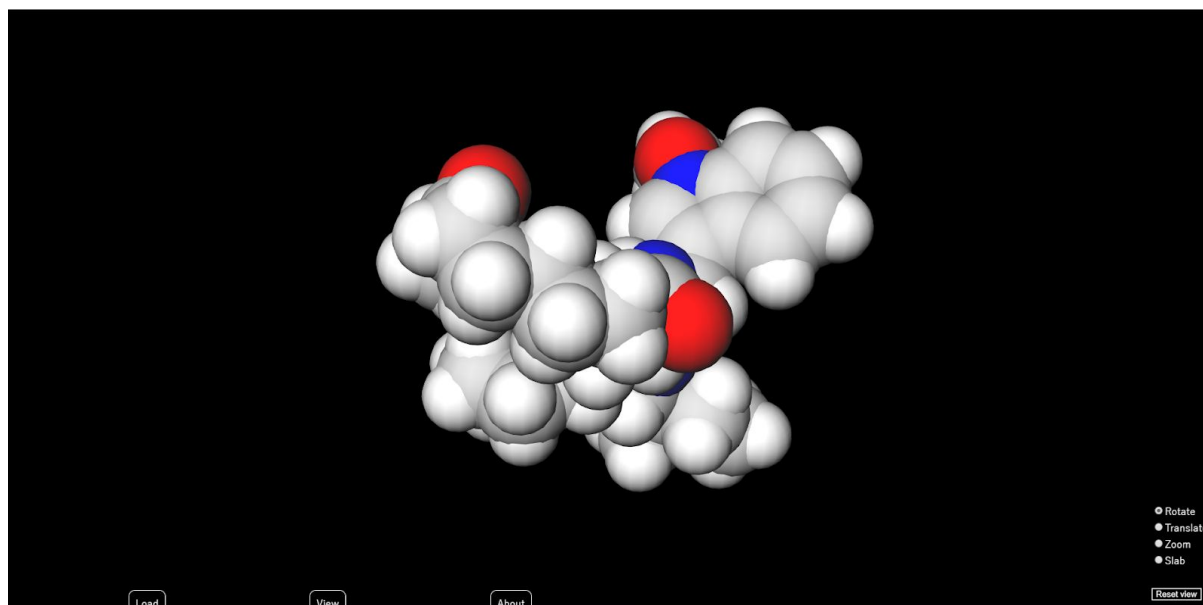


рис. 1 - интерфейс приложения GLmol

## Разработка методов построения молекулярной поверхности

Существуют различные виды молекулярной поверхности[1](рис. 2):

- Ван-дер-Ваальсова поверхность
- Solvent accessible surface
- Solvent excluded surface

а также способы представления пространственной структуры молекулы в виде маленьких шариков соединенных между собой трубами.

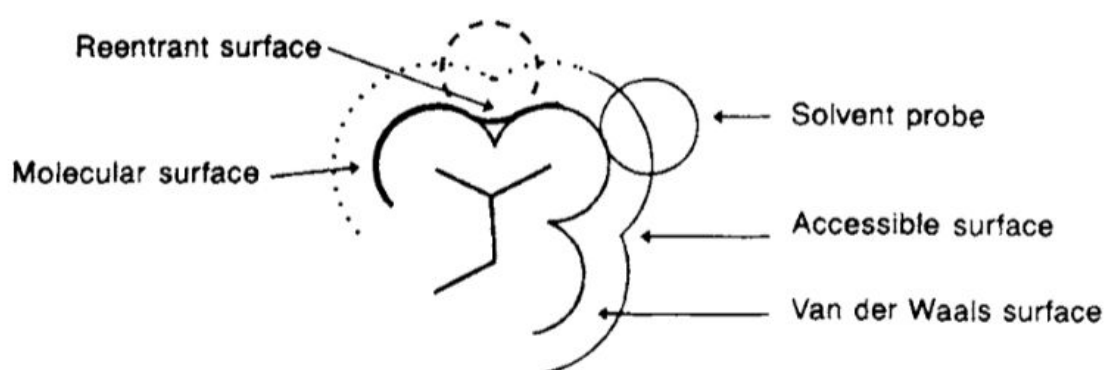


рис. 2 - различные виды поверхностей.

В разработанном приложении строятся эти три вида молекулярной поверхности. Ван-дер-Ваальсова определяется как набор сфер, где каждая сфера соответствует некоторому атому с его радиусом Ван-дер-Ваальса. SAS строится так же как и Ван-дер-Ваальсова, только радиус сфер увеличивается на величину равную радиусу растворителя. Третий вид поверхности является более сложным и требующим дополнительных вычислений. Описывается как поверхность построенная путем прокатки сферы растворителя конкретного радиуса по поверхности молекулы таким образом, что в каждый момент она касается какого либо из атомов(или более одного), но не пересекается ни с одним из остальных. Состоит поверхность из 3-х типов частей[3]:

- выпуклые сферические фрагменты
- тороидальные фрагменты

-вогнутые сферические фрагменты

Каждая из этих частей определяется набором своих параметров и имеет определенный геометрический смысл. Тороидальная часть получается заметанием дуги  $AB$  сферы, которая касается одновременно двух атомов, при вращении вокруг оси, соединяющей эти атомы (рис. 3 а). Вогнутая часть получается из условия одновременного касания трех атомов и определяется сферой растворителя и набором точек касания с этими атомами (рис. 3 б). Радиус сферы растворителя равен  $1,4 \text{ \AA}$ . Выпуклые сферические фрагменты определяются самими атомами, их положением в пространстве и радиусом.

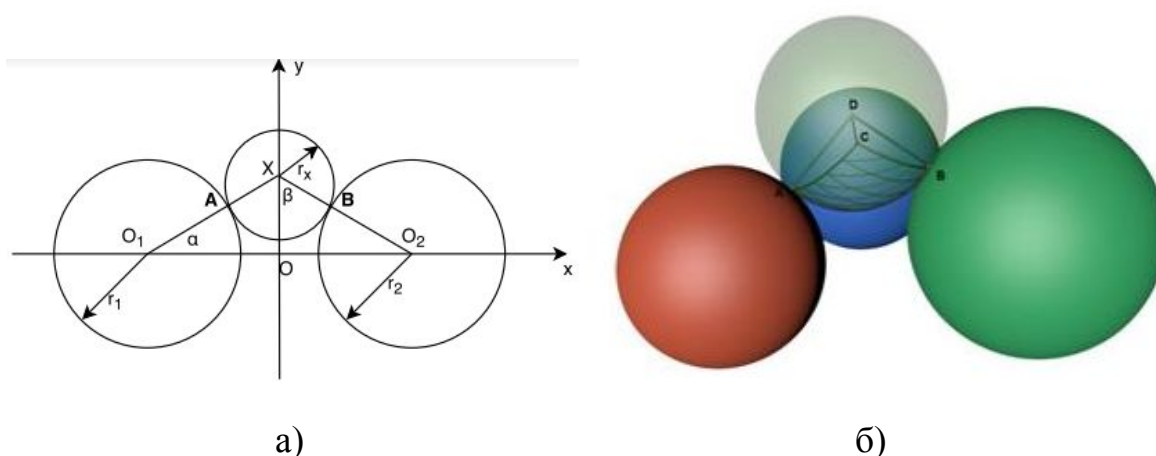


рис.3 - атом растворителя касается одновременно двух и трех атомов

Теперь можно перейти к разработке алгоритма построения. Для начала нужно определить все наборы из трех атомов, которых потенциально может касаться сфера растворителя. Для каждого из таких наборов определяется положение сферы растворителя, и если в этом положении сфера не пересекает ни один из атомов молекулы, то образуемая при помощи этой сферы часть ее поверхности добавляется к итоговой молекулярной поверхности. После чего определяются пары атомов, которых может одновременно касаться сфера растворителя и для них строится тороидальная часть поверхности. Для того, чтобы проверить, может ли растворитель касаться выбранных двух атомов

одновременно, нужно проверить, что расстояние между ними меньше чем сумма их радиусов и диаметра растворителя. Для случая с тремя атомами эту проверку нужно провести для каждой пары из тройки. После чего для каждого случая на основе ключевых параметров строится поверхность, которая добавляется в результирующую.

Для создания вогнутого сферического фрагмента необходимо уметь проецировать треугольник на сферу. Для начала рассмотрим случай, когда две точки лежат на поверхности шара и нам требуется спроецировать середину отрезка, соединяющего эти две точки, на шар(рис.5). Координаты точки С вычисляются по формуле:

$$C = X + \frac{r_x}{|XC_1|}XC_1, \quad C_1 = A + \frac{AB}{2}.$$

Теперь рассмотрим случай с треугольником. Пусть теперь на шаре лежат три точки - вершины треугольника. Для того, чтобы спроецировать его на шар, нужно разбить треугольник на четыре треугольника(рис.4) и спроецировать каждую середину отрезка на сферу как в случае с двумя точками. После чего повторяем процесс для каждого из новых треугольников до тех пор, пока не будет достигнута заданная глубина рекурсии.

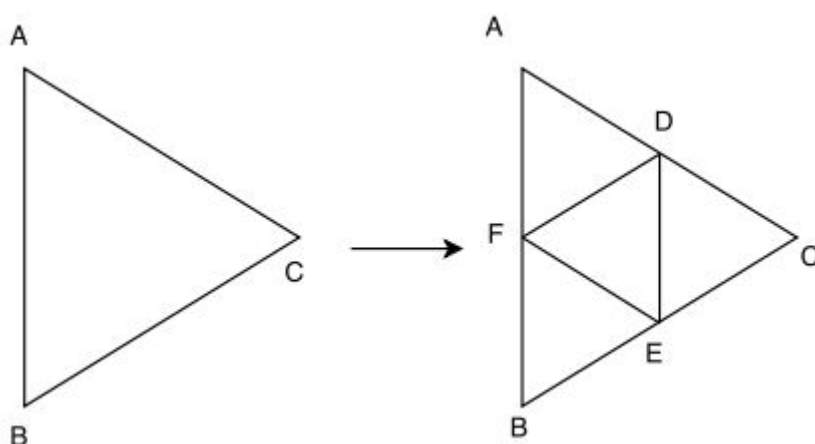


рис.4-шаг разбиения треугольника



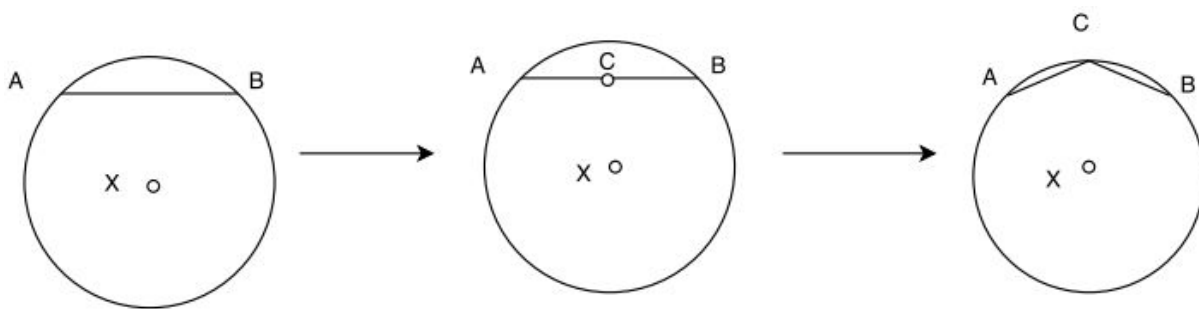


рис.5 - проекция середины отрезка на сферу

Используя данный метод, также можно создавать выпуклые сферические поверхности. Для каждого атома нужно определить вписанный в шар с радиусом Ван-дер-Ваальса этого атома правильный тетраэдр. После чего для каждой грани тетраэдра нужно провести процедуру проекции. В результате мы получим сферическую поверхность из правильных треугольников.

Рассмотрим подробнее построение вогнутой сферической поверхности. Для каждой тройки атомов, которых может касаться растворитель, нужно определить его положение. Для этого нужно определить расстояние от проекции его центра на плоскость, образованную центрами атомов, до любой из сторон, а также расстояние до этой плоскости. Можно перенести данную плоскость в плоскость  $OXY$  для более удобных расчетов(рис.6). Тогда положение растворителя определяется по формулам:

$$\cos(\gamma) = \frac{(|XO_2|^2 - |O_1X|^2 - |O_1O_2|^2)}{-2|O_1X||O_1O_2|}, \sin(\gamma) = \sqrt{1 - \cos^2(\gamma)},$$

$$|OX| = |O_1X| \cdot \sin(\gamma), r = |XO|, \cos(\psi) = \frac{(XO, XO_3)}{|XO| |XO_3|}, X = (0, r\cos(\alpha), r\sin(\alpha)),$$

$$XO = (0, -r\cos(\alpha), -r\sin(\alpha))$$

$$XO_3 = (x_3, d - r\cos(\alpha), -r\sin(\alpha)), O_3 = (x_3, d, 0), \cos(\beta) = \frac{(O_2O_1, O_2O_3)}{|O_2O_1||O_2O_3|},$$

$$\sin(\beta) = \sqrt{1 - \cos(\beta)^2}, d = \sin(\beta) \cdot |O_2O_3|,$$

$$(XO, XO_3) = (d - r\cos(\alpha))(-r\cos(\alpha)) + r^2\sin^2(\alpha), |XO_3| = r_x + r_3,$$

$$\cos(\psi) = \frac{-d\cos(\alpha) + r}{r_x + r_3}, \quad \cos(\alpha) = \frac{\cos(\psi)(r_x + r_3) - r}{-d},$$

$$N_z = \frac{(O_1 O_3 \times O_1 O_2)_z}{|O_1 O_3 \times O_1 O_2|}, \quad N_y = \frac{(N_z \times O_2 O_1)_y}{|N_z \times O_2 O_1|}, \quad O = O_1 + \frac{(r_1 + r_x) \cos(\gamma)}{|O_1 O_2|} \cdot O_1 O_2,$$

$$X = O + r \cos(\alpha) \cdot N_y + r \sin(\alpha) \cdot N_z, \quad -X = O + r \cos(\alpha) \cdot N_y - r \sin(\alpha) \cdot N_z$$

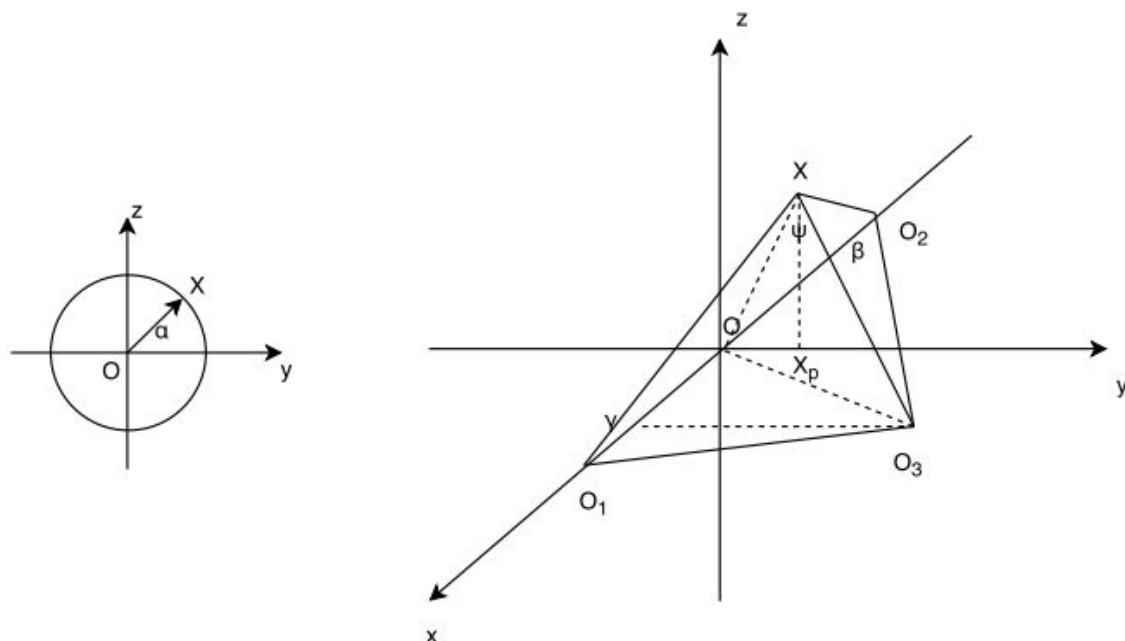


рис.6 - схема построения вогнутой поверхности

После этого нужно проверить не пересекается ли растворитель в позиции  $X$  и  $-X$  с другими атомами не из этой тройки. Если не пересекается, то нужно определить точки касания растворителя с атомами из данной тройки и полученный треугольник из трех точек спроецировать на сферу растворителя. Итоговый результат продемонстрирован на рис.7.

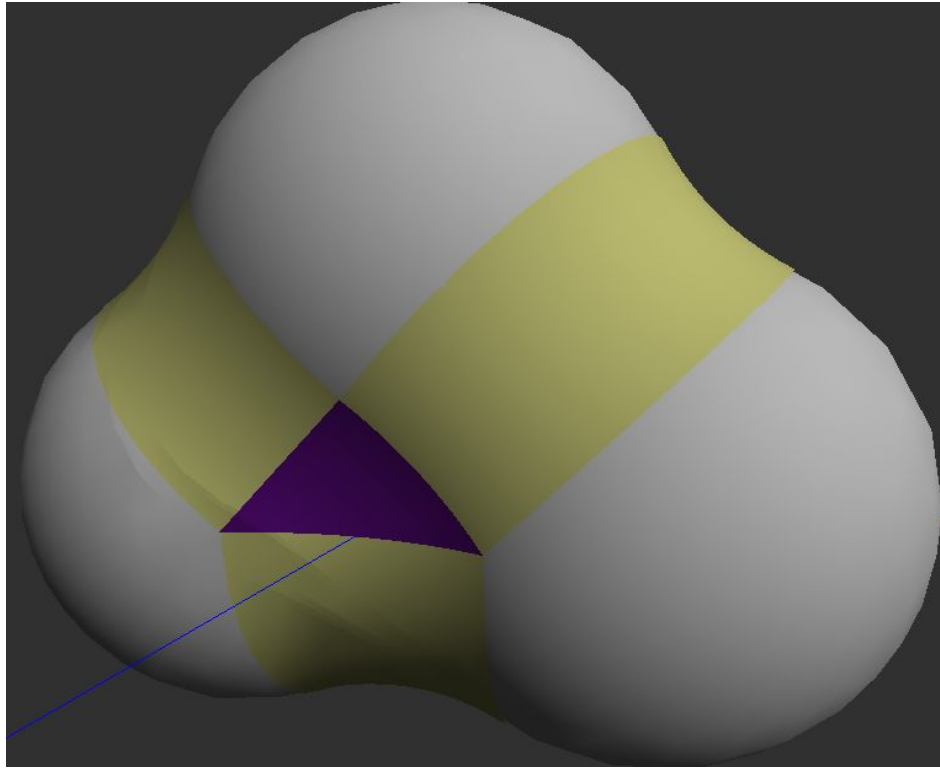


рис. 7- вогнутая поверхность изображена фиолетовым цветом

Рассмотрим подробнее расчет основных параметров для тороидальной поверхности. Для удобства перейдем из трехмерного случая к двумерному и перенесемся в начало координат как показано на рис.8. Для построения тороидальной поверхности необходимо найти позицию точки  $X$  и позицию точки  $A$  вместе с углом  $\beta$  по формулам:

$$\cos(\alpha) = \frac{(|XO_2|^2 - |O_1X|^2 - |O_1O_2|^2)}{-2|O_1X||O_1O_2|}, \sin(\alpha) = \sqrt{1 - \cos^2(\alpha)}$$

$$|OX| = |O_1X| \cdot \sin(\alpha), \quad |O_1O| = |O_1X| \cdot \cos(\beta)$$

$$X = (0, |OX|), \quad O_1 = (-|O_1O|, 0)$$

$$XA = \frac{r_x}{(r_x + r_1)} XO_1, \quad A = X + XA$$

$$\cos(\beta) = \frac{(|O_1O_2|^2 - |O_1X|^2 - |O_2X|^2)}{-2|O_1X||O_2X|}, \beta = \arccos(\cos(\beta))$$

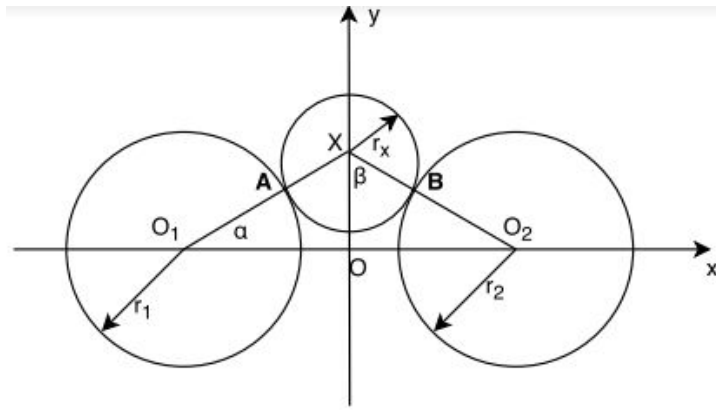


рис.8 - схема для расчета основных параметров  
тороидальной поверхности

Тороидальная поверхность получится при заметании дуги  $AB$  при вращении вокруг оси  $Ox$ . Для этого дуга  $AB$  разбивается на нужное количество кусков, которое зависит от требуемой точности при построении поверхности, обозначим его как  $n$ . Для этого используется угол  $\beta$ , из которого определяется смещение угла  $d_n = \frac{\beta}{n}$ . При помощи данного смещения вектор  $XA$  вращается вокруг оси  $XA \times XO$  ровно  $n$  раз и переносится на вектор  $OX$ , при этом каждое его положение запоминается. В итоге получается разбиение дуги  $AB$ , которое в дальнейшем используется при вращении вокруг оси  $OX$   $m$  раз на угол  $d_m = \frac{360}{m}$ , где  $m$  определяется в зависимости от требуемой точности поверхности. После того, как поверхность построена в начале координат, необходимо перенести ее в исходное положение. Для этого определяется:

$$\text{ось вращения } axis = OX \times O_1O_2 ,$$

$$\text{угол } \cos(\gamma) = \frac{(OX, O_1O_2)}{|OX| |O_1O_2|} .$$

После чего смещается на вектор:

$$A = O_1 + \frac{(r_1 + r_x) \cdot \cos(\alpha)}{|O_1O_2|} \cdot O_1O_2 .$$

Результат продемонстрирован на рис.9.

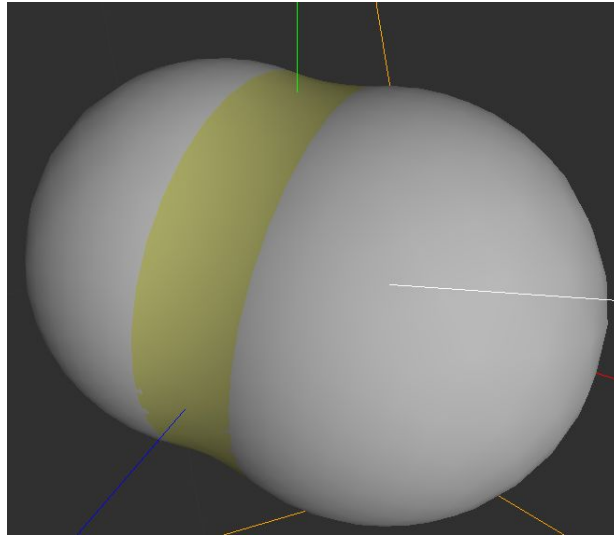


рис.9 - желтым цветом показана тороидальная поверхность

Но бывают случаи, когда растворитель пересекает ось  $Ox$  и возникает самопересечение при вращении (рис.10). Чтобы устранить это самопересечение, нужно свести алгоритм к предыдущему случаю. На рис.11 продемонстрирована схема положения сферы растворителя при самопересечении во время вращения. Чтобы исключить самопересечение вместо дуги  $AB$  нужно вращать дуги  $AA_1$  и  $B_1B$ . Для этого нужно определить положение точек  $A_1$  и  $B_1$  и углы  $\beta_1$  и  $\beta_2$  по формулам:

$$\text{уравнение окружности } x = r_x \cdot \cos(\delta), y = r_x \cdot \sin(\delta) + |OX|,$$

$$A_1 = (-r_x \cdot \cos(\delta), 0), B_1 = (r_x \cdot \cos(\delta), 0)$$

$$y = 0 = r_x \cdot \sin(\delta) + |OX|, \sin(\delta) = \frac{-|OX|}{r_x}, \cos(\delta) = \sqrt{1 - \sin^2(\delta)}$$

$$\cos(\beta_1) = \frac{(XA, XA_1)}{|XA| |XA_1|}, \cos(\beta_1) = \frac{(XB_1, XB)}{|XB_1| |XB|}$$

Итоговый результат продемонстрирован на рис.12.

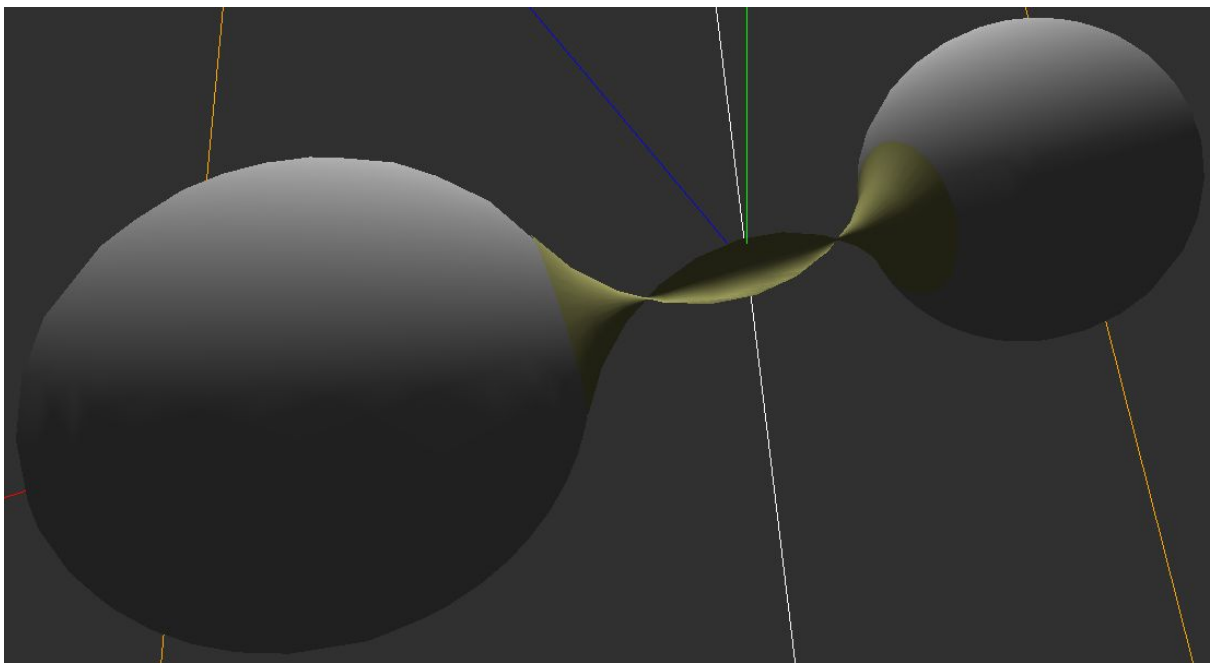


рис.10 - самопересечение тороидальной поверхности

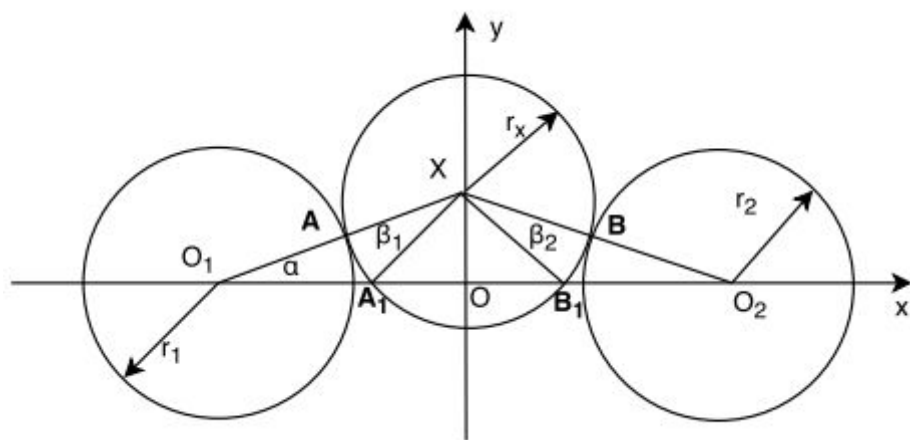


рис.11 - схема для расчета основных параметров тороидальной поверхности при самопересечении

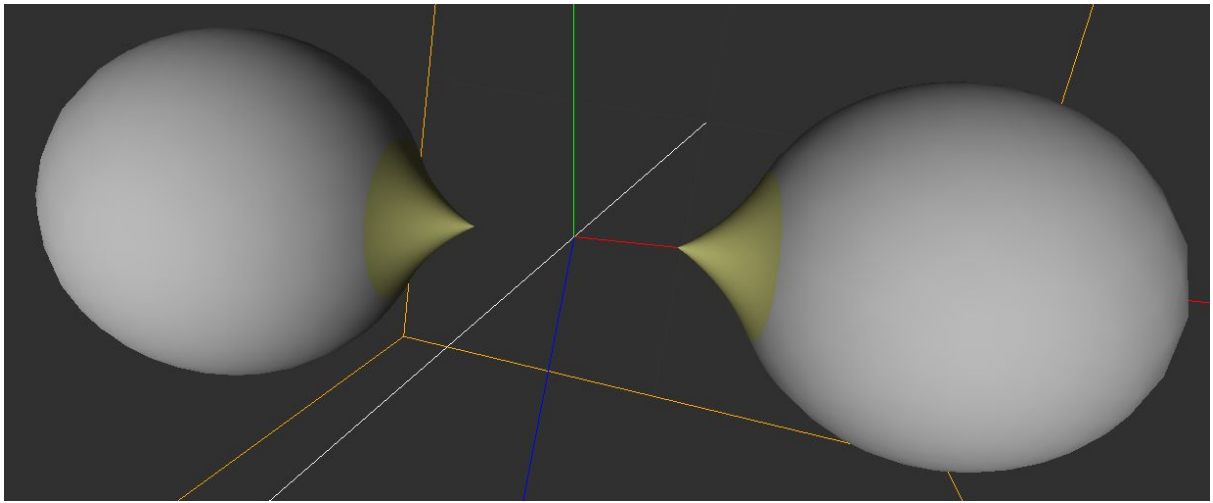


рис.12 - тороидальная поверхность после устранения самопересечения

## Реализация

В качестве языка разработки был выбран JavaScript(ECMAScript 6) - мультипарадигменный язык программирования. Он является реализацией языка ECMAScript (стандарт ECMA-262). HTML5 позволяет сделать удобный интерфейс и связать его элементы с исполняемым кодом на JavaScript. Для создания и отображения трехмерной сцены была использована библиотека Three.js[10] - обертка над WebGL, позволяющая упростить процесс написания веб-приложений с 3D графикой.

Основной сценарий работы с Three.js заключается в следующем: сначала создается объект сцены класса THREE.Scene, создается объект камеры, после чего к сцене добавляются пользовательские объекты для отрисовки. Отрисовка происходит в функции, которая передается параметром в функцию requestAnimationFrame( $f$ ), при этом для получения анимации  $f$  сама должна вызывать данную функцию.

Разработанное приложение позволяет:

- загружать молекулу из формата .mol2
- загружать молекулу из JSON формата
- загружать молекулу по ее id из базы данных[9] при помощи REST API
- управлять светом и камерой для просмотра молекулы
- строить три вида поверхности и управлять переключением отображения одной из них

Структура проекта состоит из нескольких файлов(рис.13). В файле Vector3.js находится вспомогательный класс реализующий трехмерный вектор со стандартным набором операций - векторное и скалярное произведение, нормализация, перенос, умножение на число и т.д. В файле Tables.js расположены таблицы цветов и радиусов атомов[4,5]. Файл Scene.js содержит в себе класс Scene отвечающий за создание и отрисовку сцены со всеми



необходимыми объектами - светом, камерой, молекулой. В файле ParserMol2.js находится парсер для формата .mol2, который преобразует содержимое файла в JSON формат. В Molecule.js находится класс отвечающий за создание молекулярной поверхности. В нем происходит процесс генерации каждого из трех видов поверхности, которые потом используются классом Scene при отрисовке. App.js содержит в себе функции для работы с загрузкой и чтением содержимого файлов из системы и получения молекулы при помощи REST API.

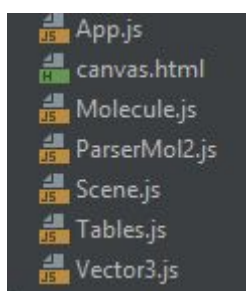


рис.13 - структура проекта

На рис.14 изображен интерфейс разработанного приложения. В нижней части окна расположена кнопка для выбора файла, а также окно для ввода id молекулы из базы данных[9]. В левом верхнем углу расположен счетчик кадров в секунду. По кнопкам  $R$ ,  $T$ ,  $Y$  можно менять режим отображения молекулы(рис.15 и рис.16).

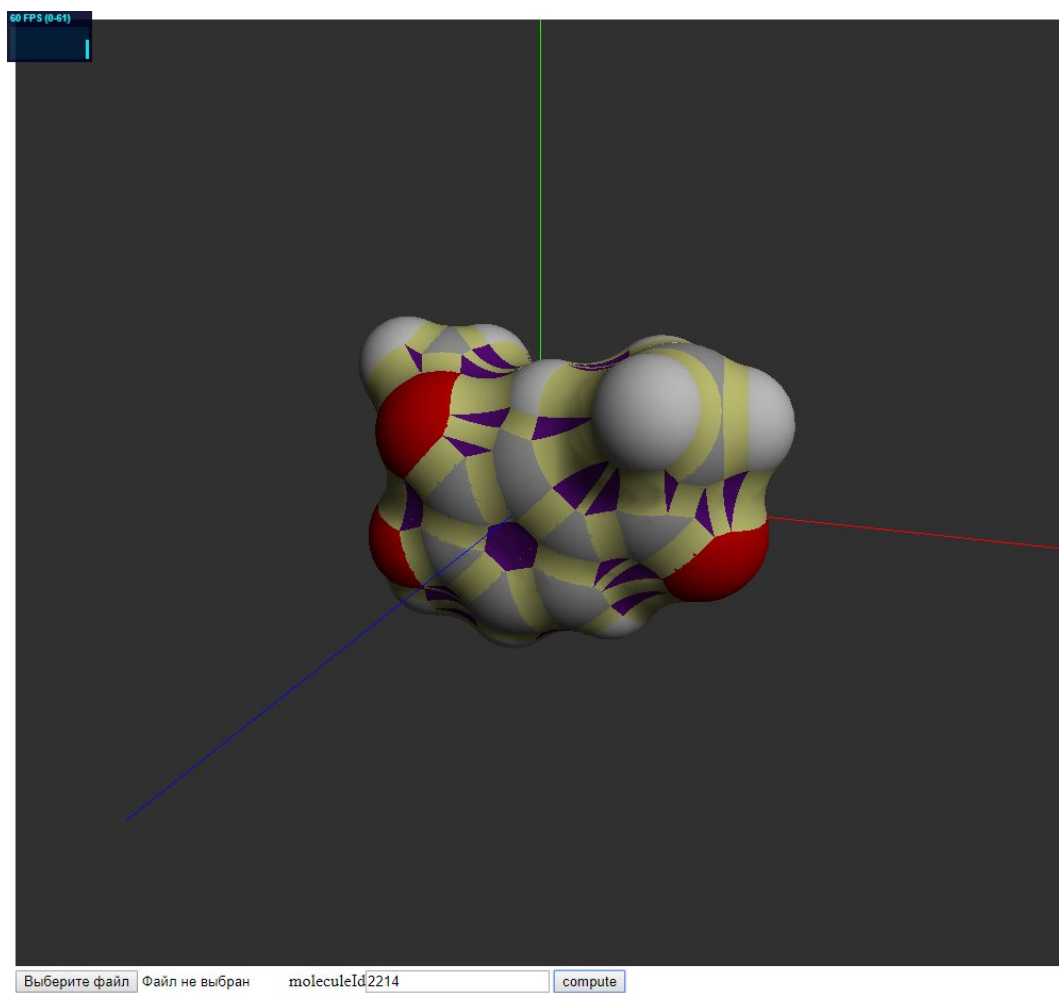


рис.14 - интерфейс разработанного приложения

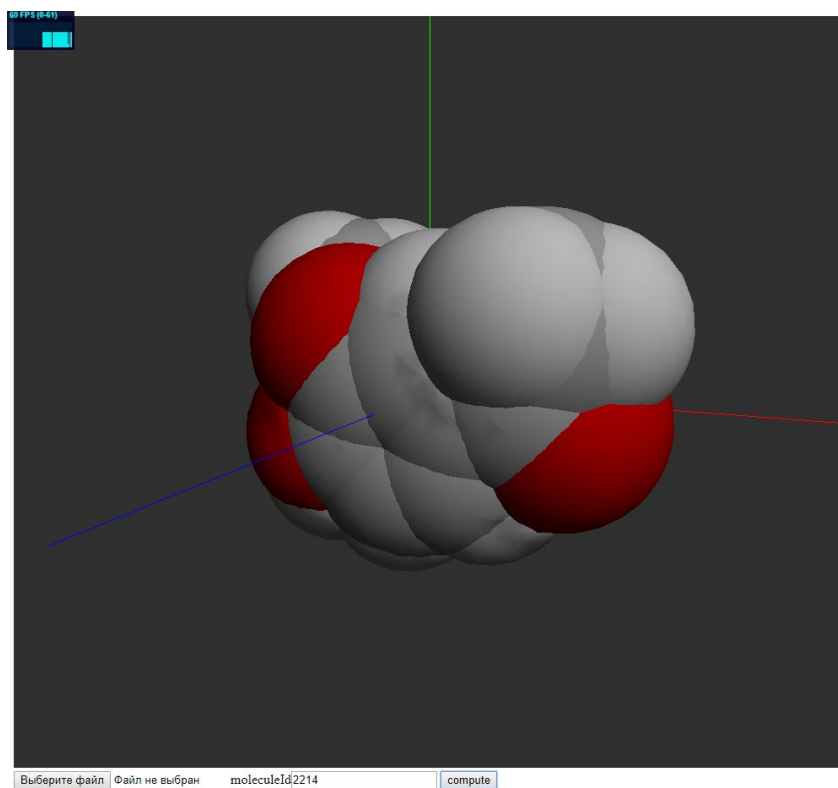


рис.15 - один из режимов отображения молекулы

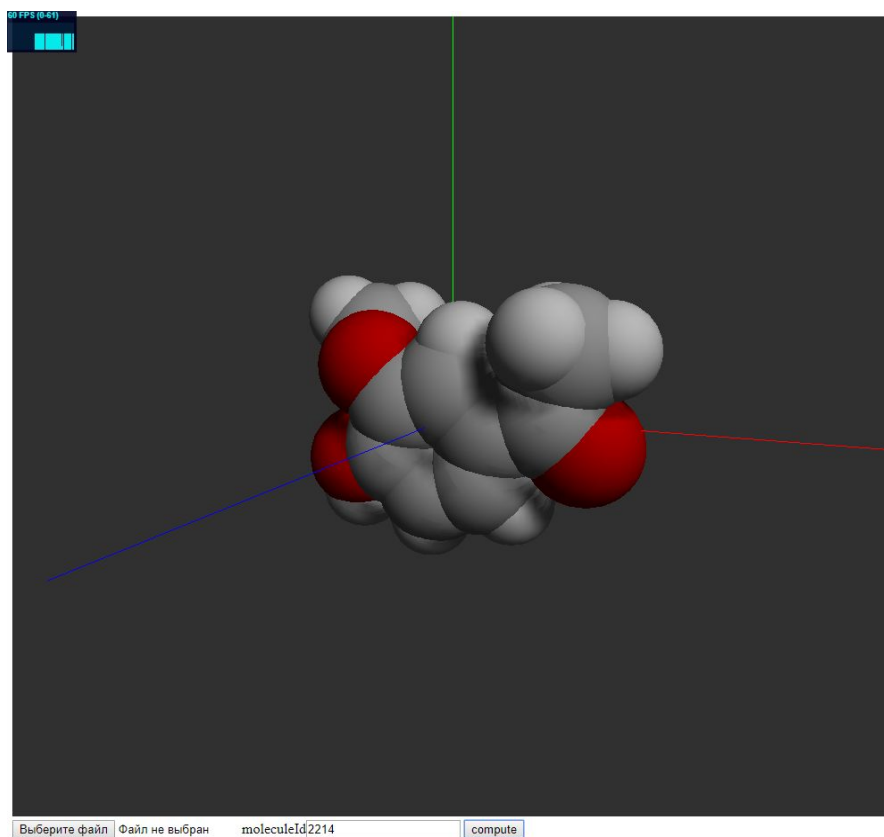


рис.16 - один из режимов отображения молекулы

При помощи клавиш W,A,S,D можно управлять положением источника света, а при зажатой левой кнопки мыши можно вращать камеру вокруг молекулы для детального рассмотрения каждой из ее частей при помощи приближения и отдаления камеры, регулируемых колесиком мыши.

## Тестирование

Тестирование данного приложения производилось на двух системах: настольный персональный компьютер (i5-3570, 3.4Ghz(4 ядра), gpu geforce gtx670) под управлением ОС Windows 10 и ноутбук (i5-8250U, 1.6 Ghz(4 ядра), Intel UHD Graphics 620) под управлением Linux Fedora 27.

Тестирование производилось в браузерах: Google Chrome (версия 65.0.3325.181), Opera (версия 51.0.2830.55) и Mozilla Firefox (версия 59.0.1). Были произведены следующие замеры: количество кадров в секунду и время, затрачиваемое на полное создание поверхности, в зависимости от количества атомов. В таблицах 1 и 2 приведены результаты тестирования.

количество атомов	количество кадров (Chrome)	время создания (Chrome)	количество кадров (Opera)	время создания (Opera)
3	60	0,05 сек.	60	0,05 сек.
49	60	0,5 сек.	60	0,58 сек.
91	60	0,98 сек.	60	1,17 сек.
156	48	1,34 сек.	50	1,3 сек.
200	32	2,1 сек.	32	2,3 сек.
246	40	2 сек.	42	2,35 сек.
304	16	4 сек.	16	4,7 сек.

Таблица 1 - тестирование на персональном компьютере

количество атомов	количество кадров (Chrome)	время создания (Chrome)	количество кадров (Firefox)	время создания (Firefox)
3	60	0,05 сек.	60	0,07 сек.
49	60	0,5 сек.	43	0,7 сек.
91	50	0,8 сек.	20	1,3 сек.
156	35	1,38 сек.	14	2,17 сек.
200	26	1,9 сек.	9	3,35 сек.
246	32	1,8 сек.	12	2,41 сек.
304	13	3,6 сек.	5	5 сек.

Таблица 2 - тестирование на ноутбуке

## **Результаты**

В ходе выполнения курсовой работы было реализовано расширяемое приложение для визуализации молекулярной поверхности в браузере, способное создавать несколько ее видов с возможностью детального рассмотрения каждого из них. Для этого были изучены язык JavaScript(ESMAScript 6) и библиотека Three.js, а также язык разметки HTML5, используемый для создания веб-интерфейса приложения, и теоретические основы построения молекулярных поверхностей.

## Литература

1. Computer-Aided Molecular Design: Theory and Applications. Jean-Pierre Doucet, Jacques Weber. стр. 239-265.
2. CPK coloring. URL: [http://www.wikiwand.com/en/CPK\\_coloring](http://www.wikiwand.com/en/CPK_coloring)
3. Solvent-Accessible Surfaces of Proteins and Nucleic Acids. Michael L. Connolly. Science, New Series, Vol. 221, No. 4612 (Aug. 19, 1983), стр. 709-713
4. Van Der Waals Radius of the elements.  
URL: <http://periodictable.com/Properties/A/VanDerWaalsRadius.v.html>
5. The RasMol cpk color scheme.  
URL: <http://life.nthu.edu.tw/~fmhsu/rasframe/CPKCLRS.HTM>
6. GLmol. URL: <http://webglmol.osdn.jp/index-en.html>
7. PyMol. URL: <https://pymol.org/2/>
8. Avogadro. URL: <https://avogadro.cc/>
9. PubChem — база данных химических соединений.  
URL: <https://pubchem.ncbi.nlm.nih.gov/>
10. Three.js - библиотека для работы с 3D графикой. URL: <https://threejs.org/>