

Глава 6. Прямая задача теории колебаний молекул в одной программе

6.1. Короткая житейская история, поясняющая, какую пользу можно получить от чтения этой главы

В свои 32 года, в 1967 году, будучи преподавателем кафедры физики Тимирязевской сельхоз академии, я находился в глубоком профессиональном кризисе. Два года назад я защитил кандидатскую диссертацию по методам радиометрии. Написал и уже издал небольшую книжку на эту тему «Радиометрия малых активностей радиоактивных препаратов», тут же переизданную в Польше. Но я совершенно не понимал, что мне делать дальше, когда у меня нет доступа к лаборатории, где я аспиранствовал. Мои ожидания, что изданная книжка поможет мне найти заинтересованные во мне как в специалисте научные лаборатории, не оправдались. За 5 лет после выхода в свет книжки только один раз ко мне приехал проконсультироваться сотрудник Обнинского метеоцентра.

Тут на кафедру физики пришел доктор физ.-мат. наук Л.А. Грибов, которому было 34 года. Он присмотрелся к составу кафедры и решил всех постепенно заменить сотрудниками, более способными к самообучению, чем имевшиеся в наличии. Присмотрелся ко мне и предложил подумать, не сменить ли мне научную тематику – бросить радиометрию, а взяться за работу по его тематике, по теории колебаний молекул.

Думать мне было нечего, я, утопая, готов был ухватиться за любую научную соломинку. Конечно, я ухватился за Грибова.

Это вводная часть истории. А вот сама история о том, как я хватался за новую тематику. Я уверен, если читатель, еще не вошедший глубоко в теорию колебаний молекул и в структуры ее вычислительных алгоритмов, пройдет со мной эту историю заново, то он овладеет этой теорией в достаточной мере, чтобы заниматься моделированием колебательных состояний сложных молекул и при этом чувствовать себя вполне комфортно. Я освоил эту науку в такой мере за 2 месяца.

Я вооружился книжкой Грибова, изданной в ЛГУ, где он уже прочел курс лекций по такому спецкурсу: Л.А. Грибов. Введение в теорию и расчет колебательных спектров многоатомных молекул. Изд. ЛГУ, Ленинград, 1965, 125 с.

Кроме того, я вооружился блокнотом, карандашом, резинкой и логарифмической линейкой. Нарисовал в блокноте простую модель молекулы этилена и выполнил вручную весь описанный в книге алгоритм расчета частот в колебательном спектре этой молекулы. После этого для меня загадок в теории колебаний молекул и в методике построения ее вычислительных процедур уже не было.

Затем я освоил программирование непосредственно в машинных кодах ЭВМ Минск 22, к 1969 году написал все программы для формирования колебательных моделей и для расчета их колебательных спектров (прямая задача теории колебаний молекул). И с тех пор всё моделирование колебательных состояний молекул в исследовательской группе Грибова выполняется только с помощью программ собственного изготовления. Меняются компьютеры, операционные системы и языки программирования, наши программы становятся мощнее и умнее. Но идеология вычислительных алгоритмов теории колебаний молекул, вплоть до полимеров,

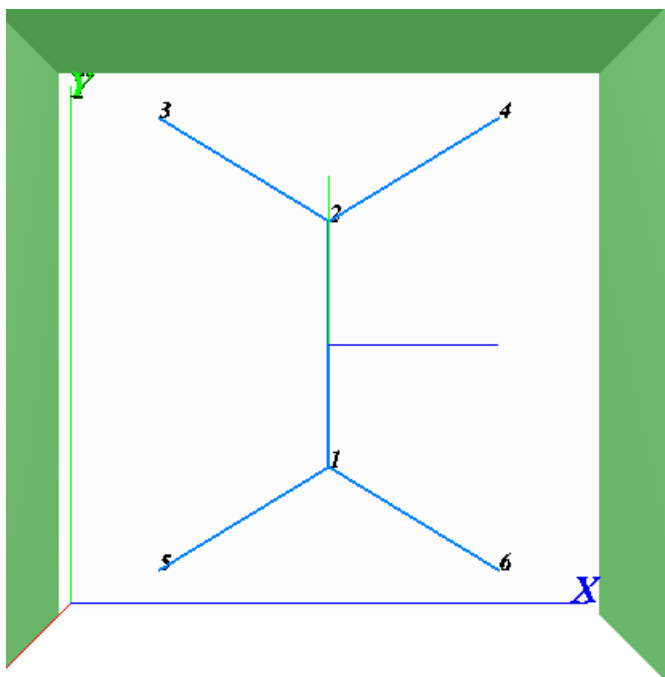
кристаллов и наночастиц, у нас остается той же самой, что была счастливо угадана к 1970 году, когда у нас появились уже и программы решения обратных задач этой теории.

Вот я и приглашаю не очень умудренного в теории читателя – присоединяйтесь, барон, присоединяйтесь...

Конечно, сейчас смешно брать в руки логарифмическую линейку. Сейчас у просвещенного читателя в руках MatLab. И славно. Пройдем в этой главе все формулы решения прямой задачи теории колебаний молекул вместе, пользуясь средствами MatLab. При этом осмотрим все промежуточные результаты вычислений на примере простой модели молекулы этилена. И тогда я буду уверен, что у меня с читателем не осталось загадок в постановке и решении прямой задачи теории колебаний молекул.

6.2. Формирование машинного представления колебательной модели молекулы этилена

Мы хотим перенести в компьютер вот такое наше представление о модели молекулы этилена.



Находим данные о молекуле. Их, например, можно получить, пользуясь программой ChemOffice.

Координаты (радиусы-векторы) атомов в приведенной на рисунке декартовой системе координат выглядят так.

	x	y	z
1	0.0000	-0.6690	0.0000
2	0.0000	0.6690	0.0000
3	-0.9284	1.2324	0.0000
4	0.9284	1.2324	0.0000
5	-0.9284	-1.2324	0.0000
6	0.9284	-1.2324	0.0000

В среде MatLab настроимся на подходящую текущую директорию (например, C:\ML_Works). Там будем создавать скрипты решения прямой задачи. Скрипты (m-файлы) удобны во многих отношениях. Их можно запускать неоднократно после исправления всех ошибок. Переменные, объявляемые в скриптах, всегда доступны для просмотра в среде MatLab.

Напишем и сохраним в ML_Works скрипт в файле Atoms.m:

```
% AtomCoords  
  
AtomCoords = [ 0.0000    -0.6690     0.0000; ...  
               0.0000     0.6690     0.0000; ...  
              -0.9284     1.2324     0.0000; ...  
               0.9284     1.2324     0.0000; ...  
              -0.9284    -1.2324     0.0000; ...  
               0.9284    -1.2324     0.0000 ];
```

После выполнения этого скрипта в рабочей области MatLab появляется переменная AtomCoords размера <6x3 double>, содержащая массив координат атомов модели.

Для того, чтобы чтение этой главы и воспроизведение приведенных в ней скриптов имело смысл, полезно просматривать возникающие после выполнения скрипта конечные результаты. Это делается вызовом переменных из рабочей области в окно редактора массивов.

Напишем и сохраним в ML_Works скрипт в файле Masses.m:

```
% AtomMasses  
  
AtomMasses = [12 12 1.088 1.088 1.088 1.088 ];
```

После выполнения этого скрипта в рабочей области MatLab появляется переменная AtomMasses, содержащая массив масс атомов модели.

Напишем и сохраним в ML_Works скрипт в файле Connect.m:

```
% Bonds  
  
Bonds = [ 1 2 ; ...  
          2 3 ; ...  
          2 4 ; ...  
          1 5 ; ...  
          1 6 ];
```

После выполнения этого скрипта в рабочей области MatLab появляется переменная Bonds, содержащая массив номеров атомов модели, соединенных валентными связями.

При написании этого скрипта следует сразу обратить внимание на порядок следования номеров атомов в каждой строке массива связей. Дело в том, что в дальнейшем, при вычислении абсолютных интенсивностей в ИК спектре модели, нам понадобятся такие величины, как дипольные моменты связей и их производные по колебательным координатам. Для эквивалентных связей модели эти величины должны удовлетворять полной и локальной симметрии модели. Это приводит к требованию, чтобы единичные векторы валентных связей также удовлетворяли этим условиям. Наша будущая программа проводит единичный вектор от первого атома в описании данной связи ко второму. Поэтому таблица описания связей в скрипте должна быть составлена именно так. Если мы напишем в последней строке таблицы «6 1», а не «1 6», то симметрия модели будет нарушена, что может привести к неправильному вычислению интенсивностей. Первая связь в таблице является полностью симметричной, поскольку образована двумя эквивалентными атомами, находящимися в эквивалентных окружениях. По физике, такая связь не может вносить вклада в ИК спектр модели, ее электрооптические параметры должны быть

нулевыми. Поэтому для первой связи в нашей модели мы можем строку в таблице задать либо как «1 2» либо как «2 1».

Кстати, матрицу коннективности модели (таблицу валентных связей) и таблицу масс атомов можно получить средствами программы ChemOffice. Для крупных моделей это может быть удобно. Но надо иметь в виду, что данные о модели из программы ChemOffice не обладают той точностью, которая позволяла бы учитывать симметрию модели и автоматизировать формирование характерных силовых и электрооптических параметров модели.

Теперь надо позаботиться о системе колебательных координат модели. Дело в том, что по ряду причин декартова система координат не совсем удобна для анализа колебаний молекулы. В декартовой системе удобно изображать равновесное состояние модели, но крайне неудобно изображать внутренние колебательные движения частей модели. Подробный анализ этих неудобств можно найти в книгах Л.А. Грибова. Я же, для примера, сошлюсь только на одно неудобство.

В декартовой системе координат каждая материальная точка обладает тремя степенями свободы. Молекула, состоящая из N атомов, обладает $3N$ степенями свободы. Учитывать все $3N$ степеней свободы полезно, если мы хотим проследить не только за колебаниями атомов в молекуле, но и за ее поступательным и вращательным движением в пространстве. Если же нас интересуют только колебания атомов, то удобно мысленно поместить молекулу в неподвижную лабораторную систему декартовых координат, как это показано на рисунке. Там лабораторные оси обозначены заглавными X и Y , а внутренние декартовы оси, связанные с моделью, никак не обозначены, но раскрашены теми же цветами. Мы не хотим, чтобы оси молекулы вращались или перемещались в лаборатории. Тогда у нас должны пропасть 6 степеней свободы молекулы – три направления движения центра масс молекулы и три оси моментов ее вращения. Вот для того, чтобы отделить колебания от перемещений и вращений, и надо переходить в другую систему координат. Эти обобщенные координаты называются естественными колебательными координатами. Они теснейшим образом связаны со структурными особенностями модели, чем и помогают анализировать колебательные движения отдельных частей молекулы друг относительно друга.

Простейший тип колебательных координат это координаты типа q растяжения валентной связи. Для равновесного состояния валентной связи эта координата равна нулю. Если связь в колебаниях растянулась, то $q > 0$. Эта величина показывает, на сколько ангстрем удлинилась связь. Если связь в колебаниях сжалась, то $q < 0$. По определению координата q скалярна. А смещения атомов в колебаниях описываются приращениями радиусов-векторов атомов. Для перехода от одной системы координат к другой требуется линейное преобразование

$$q = \mathbf{b}_i \Delta \mathbf{r}_i + \mathbf{b}_j \Delta \mathbf{r}_j.$$

Здесь i – номер первого атома в данной связи, j – номер второго атома в данной связи, \mathbf{b} – векторный коэффициент, $\Delta \mathbf{r}$ – вектор смещения атома из положения равновесия. Можно показать, что коэффициенты \mathbf{b} выражаются через единичные направляющие векторы соответствующих валентных связей. Если задан единичный направляющий вектор \mathbf{e} данной связи, формула выглядит так.

$$q = -\mathbf{e} \Delta \mathbf{r}_i + \mathbf{e} \Delta \mathbf{r}_j.$$

Для того, чтобы сразу преобразовать все смещения атомов в колебательные координаты типа q , надо выполнить матричное преобразование $q = \mathbf{B}\Delta\mathbf{r}$. Теперь здесь q это матрица-столбец значений всех координат типа q модели, $\Delta\mathbf{r}$ – матрица-столбец всех смещений атомов в деформированном состоянии модели. \mathbf{B} представляет собой прямоугольную матрицу, элементами которой являются векторы \mathbf{b} . Число строк матрицы равно числу координат типа q , а число столбцов равно числу атомов в модели. Матрица \mathbf{B} в основном заполнена нулями. Ненулевые векторные элементы стоят только там, где номер столбца совпадает с номером атома, входящего в описание данной координаты типа q . В конкретной строке матрицы, относящейся к связи, направляющий единичный вектор которой равен \mathbf{e} , в столбце с номером i стоит элемент $\mathbf{b}_i = -\mathbf{e}$. В столбце с номером j стоит элемент $\mathbf{b}_j = \mathbf{e}$.

Как видно, матрица \mathbf{B} имеет простую структуру и вычисляется для координат типа q очень просто. В теории колебаний молекул и в расчетных алгоритмах эта матрица играет очень существенную роль. Поэтому сейчас сосредоточимся на скрипте, который будет формировать эту матрицу для координат типа q нашей модели. Напишем скрипт и сохраним его в файле `Bq.m`:

```
% B submatrix for q-coordinates

nAtoms = size(AtomCoords, 1);
nBonds = size(Bonds, 1);
e_Bonds = zeros(nBonds, 3); % bonds unit vectors
s_Bonds = zeros(nBonds, 1); % bonds lengths
% Vector-Matrix Bq is represented with three scalar matrices
Bqx = zeros(nBonds, nAtoms);
Bqy = zeros(nBonds, nAtoms);
Bqz = zeros(nBonds, nAtoms);
for i = 1:nBonds
    iatom = Bonds(i, 1);
    jatom = Bonds(i, 2);
    [e,s] = atom_pair_vector(iatom, jatom, AtomCoords);
    e_Bonds(i,:) = e;
    s_Bonds(i) = s;
    Bqx(i, iatom) = -e(1);
    Bqx(i, jatom) = e(1);
    Bqy(i, iatom) = -e(2);
    Bqy(i, jatom) = e(2);
    Bqz(i, iatom) = -e(3);
    Bqz(i, jatom) = e(3);
end % Bonds
```

Для обеспечения работоспособности данного скрипта потребуется сохранить в рабочей директории файл `atom_pair_vector.m`. Этот файл содержит текст функции, которая находит значения переменных e и s . Вот текст этой функции

```
function [r,varargout] = atom_pair_vector(Atom1, Atom2, AtomCoords)
% construct unit vector from Atom1 to Atom2 using AtomCoords

r = AtomCoords(Atom2,:) - AtomCoords(Atom1,:);
s = norm(r);
r = r/s;
if nargin > 0, varargout{1} = s; end % if
```

После выполнения этого скрипта в рабочей области MatLab появляются переменные `nAtoms`, `nBonds` и три скалярных компонента векторной субматрицы $\mathbf{B}q$ будущей полной матрицы \mathbf{B} . Кроме того, для дальнейших прикладных расчетов и для справок одновременно рассчитываются и сохраняются в рабочей области направляющие векторы `e_Bonds` и длины `s_Bonds` связей. Просмотрите значения этих переменных средствами MatLab, рассматривая одновременно рисунок модели этилена. Все числа должны быть совершенно понятны.

Ранее созданный массив Bonds представляет собой список координат типа q модели с подробным описанием структуры каждой координаты. А возникшие теперь переменные Bqx, Bqy и Bqz позволяют представить себе связь между характером деформации модели и значениями координат типа q.

Поясним это на примере первой же такой координаты. Это координата растяжения связи C1-C2. Для этой координаты в строках матриц Bqx и Bqz стоят только нули. Следовательно, никакие смещения атомов C1 и C2 вдоль осей X и Z не приведут к растяжению этой связи. В первой строке матрицы Bqy для атома C1 коэффициент равен -1. Для атома C2 коэффициент равен +1. Это значит, что преобразование $q = (-1)\Delta r_{1y} + (1)\Delta r_{2y}$ даст положительный максимальный результат только в том случае, если атом C1 в данной деформации смещается в отрицательную сторону оси Y, а атом C2 – в положительную сторону. Любая другая ориентация векторов смещений при деформации даст меньшее значение колебательной координаты. Пусть, например, малое смещение атома C1 направлено вдоль оси X, а такое же малое смещение атома C2 направлено в противоположную сторону. Тогда на рисунке модели мы ясно увидим, что такая деформация приводит к малому повороту связи C1-C2 против часовой стрелки, но не к ее растяжению. Этот же результат, $q = 0$, мы получим с помощью матрицы **B** модели. Очень умная матрица, а нам думать уже не надо. Однако ради понимания смысла всех типов колебательных координат очень полезно провести такой анализ результатов работы наших скриптов и дальше.

Координаты типа q, введенные нами, описывают не все возможные деформации модели. Для полного описания требуется ввести еще координаты других типов. Принято вводить координаты типов α , γ , ρ и χ .

Координата типа α описывает изменение валентного угла при деформации модели. Валентный угол определяется тройкой атомов. Мы должны ввести в машинное представление модели список координат типа α , а также рассчитать все элементы субматрицы Ba полной матрицы **B** модели.

Напишем, ориентируясь на рисунок модели, сохраним в файле Angles.m и выполним скрипт, который создаст в рабочей области таблицу Alphas. Каждая строка этой таблицы описывает структуру одного валентного угла. В строке стоят номера тройки атомов. Вторым стоит номер центрального атома. В каком порядке заданы номера двух крайних атомов – неважно. Приготовим этот скрипт в таком виде.

```
% Alphas
```

```
Alphas = [ 1 2 3 ; ...
           1 2 4 ; ...
           2 1 5 ; ...
           2 1 6 ; ...
           3 2 4 ; ...
           5 1 6 ];
```

Теперь надо определить колебательные координаты типа α для всех описаний валентных углов в данной таблице. В равновесном состоянии модели $\alpha = 0$. Координата $\alpha > 0$, когда при деформации модели данный валентный угол увеличивается. Переход от смещений атомов к значению координаты α задается линейным преобразованием

$$\alpha = \mathbf{b}_1\Delta\mathbf{r}_1 + \mathbf{b}_2\Delta\mathbf{r}_2 + \mathbf{b}_3\Delta\mathbf{r}_3.$$

Здесь 1,2, 3 – индексы атомов, входящих в строку описания данной координаты. Индекс 2 относится к центральному атому валентного угла. Для малых смещений атомов векторные коэффициенты в этой формуле являются константами. При больших деформациях эти коэффициенты зависят от величины деформации, следовательно, при колебаниях модели они

меняются со временем. Мы будем рассматривать только малые деформации. Тогда эти коэффициенты для конкретной координаты можно рассчитать следующим образом.

Находим значения нескольких вспомогательных переменных. Это \mathbf{e}_1 и \mathbf{e}_2 – единичные векторы, протянутые от центрального атома к крайним атомам тех временных связей, которые возникают в описании валентного угла. Переменные s_1 и s_2 – длины этих временных связей. Понадобятся синус и косинус данного валентного угла β . Тогда

$$\mathbf{b}_1 = -\frac{(\mathbf{e}_2 - \mathbf{e}_1 \cos \beta)}{s_1 \sin \beta}, \mathbf{b}_3 = -\frac{(\mathbf{e}_1 - \mathbf{e}_2 \cos \beta)}{s_2 \sin \beta}, \mathbf{b}_2 = -(\mathbf{b}_1 + \mathbf{b}_3).$$

В наших вычислительных алгоритмах принято нормировать длины связей на стандартную длину связи СН в алканах (1.09 Å). Поэтому в программах мы используем не переменную s , а $s/1.09$.

Напишем и сохраним в файле `Va.m` скрипт, который строит субматрицу `Va` полной матрицы `V` модели.

```
% % B submatrix for alpha-coordinates

nAlphas = size(Alphas, 1);
% Vector-Matrix Ba is represented with three scalar matrices
Bax = zeros(nAlphas, nAtoms);
Bay = zeros(nAlphas, nAtoms);
Baz = zeros(nAlphas, nAtoms);
for ia = 1:nAlphas
    Atom1 = Alphas(ia, 1);
    Atom2 = Alphas(ia, 2);
    Atom3 = Alphas(ia, 3);
    [e1,s1] = atom_pair_vector(Atom2, Atom1, AtomCoords);
    s1 = 1.09/s1;
    [e2,s2] = atom_pair_vector(Atom2, Atom3, AtomCoords);
    s2 = 1.09/s2;
    cs = e1*e2';
    sn = sqrt(1 - cs^2);
    b1 = (e2 - e1*cs)*s1/sn;
    b3 = (e1 - e2*cs)*s2/sn;
    b2 = b1 + b3;
    b1 = -b1;
    b3 = -b3;
    Bax(ia, Atom1) = b1(1);
    Bax(ia, Atom2) = b2(1);
    Bax(ia, Atom3) = b3(1);
    Bay(ia, Atom1) = b1(2);
    Bay(ia, Atom2) = b2(2);
    Bay(ia, Atom3) = b3(2);
    Baz(ia, Atom1) = b1(3);
    Baz(ia, Atom2) = b2(3);
    Baz(ia, Atom3) = b3(3);
end % Alphas
```

Выполним скрипт `Va.m`.

Если в модели есть линейные участки, то элементы субматрицы `V` для координат типа γ излома этих участков вводятся иначе, поскольку $\sin \beta = 0$. В нашей модели таких участков нет, и мы этими координатами сейчас заниматься не будем.

Займемся неплоскими колебательными координатами.

Координата типа ρ описывает деформацию такого структурного элемента. Имеется валентный угол, определяемый тремя атомами. Эти атомы определяют и плоскость, в которой лежит валентный угол. К центральному атому угла присоединен четвертый атом. Следовательно, имеется третья связь в данном структурном элементе модели. Эта третья связь может изменять тот угол, который она составляет с нормалью к данной плоскости. Мерой изменения этого угла

служит колебательная координата типа ρ . В частном случае, когда третья связь лежит в плоскости валентного угла, эту координату называют выходом связи из плоскости.

На рисунке нашей модели мы видим, что связь C1-C2 соединена атомом C1 с плоскостью H5- C1- H6. Это дает нам право ввести координату типа ρ , описывающую деформацию этого структурного элемента. Вторая координата данного типа будет описывать выход связи C2-C1 из плоскости H3- C2- H3.

Напишем, сохраним и выполним скрипт OutPlanes.m:

```
% Out-of-Planess
Ros = [ 3  2  4  1 ; ...
        6  1  5  2  ];
```

В каждой строке массива `Ros` четыре номера атомов описывают структуры координаты типа ρ . Первыми указаны три номера атомов, определяющих плоскость, в которой лежит угол. При этом второй номер относится к центральному атому. Четвертый номер относится к той связи, которая выходит из плоскости. Это концевой атом связи, поскольку считается, что данная связь начинается от центрального атома угла. Важно, в каком порядке мы записали первые три номера в строке. Этот порядок по правилу буравчика определяет направление нормали к плоскости и, следовательно, направление движения центрального атома плоскости, когда координата ρ положительна. Таким образом можно согласовать направления движений в координатах этого типа с движениями других типов, чтобы симметрия колебаний соответствовала симметрии равновесной модели. Это также облегчает формирование характерных силовых и электрооптических параметров модели.

Напишем скрипт `Bg.m`, который сформирует субматрицу `Bg` полной матрицы **B**.

```
% % B submatrix for ro-coordinates

nRos = size(Ros, 1);
% Vector-Matrix Ba is represented with three scalar matrices
Brx = zeros(nRos, nAtoms);
Bry = zeros(nRos, nAtoms);
Barz = zeros(nRos, nAtoms);
for ir = 1:nRos
    Atom1 = Ros(ir, 1);
    Atom2 = Ros(ir, 2);
    Atom3 = Ros(ir, 3);
    Atom4 = Ros(ir, 4);

    [e1, s1] = atom_pair_vector(Atom2, Atom1, AtomCoords);
    [e2, s2] = atom_pair_vector(Atom2, Atom3, AtomCoords);
    [e4, s4] = atom_pair_vector(Atom2, Atom4, AtomCoords);
    s1 = 1.09/s1;
    s2 = 1.09/s2;
    s4 = 1.09/s4;
    eAux = cross(e1, e2);
    s = norm(eAux);
    eAux = eAux/s; % it is normal to the Plane
    cs = e4*eAux';
    sn = sqrt(1 - cs^2);
    Auxr1 = (e4 - eAux*cs)/sn; % it is g-vector
    Auxr2 = (eAux - e4*cs)/sn; % it is f-vector
    temp = cross(e2, Auxr1);
    e3 = cross(e1, Auxr1);
    b1 = s1*temp/s;
    b3 = -s2 * e3/s;
    b4 = s4*Auxr2;
    b2 = -(b4 + b1 + b3);
    Brx(ir, Atom1) = b1(1);
    Brx(ir, Atom2) = b2(1);
    Brx(ir, Atom3) = b3(1);
    Brx(ir, Atom4) = b4(1);
```



```

    Bry(ir, Atom1) = b1(2);
    Bry(ir, Atom2) = b2(2);
    Bry(ir, Atom3) = b3(2);
    Bry(ir, Atom4) = b4(2);
    Brz(ir, Atom1) = b1(3);
    Brz(ir, Atom2) = b2(3);
    Brz(ir, Atom3) = b3(3);
    Brz(ir, Atom4) = b4(3);
end % Ros

```

Смысл почти всех переменных в этом скрипте нам уже знаком. Суффиксами и префиксами *Aux* снабжены новые вспомогательные переменные, необходимые для получения конечных результатов.

Выполним этот скрипт. Просматривая субматрицы матрицы **B**, соответствующие разным типам колебательных координат, мы заметим, что все элементы субматриц *B_q* и *B_a* лежат в плоскости XY, а все элементы субматрицы *B_γ* направлены вдоль оси Z. Это позволяет нам сразу прогнозировать для данной модели независимость неплоских координат ρ и соответствующих движений атомов от плоских координат q и a . Это позволяет существенно упростить и решение прямой задачи без учета симметрии, и составление матрицы **C** при решении с учетом симметрии. В других моделях такой независимости движений может и не быть.

Наконец, введем координату типа χ кручения двух плоскостей друг относительно друга. В нашей модели имеются две плоскости H5- C1- H6 и H3- C2- H3, которые могут вращаться в противоположные стороны вокруг оси, определяемой общей связью C1- C2. Такая деформация называется внутренним вращением в молекуле.

Напишем, сохраним и выполним скрипт InnerTurn.m.

```

% His
His = [ 3 2 4 0 1 2 5 1 6 0 2 1 ];

```

Обратим внимание на структуру единственной строки в массиве *his*. Это инструкция для будущей программы. В ней указаны не только номера атомов, которые должны войти в описание координаты типа χ . После трех атомов, описывающих первую плоскость, идет группа чисел 0 1 2. Первое число в этой группе ссылается на номер готового единичного вектора оси поворота этой плоскости, когда координата χ положительна. Если здесь стоит ноль, то программа построит временный единичный вектор оси, протянув его от атома номер 1 к атому номер 2. Эти номера и стоят вторым и третьим в данной группе. Следующие 6 чисел в строке относятся ко второй плоскости и имеют тот же смысл. При указании номеров атомов плоскости их можно обходить в любом порядке, только упоминая центральный атом вторым. А вот направление оси вращения прямо связано с порядком указания номеров двух атомов. Обратим внимание на группу номеров 0 2 1 в конце описания второй плоскости. Та же связь, вокруг которой вращаются обе плоскости, описана здесь иначе, чем для первой плоскости. Это и обеспечивает внутреннее вращение плоскостей в противоположных направлениях.

Напишем скрипт *Bh.m*, который сформирует субматрицу *B_h* полной матрицы **B**.

```

% % B submatrix for hi-coordinates

nHis = size(His, 1);
% Vector-Matrix Ba is represented with three scalar matrices
Bhx = zeros(nHis, nAtoms);
Bhy = zeros(nHis, nAtoms);
Bhz = zeros(nHis, nAtoms);
for ih = 1:nHis
    % Plane 1
    Atom1 = His(ih, 1);
    Atom2 = His(ih, 2);
    Atom3 = His(ih, 3);
    [e1, s1] = atom_pair_vector(Atom2, Atom1, AtomCoords);
    [e2, s2] = atom_pair_vector(Atom2, Atom3, AtomCoords);

```

```

s1 = 1.09/s1;
s2 = 1.09/s2;
    % Get eaux as the turn axis
i = His(ih, 4);
if i > 0 % axis is a ready unit vector
    eaux = handles.Vectors(i,:);
else
    eaux = atom_pair_vector(His(ih, 5), His(ih, 6), AtomCoords);
end % if > 0
    % Get normal
normal = cross(e1, e2);
normal = normal/norm(normal);
ete1 = eaux*e1';
ete2 = eaux*e2';
e1e2 = e1*e2';
A = sqrt(ete1^2 + ete2^2 - 2*e1e2*ete1*ete2);
b1 = s1*normal*ete2/A;
b3 = -s2*normal*ete1/A;
b2 = -(b1 + b3);
    % Plane 2
Atom4 = His(ih, 1+6);
Atom5 = His(ih, 2+6);
Atom6 = His(ih, 3+6);
[e1, s1] = atom_pair_vector(Atom5, Atom4, AtomCoords);
[e2, s2] = atom_pair_vector(Atom5, Atom6, AtomCoords);
s1 = 1.09/s1;
s2 = 1.09/s2;
    % Get eaux as the turn axis
i = His(ih, 4+6);
if i > 0 % axis is a ready unit vector
    eaux = handles.Vectors(i,:);
else
    eaux = atom_pair_vector(His(ih, 5+6), His(ih, 6+6), AtomCoords);
end % if > 0
    % Get normal
normal = cross(e1, e2);
normal = normal/norm(normal);
ete1 = eaux*e1';
ete2 = eaux*e2';
e1e2 = e1*e2';
A = sqrt(ete1^2 + ete2^2 - 2*e1e2*ete1*ete2);
b4 = s1*normal*ete2/A;
b6 = -s2*normal*ete1/A;
b5 = -(b4 + b6);
Bhx(ih, Atom1) = b1(1);
Bhx(ih, Atom2) = b2(1);
Bhx(ih, Atom3) = b3(1);
Bhx(ih, Atom4) = b4(1);
Bhx(ih, Atom5) = b5(1);
Bhx(ih, Atom6) = b6(1);
Bhy(ih, Atom1) = b1(2);
Bhy(ih, Atom2) = b2(2);
Bhy(ih, Atom3) = b3(2);
Bhy(ih, Atom4) = b4(2);
Bhy(ih, Atom5) = b5(2);
Bhy(ih, Atom6) = b6(2);
Bhz(ih, Atom1) = b1(3);
Bhz(ih, Atom2) = b2(3);
Bhz(ih, Atom3) = b3(3);
Bhz(ih, Atom4) = b4(3);
Bhz(ih, Atom5) = b5(3);
Bhz(ih, Atom6) = b6(3);
end % His

```

Внимание!

Строки вида `Bhx(ih, Atom1) = b1(1)` дадут для этой конкретной модели правильный результат. Однако в общем случае эти строки должны иметь вид

$$B_{hx}(ih, Atom1) = B_{hx}(ih, Atom1) + b1(1);$$

Дело в том, что в сложной модели типа бензола две вращающиеся друг относительно друга плоскости могут содержать атомы с одинаковыми номерами. В матрице **B** два элемента $b1(1)$, относящиеся к одному и тому же атому, должны быть сложены, а не заменять друг друга, как это было бы в нашем простом скрипте. В модели типа этилена такой опасности нет.

Выполним этот скрипт.

Теперь у нас в рабочей области MatLab есть все субматрицы полной матрицы **B**. Соединим эти субматрицы, чтобы получить скалярные компоненты V_x, V_y, V_z полной матрицы **B**. Для этого напишем, сохраним в файле `Bmatrix.m` и выполним скрипт

```
% Matrix B
Vx = [Bqx; Bax; Brx; Bhx];
Vy = [Bqy; Bay; Bry; Bhy];
Vz = [Bqz; Baz; Brz; Bhz];
clear Bqx Bax Brx Bhx
clear Bqy Bay Bry Bhy
clear Bqz Baz Brz Bhz
```

Теперь уже можно сформировать кинематическую часть колебательного гамильтониана модели. Это матрица кинематических коэффициентов T_p , с помощью которой можно в процессе колебаний рассчитывать кинетическую энергию модели. Кинетическая энергия выражается формулой

$$E_k = \frac{1}{2} p^T T_p p.$$

Здесь p есть матрица-столбец значений импульсов, сопряженных обобщенным колебательным координатам. Штрих означает операцию транспонирования, как это принято в языке MatLab. Как вычисляются эти импульсы, будет сказано позже. Сейчас важно отметить, что матрица T_p несет полную информацию о кинематической части гамильтониана, в частности, о взаимных влияниях колебательных координат друг на друга при деформациях молекулы и о зависимости колебательных движений модели от масс атомов и от структурных особенностей равновесного состояния молекулы. Матрица T_p вычисляется по формуле

$$T_p = BEB'.$$

Здесь E – диагональная матрица обратных масс атомов, умноженных на 1.088. Этот множитель возник в теории колебаний молекул из стремления частично учесть ангармонизм колебаний, присущий валентным связям и углам, куда входят атомы водорода. Можно обойтись без этого множителя, но тогда надо перемасштабировать силовые постоянные моделей органических молекул в уже накопленных банках данных таких готовых моделей.

Напишем, сохраним в файле `MatTp.m` и выполним скрипт

```
% Matrix Tp
E = 1.088./AtomMasses;
E = diag(E);
Tp = Vx*E*Vx' + Vy*E*Vy' + Vz*E*Vz';
```

Рассматривая полученную матрицу T_p , мы заметим, что она имеет блочную, квазидиагональную структуру. В строке 14 и в столбце 14 найдется единственный элемент, не равный нулю. Элемент

$\text{Tr}(14, 14) = 1.3784$, относящийся к единственной координате типа χ , не взаимодействует ни с какими другими колебательными координатами. Это значит, что мы могли бы, приписав этой координате определенную силовую постоянную, скажем $u = 0.755$, сразу вычислить частоту единственного колебания типа внутреннего вращения в модели. Для этого достаточно выполнить в командном окне MatLab команду

```
freq_hi = 1000*sqrt(0.755*Tr(14,14))
```

Получим ответ $\text{freq_hi} = 1.0202\text{e}+003$. Это как раз то значение частоты 1020 см^{-1} , которое мы получим, решив далее задачу о всех колебаниях модели.

Вот как просто иногда работает теория частот в колебательных спектрах молекул. Вот насколько полезно проникнуть вовнутрь теории и обжиться там до полного комфорта. Представим себя такими жильцами, и пусть нас интересует реальный этилен, его колебательный спектр. Но пусть в некоем прикладном исследовании нас в спектре волнует только проявление внутренних вращений в этих молекулах вещества. Как найти эти проявления в спектре? Ведь спектральная картина в приборе это просто сосульки, на которых ничего не написано. Зная же эту особенность внутреннего вращения в модели молекулы этилена, мы можем опустить в расчетах все описанные выше шаги, кроме работы с координатой типа χ . Кинематический коэффициент 0.755 для этой координаты можно получить, выполняя в окне MatLab лишь те строчки скриптов, которые относятся к данной колебательной координате. Найдя в литературе силовую постоянную для этой координаты, мы сразу вычислим частоту этого колебания. И можно двигаться дальше в основном исследовании, как-то воздействуя на вещество и следя за поведением сосульки в районе 1020 см^{-1} . А если еще рассчитать интенсивности в ИК спектре этой модели, то можно узнать, что данное колебание никак не проявляется в этом спектре. Надо искать другой спектроскопический метод наблюдения за веществом. К счастью, таких методов теперь можно найти множество. Были бы деньги на покупку соответствующих спектральных приборов. А теоретический анализ, как мы видим, ничего не стоит. Кроме жизни, загубленной в подземельях теории колебаний молекул.

Теперь надо сформировать матрицу U_q силовых постоянных модели. Эта матрица представляет вторую часть гамильтониана модели. С помощью матрицы U_q можно записать выражение для потенциальной энергии E модели в ее деформированном состоянии.

$$E = \frac{1}{2} q' U_q q.$$

Здесь q есть матрица-столбец значений всех колебательных координат в данной деформации модели.

Матрица U_q симметрична, поэтому можно приготовить и ввести только ее верхний треугольник, а программа сформирует полный вид квадратной матрицы. Кроме того можно учесть, что обычно вне главной диагонали матрица содержит много нулевых элементов. Поэтому удобно ввести матрицу в записи, принятой для редких матриц. Найдём в литературе значения силовых постоянных молекулы этилена, напишем и сохраним в файле `Uqdata.m` скрипт для ввода этих данных в программу.

```
% Matrix Uq sparse
UqSparse = [ ...
    1    1  14.2; ...
    1    2   0.1; ...
    1    3   0.1; ...
```

```

1 4 0.1; ...
1 5 0.1; ...
1 6 0.42; ...
1 7 0.42; ...
1 8 0.42; ...
1 9 0.42; ...
2 2 8.55; ...
2 3 0.03; ...
2 6 0.2; ...
2 10 0.12; ...
3 3 8.55; ...
3 7 0.2; ...
3 10 0.12; ...
4 4 8.55; ...
4 5 0.03; ...
4 8 0.2; ...
4 11 0.12; ...
5 5 8.55; ...
5 9 0.2; ...
5 11 0.12; ...
6 6 0.77; ...
6 8 -0.025; ...
6 9 0.095; ...
7 7 0.77; ...
7 8 0.095; ...
7 9 -0.025; ...
8 8 0.77; ...
9 9 0.77; ...
10 10 0.57; ...
11 11 0.57; ...
12 12 0.35; ...
12 13 0.047; ...
13 13 0.35; ...
14 14 0.755; ];

```

Выполним этот скрипт. В рабочей области появится массив UqSparse.

Теперь надо развернуть этот массив в полную матрицу U_q . Для этого напишем, сохраним в файле Ufull.m и выполним следующий скрипт.

```

% Uq from UqSparse

n = size(Tp, 1);
Uq = zeros(n);
mu = size(UqSparse, 1);
for i = 1:mu
    iu = UqSparse(i,1);
    ju = UqSparse(i,2);
    u = UqSparse(i,3);
    Uq(iu, ju) = u;
    Uq(ju, iu) = u;
end % i
clear UqSparse iu ju u mu

```

В рабочей области возникнет матрица U_q . Требуется, чтобы эта матрица была положительно определенной. Это обеспечивается, если на диагонали матрицы стоят положительные числа, а недиагональные элементы по абсолютной величине не превосходят диагональные.

Теперь вся механическая часть модели представлена в компьютере гамильтонианом с матрицами T_p и U_q . Можно выполнить расчет частот и форм колебаний модели. В теории колебаний фигурирует еще и матрица $D = T_p * U_q$. Известно, что частоты колебаний модели находятся через собственные числа матрицы D . Однако алгоритм вычисления тогда получается неудобным, поскольку матрица D не является симметричной. Значительно более эффективным оказывается

алгоритм последовательной диагонализации симметричных матриц T_p и U_q . Этот алгоритм полностью описан в следующем скрипте, который мы сохраним в файле `vibrations.m`.

```
% calculation of frequencies and forms of vibrations

% DiagTp
[Ltau, d] = eig(Tp);
d = diag(d);
% remove zero eigen values
indzero = [];
for i = 1:n
    if abs(d(i)) < 0.001
        indzero = [indzero, i];
    end % if
end % i
d(indzero) = [];
m = length(d); % number of normal vibrations
Ltau(:,indzero) = [];
clear indzero
tau = sqrt(d);
for i = 1:m
    Ltau(:,i) = Ltau(:,i)*tau(i);
end % i

% diaguq
Utau = Ltau'*Uq*Ltau;
[Lu, d] = eig(Utau);
freqs = sqrt(diag(d))*1000;
Lq = Ltau*Lu;
for i = 1:m
    Ltau2(:,i) = Ltau(:,i)/(tau(i)^2);
end % i
Lp = Ltau2*Lu;
```

После выполнения этого скрипта мы получаем в рабочей области частоты и формы колебаний модели. Этими данными можно сразу воспользоваться в интересующих пользователя прикладных работах. Тем и сильна система вычислений MatLab, что пользователь может воспользоваться чужим алгоритмом с открытым кодом в своих целях, не заботясь о способах добычи любых промежуточных результатов работы чужой программы, если эти результаты полезны в его собственном компьютерном эксперименте. Достаточно написать свою программу, которая будет запущена после окончания чужой программы и использовать любые переменные, остающиеся в рабочей области MatLab.

Комментарий `% remove zero eigen values` требует дополнительных пояснений. Молекулярная колебательная модель обычно содержит лишние, линейно зависимые колебательные координаты. Это удобно для учета симметрии модели. Кроме того, создание системы линейно независимых координат часто бывает очень трудным делом даже для опытного исследователя. В нашей простой модели этилена мы могли бы не вводить координаты типа α для валентных углов НСН. Тогда число колебательных координат получилось бы равным числу колебательных степеней свободы $3N - 6 = 12$. У нас же получилось 14 колебательных координат. Значит, в системе колебательных координат имеется две линейных зависимости. В этой простой модели линейные зависимости вскрываются легко. Можно убедиться расчетом, что для каждой тройки координат типа α , расположенных вокруг двух атомов С, выполняется равенство $\alpha_1 + \alpha_2 + \alpha_3 = 0$. Но это понятно и без расчетов. Как бы ни была деформирована модель в ее плоскости, сумма этих трех валентных углов равна 360° и постоянна. В более сложных молекулах бывают и более сложные линейные зависимости.

Наличие зависимостей приводит к тому, что среди собственных чисел матрицы T_p возникают нулевые значения. Они должны быть вычеркнуты вместе с соответствующими собственными

векторами. Этим и занимается часть алгоритма после данного комментария. В результате определяется число ненулевых собственных чисел, которое является числом нормальных колебаний модели.

После нахождения частот freqs нормальных колебаний алгоритм находит формы колебаний в виде матриц L_q и L_p . Матрица L_q есть матрица форм колебаний в естественных колебательных координатах. Столбец $L_q(:, i)$, соответствующий частоте $\text{freqs}(i)$, показывает, насколько велика каждая колебательная координата, если в модели возбуждено только данное нормальное колебание. Для того, чтобы получить формы колебаний в привычных единицах (в \AA для координат типа q и в радианах для угловых координат), надо умножить столбец $L_q(:, i)$ на амплитуду данного нормального колебания. Нулевая амплитуда (когда колебательное квантовое число $v = 0$) $4/\sqrt{\text{freqs}(i)}$ \AA . Для другого значения v нулевую амплитуду надо умножить на $\sqrt{v + 1}$. В этом абзаце формулы записаны в нотации MatLab, чтобы их можно было сразу использовать в расчетах.

Остается рассчитать абсолютные интенсивности поглощения в ИК спектре модели. Для этого надо ввести в модель ее электрооптические параметры. Находим в литературе значения дипольных моментов связей в единицах D и значения производных от дипольных моментов связей по естественным колебательным координатам в единицах $D/\text{\AA}$. Напишем, сохраним и выполним скрипт Eops.m.

```
% Miu and dmdqSparse
```

```
Miu = [ ...
    0.00000; ...
    0.71900; ...
    0.71900; ...
    0.71900; ...
    0.71900 ];

dmdqSparse = [ ...
    2 2 0.49300; ...
    2 3 0.04100; ...
    3 2 0.04100; ...
    2 6 0.48100; ...
    2 10 0.55100; ...
    3 3 0.49300; ...
    3 7 0.48100; ...
    3 10 0.55100; ...
    4 4 0.49300; ...
    4 5 0.04100; ...
    5 4 0.04100; ...
    4 8 0.48100; ...
    4 11 0.55100; ...
    5 5 0.49300; ...
    5 9 0.48100; ...
    5 11 0.55100 ];
```

Интенсивность ИК поглощения на частоте данного нормального колебания определяется квадратом скорости изменения полного дипольного момента модели. Дипольный момент модели складывается из дипольных моментов отдельных связей. Дипольный момент связи есть μe , μ – электрооптический параметр связи, а e – ее единичный направляющий вектор. Есть две причины изменения со временем суммы таких произведений – полного дипольного момента модели.

1. В процессе нормального колебания связи могут периодически поворачиваться в пространстве. При этом скорость изменения дипольного момента модели определяется параметрами μ модели и скоростями поворотов связей.

2. В процессе нормального колебания величина дипольного момента каждой связи может измениться в зависимости от деформации модели. Для расчета вклада таких изменений в скорость изменения дипольного момента модели и вводятся параметры вида $\partial\mu/\partial q, \partial\mu/\partial\alpha$ и аналогичные производные по колебательным координатам остальных типов.

После выполнения скрипта Eops.m в рабочей области возникают два массива Miu и dmdqSparse, содержащие описанные электрооптические параметры модели.

Для вычисления интенсивностей полос поглощения, соответствующих нормальным колебаниям модели, в рабочей области теперь есть все необходимые данные. Напишем и выполним скрипт Intir.m.

```
% Intensities of IR absorbtion

dmdq = zeros(nBonds, n);
mdmdq = size(dmdqSparse, 1);
for i = 1:mdmdq
    idmdq = dmdqSparse(i,1);
    jdmdq = dmdqSparse(i,2);
    r = dmdqSparse(i,3);
    dmdq(idmdq, jdmdq) = r;
end % i

Iabs = zeros(m,1);
sigma = 1.0./s_Bonds;
for i = 1:m
    % atomic shifts for i-th normal vibration
    rx=E*Bx'*Lp(:,i);
    ry=E*By'*Lp(:,i);
    rz=E*Bz'*Lp(:,i);
    Ix = 0;
    Iy = 0;
    Iz = 0;
    for j = 1:nBonds
        Ix = Ix + Miu(j)*sigma(j)*(rx(Bonds(j,2)) - rx(Bonds(j,1)) - ...
            e_Bonds(j,1)*Lq(j,i));
        Iy = Iy + Miu(j)*sigma(j)*(ry(Bonds(j,2)) - ry(Bonds(j,1)) - ...
            e_Bonds(j,2)*Lq(j,i));
        Iz = Iz + Miu(j)*sigma(j)*(rz(Bonds(j,2)) - rz(Bonds(j,1)) - ...
            e_Bonds(j,3)*Lq(j,i));
    end % j
    Ix = Ix + e_Bonds(:,1) '*dmdq*Lq(:,i);
    Iy = Iy + e_Bonds(:,2) '*dmdq*Lq(:,i);
    Iz = Iz + e_Bonds(:,3) '*dmdq*Lq(:,i);
    Iabs(i) = 20*(Ix^2 + Iy^2 + Iz^2);
end % i
```

Программа находит для каждого нормального колебания не только абсолютную интенсивность Iabs в единицах 10^{-8} см²/с на одну молекулу, но и вклады в интенсивность от составляющих Ix, Iy, Iz скорости изменения полного дипольного момента модели по осям координат. Эта информация нужна при анализе поляризаций в ИК спектре вещества. Множитель 20 служит переводным коэффициентом для пересчета размерностей.

Для наглядного представления полученных результатов напишем и выполним скрипт IRcurve.m.

```
% IR spectral curve

x = 200:5:3400; % frequencies points
nx = length(x);
x = x';
sig_Gauss = 10.0; % parameter of Gausse function
```

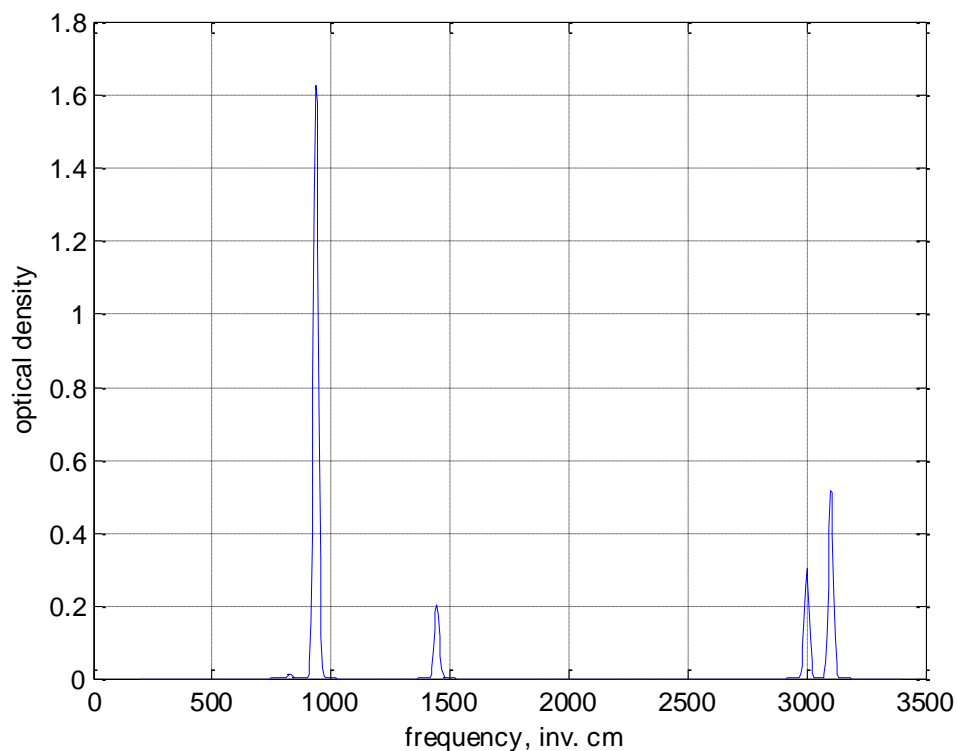


```

Dy=zeros (nx,m);
for j=1:m
    Dy(:,j)=(Iabs(j)/(2.5*sig_Gauss))* ...
        exp(-(0.5/sig_Gauss^2)*(x-freqs(j)).^2);
end %j
y=(sum(Dy'))';
plot(x,y)
grid
xlabel('frequency, inv. cm')
ylabel('optical density')

```

Получим следующий вид спектрального распределения поглощения молекулой этилена ИК излучения.



6.3. Выводы

Просмотрим список частот колебаний и абсолютных интенсивностей в ИК спектре модели этилена, полученный в расчете.

```

FreqInt = [freqs Iabs]
FreqInt =
    1.0e+003 *
    1.6210    0.0000
    3.0174    0.0000
    1.3398    0.0000
    3.0009    0.0075
    3.0990    0.0000
    3.1022    0.0133
    0.8246    0.0003
    1.4443    0.0051
    1.2467    0.0000
    0.9571    0.0000
    0.9369    0.0414
    1.0202     0

```

Можно убедиться, что получено совпадение с результатами расчета, выполненного на машине ЕС ЭВМ и опубликованного нами в книге: Л.А.Грибов, В.А. Дементьев, А.Т.Тодоровский. Интерпретированные колебательные спектры алканов, алкенов и производных бензола. М. Наука, 1986.

Программы на Фортране для ЕС ЭВМ я перекодировал и переорганизовывал, опираясь на свои же тексты на Фортране для машины Минск 32. Это заняло целый год упорной работы.

Приведенные в этой главе скрипты MatLab я сочинил в процессе написания текста этой главы. Вся работа заняла один рабочий день, включая отладку программ на материале известного мне тестового примера модели этилена.

Комментарии излишни.

Но имеет смысл обсудить различия между приведенными здесь скриптами и теми m-файлами программ, которые находятся в папке LevInfinite\Programs. Работе с этими программами были посвящены предыдущие главы книги. А в тексты этих программ мы не заглядывали. Если же заглянуть, то мы обнаружим, что даже на таком компактном языке программирования, который составляет язык операторов и функций MatLab, тексты программ расплзаются на тысячи строк. Почему в этой главе удалось изложить всё в одной сравнительно короткой программе? Ведь все скрипты можно соединить в один длинный скрипт и получится одна программа. Почему она получается вполне обозримой?

Потому что в ней отсутствует графический интерфейс пользователя. И нет обмена между программой и внешними файлами с данными о модели. Нет операций автоматического распознавания характерных элементов матриц силовых и электрооптических параметров модели. Нет поиска группы симметрии, перехода к координатам симметрии, получения решения прямой задачи с сортировкой колебаний по типам симметрии. Нет расчета смещений атомов из положений равновесия при колебаниях. Наконец, нет вычисления интенсивностей в ИК спектре модели. Я здесь пренебрег всеми удобствами, которые создавал для пользователя системы LevML средствами программирования. Зато я предъявил читателю весь основной алгоритм постановки и решения прямой задачи теории колебаний молекул. Весь этот алгоритм, включающий действия, как пользователя, так и компьютера, описан на одном формальном языке настолько компактно, что читатель может осмотреть его в один присест.

Возможно, проделанная здесь работа окажется полезной не только в учебных целях для начинающего пользователя, но и для опытного исследователя, если тому не нужны все сложности молекулярного моделирования, да еще и визуального. Если пользователь не боится вводить данные в программу путем редактирования m-файлов, если ему нужны только частоты и формы колебаний малых моделей, то приведенная в этой главе программа может оказаться вполне пригодным инструментом для исследовательской работы.

6.4. Рецепт последовательной диагонализации двух симметричных матриц с автоматическим исключением лишних координат.

Этот вычислительный рецепт придумал Л.А. Грибов, но не опубликовал доказательства его правильности. Не догадался, что вывод формул данного рецепта достоин отдельной статьи. Считаю это несправедливым. И хочу исправить оплошность автора этого замечательного рецепта, выручающего нас много лет и во всех вариантах наших колебательных программ.

В данной книге мы уже сталкивались с этим рецептом. В главе 5.2, в разделе «Что удалось найти в алмазной пыли» мы убедились, что рецепт Грибова прекрасно справляется с колебательным уравнением порядка 3736, исключая из этого гигантского количества колебательных координат все линейные зависимости так, что автоматически осуществляется переход к 1891 независимым координатам. Это и позволяет найти для такой громоздкой модели алмаза, состоящей из 617 атомов, независимые нормальные колебания, несмотря на наличие 1845 зависимостей между исходными колебательными координатами. Почти половина исходных колебательных координат оказалась лишней! Однако без этих лишних координат пользователю было бы чрезвычайно трудно, почти невозможно технически составить колебательный гамильтониан крупной модели алмаза.

В главе 6.2 мы еще раз встретились с рецептом Грибова. Там он строго изложен на языке MatLab в скрипте `vibrations.m` после комментария `% remove zero eigen values`. В скрипте, реализующем рецепт Грибова, всего 15 строк.

Теперь я собираюсь привести все рассуждения, которые поясняют внутреннее устройство этого рецепта. Эти рассуждения были изложены Грибовым в лекции, которую он прочел своим молодым сотрудникам в начале славных дел, развивавшихся далее под вывеской «Научная школа Грибова». Боюсь, что в моем пересказе, с теоретическим вступлением и с выкладками, изложение займет страниц 15, sorry.

Оттолкнемся от теории одномерного гармонического осциллятора. Материальная точка массы m прикреплена пружинкой жесткости k к стенке, имеющей бесконечно большую массу. Уравнение движения материальной точки в предположении, что сила $f(x)$, развиваемая пружиной, подчиняется закону Гука при любых деформациях пружины x , запишем в соответствии со вторым законом Ньютона

$$m \frac{d^2 x}{dt^2} = -kx.$$

Или

$$\frac{d^2 x}{dt^2} = -\frac{k}{m} x.$$

Известно решение этого уравнения

$$x(t) = A \sin(\omega t + \phi).$$

Амплитуда A и начальная фаза ϕ гармонического колебания зависят от начальных условий. Квадрат круговой частоты этого колебания

$$\omega^2 = \frac{k}{m}.$$

Возникли определенные признаки, по которым можно проследить причинно-следственные связи.

Если физическая ситуация описывается вот таким дифференциальным уравнением («физическая ситуация» в переводе на русский означает «мы знаем, кто кого и как тянет или толкает»); в данном случае пружинка тянет или толкает материальную точку так, что выполняется закон Гука), то верно предсказание: материальная точка будет колебаться вдоль координаты x гармонически с частотой ω . Размах колебаний будет зависеть от начального растяжения пружинки или от скорости толчка, но частота зависит только от k и m .

Эти же признаки мы можем увидеть, рассматривая энергетические свойства этой простой физической системы. Кинетическая энергия системы E_k связана только с импульсом $p = mv$ материальной точки

$$E_k = \frac{p^2}{2m} = \frac{1}{2} p \frac{1}{m} p.$$

Правое выражение является простейшей квадратичной формой относительно импульсов.

Коэффициент $\frac{1}{m}$ будем называть кинематическим коэффициентом этой формы.

Потенциальная энергия системы E_p связана только с деформацией пружины x

$$E_p = \frac{1}{2} kx^2 = \frac{1}{2} xkx.$$

Коэффициент k будем называть динамическим коэффициентом этой квадратичной формы.

Присмотревшись к двум последним выражениям, можем выделить признак возможного гармонического колебания в системе. Если полная механическая энергия системы выражается суммой вот таких выражений (квадратичных форм с коэффициентами $\frac{1}{m}$ и k), то система может существовать в состоянии гармонического колебательного движения с круговой частотой

$$\omega = \sqrt{\frac{k}{m}} = \sqrt{\frac{1}{m}k}.$$

Неважно, каковы конкретные значения координат и импульсов в системе, каковы конкретные значения кинетической и потенциальной энергий в данный момент. Важно, что система способна совершать гармонические колебания, а для предсказания частоты такого колебания достаточно знать конкретные величины коэффициентов, входящих в выражения для двух видов механической энергии. В сложной системе более удобным оказывается именно этот признак возможности гармонических колебаний.

Для системы, состоящей из нескольких материальных точек, соединенных несколькими пружинками, выражения для энергий принимают более обобщенный характер. Адекватной оказывается квадратичная форма. Адекватной также оказывается некая система обобщенных координат и сопряженных им импульсов. С записью квадратичной формы нет никаких принципиальных трудностей. Достаточно усвоить рецепты из учебника «Линейная алгебра и аналитическая геометрия» любого автора, для любого вуза.

С выбором и назначением адекватной системы обобщенных координат дело обстоит худо. Надо быть очень талантливым человеком, даже гением, чтобы придумать адекватную систему обобщенных координат для описания событий в такой сложной механической системе. Вспомним, что образы молекулы и атома – это достижение химиков. Физики в исследовании микромира идут за химиками, стараясь проектировать эти образы на физические образы. Своеволие они проявили лишь в догадке, что атом (Атом по-гречески = «персона, действующее лицо», а в первоначальной научной терминологии – «неделимый»), милый всем мыслителям неделимый атом оказался делимым на ядро и кучу электронов. Вспомнили. Значит, поймем, что колебаниями молекул одновременно интересуются и физики, и химики, люди, редко понимающие друг друга в силу различия языков, на которых они описывают свои объекты. А тут надо, чтобы физики и химики

одинаково ловко разбирались в колебательных молекулярных событиях. Для этого и нужна такая система обобщенных координат, которая была бы одинаково удобна и физикам, и химикам.

Удачную систему обобщенных колебательных координат придумал Михаил Александрович Ельяшевич. Было это давно, до Великой Отечественной. Эти координаты завоевали себе почетное место в работах всех теоретиков еще до того, как на свет появился Анатолий Иванович Павлючко. Потом он рос, учился, аспирантовал у Грибова, стал доктором наук, профессором. Состарился. И заявил мне на конференции по молекулярному моделированию 2011 года:

– А не считаете ли Вы, что со времени Ельяшевича в теории колебаний молекул ничего нового так и не было сделано?

Вот как высоко оценил роль Ельяшевича, изобретателя естественных колебательных координат, один из современных теоретиков.

Я вежливо ответил, что можно и так считать, если очень хочется. Хотя, опять же при желании, можно вспомнить такое событие, как открытие Грибовым ключевой роли колебаний молекул в химических реакциях. А мог бы ограничиться и упоминанием рецепта Грибова, без которого никто никогда не смог бы разобраться в колебательных состояниях крупных химических объектов. Так я оцениваю роль и М.А. Ельяшевича, и других теоретиков, чьи достижения сам М.А. оценивал со всей доброжелательностью.

К делу. Будем знакомиться с особенностями рецепта Грибова на примере несложной молекулярной модели. Модель напоминает молекулу CO_2 , но мы ограничимся описанием лишь тех ее движений, которые происходят вдоль линии О-С-О. Изломов этой линии рассматривать не будем.

Пронумеруем атомы так: О2-С1-О3. Введем систему естественных колебательных координат по Ельяшевичу:

q_1 – изменение равновесного расстояния между атомами 1 и 2.

q_2 – изменение равновесного расстояния между атомами 1 и 3.

q_3 – изменение равновесного расстояния между атомами 2 и 3.

Убедимся, что эта система координат удобна и физикам, и химикам.

Химику удобно видеть, насколько сильно изменилась валентная связь в молекуле. Значение колебательной координаты q_1 сразу скажет ему, что происходит с валентной связью О2-С1. Если $q_1 = 0$, то связь осталась такой, как в равновесном состоянии молекулы. Если $q_1 = -0.01$, то связь сжалась на 0.01 \AA . При этом химику совершенно не надо думать о декартовых координатах всех атомов молекулы. Координата q_3 , пожалуй, химику не нужна. Зачем ему следить за изменениями расстояния между крайними атомами молекулы?

Физику удобно видеть, насколько сильно изменяется протяженность всей молекулы в процессе колебаний, а вместе с этим изменением и объем молекулы. Следя за значением координаты q_3 , можно легко связать объем молекулы с температурой и, тем самым, теоретически найти коэффициент объемного расширения данного вещества. Конечно, необязательно вводить координату q_3 для слежения за распуханием молекулы в процессе колебаний. Можно учесть, что $q_3 = q_1 + q_2$. Но физик тоже человек, и ему лень проводить в уме даже такие простые вычисления, чтобы следить за распуханием молекулы. Удобнее сразу получать нужный для размышлений результат.

Мы уже знаем, что в сложной механической системе, состоящей из материальных точек и соединяющих их пружинок, кинетическая и потенциальная энергии выражаются более сложными квадратичными формами.

$$E_k = \frac{1}{2} p' T_p p, \quad E_p = \frac{1}{2} q' U_q q.$$

Здесь q – столбец естественных колебательных координат модели. p – столбец сопряженных им импульсов, штрих обозначает операцию транспонирования, T_p – матрица кинематических коэффициентов. U_q – матрица динамических коэффициентов. Эти матрицы мы уже умеем формировать средствами наших программ визуального моделирования.

По аналогии с одномерным гармоническим осциллятором мы можем видеть здесь признак возможности гармонических колебаний в системе. Также мы можем ожидать, что частоты этих колебаний определяются произведением матриц $T_p U_q = D$. Действительно, набор всех квадратов частот колебаний модели есть набор всех собственных чисел матрицы D .

Казалось бы, задача о колебаниях сложной системы уже решена. Осталось найти собственные числа матрицы D . Да, именно так решались сложные механические задачи еще в 19-м веке. И тогда же было замечено, что искать собственные числа несимметричной матрицы D очень сложно. И тогда же был изобретен рецепт последовательной диагонализации двух симметричных матриц, T_p и U_q . Этот рецепт приводил к тем же результатам, что и диагонализация матрицы D , но только в том случае, когда координаты q были независимыми. Вот тут и возникла специфическая проблема теории колебаний молекул.

Проблема состоит в том, что для физических и химических приложений удобно описывать модель молекулы в естественных колебательных координатах Ельяшевича, но очень трудно сформировать такое описание в независимых колебательных координатах. В нашем простом примере мы могли вводить, а могли и не вводить координату «распухания модели» q_3 . Могли бы сразу исключить эту координату и получить систему из двух независимых колебательных координат. Но в более сложных моделях очень трудно установить зависимость между координатами, описывающими все возможные деформации структурных элементов молекулы, привычных для химиков. Тем самым, эти зависимости трудно исключить на этапе формирования гамильтониана модели. Вплоть до 70-х годов 20-го века теоретики проявляли героизм, выполняя такую работу, что заметно сдерживало темп исследований.

Вот тут и появился рецепт Грибова. Физик Грибов на основе хорошего понимания линейной алгебры и аналитической геометрии догадался, что можно не исключать заранее зависимости между колебательными координатами, а сделать это автоматически в процессе последовательной диагонализации двух симметричных матриц T_p и U_q . Это позволило нам, сотрудникам Грибова, почувствовать себя свободными в выборе систем колебательных координат при описании сложных молекулярных моделей. Вводим в качестве колебательных координат все возможные деформации привычных структурных элементов молекулы – валентных связей, углов плоских и торсионных, изломов линейных участков и углов между связями и плоскостями. Данная работа столь рутинна, что ее можно поручить компьютерной программе. Если при этом возникнут линейные зависимости между координатами, то рецепт Грибова сам обнаружит и исключит эти зависимости.

Понаблюдаем теперь за всем процессом формирования колебательной модели молекулы и получением информации о частотах и формах колебаний этой модели. Воспользуемся для этого

моделью молекулы CO₂, которую отобразим в системе MatLab с помощью скриптов, описанных в разделе 2 этой главы. Отредактируем и выполним следующие скрипты.

Файл Atoms.m:

```
% AtomCoords

AtomCoords = [ 0.0000    0.0000    0.0000; ...
               -1.4000    0.0000    0.0000; ...
                1.4000    0.0000    0.0000];
```

Файл Masses.m:

```
% AtomMasses

AtomMasses = [12 16 16];
```

Файл Connect.m:

```
% Bonds

Bonds = [ 1 2 ; ...
          1 3 ; ...
          2 3 ];
```

В файле Vq.m ничего редактировать не надо, достаточно его просто выполнить.

В файле Vmatrix.m нет необходимости, поскольку у нас нет других типов колебательных координат. Но надо выполнить команды для переименования массивов:

```
>> Vx = Vqx;
>> Vy = Vqy;
>> Vz = Vqz;
```

Выполним скрипт MatTp.m.

```
Tp =
    0.1587 -0.0907  0.0680
   -0.0907  0.1587  0.0680
    0.0680  0.0680  0.1360
```

Мы можем прямо в рабочей области MatLab сформировать матрицу U_q модели, не прибегая к скриптам Uqdata.m и Ufull.m. Даем сразу команду

```
Uq = [ 22 -0.2 0; -0.2 22 0; 0 0 0.1];
Uq =

    22.0000   -0.2000     0
   -0.2000    22.0000     0
     0         0         0.1000
```

Колебательное уравнение модели сформировано. Теперь его надо решить. Будем давать необходимые команды прямо в командном окне MatLab. Прежде всего, найдем собственные значения и собственные векторы матрицы Tp.

```
[Ltau, d] = eig(Tp);
d = diag(d);
d =

   -0.0000
    0.2040
```

0.2493

Первое из собственных значений матрицы T_r оказалось нулевым. На языке линейной алгебры это означает, что пространство колебательных координат только притворяется трехмерным. На самом деле оно натянуто на два ортогональных вектора, оно двумерно. В данном примере мы заранее это знали, но приятно, что нам позволено не быть столь умными, поскольку вычислительная процедура диагонализация автоматически устанавливает этот факт по признаку равенства нулю одного из собственных значений.

Рассмотрим теперь соответствующую матрицу собственных векторов.

$L_{\tau} =$

$$\begin{pmatrix} 0.5774 & 0.4082 & 0.7071 \\ 0.5774 & 0.4082 & -0.7071 \\ -0.5774 & 0.8165 & 0 \end{pmatrix}$$

Обратим внимание на первый столбец. Это собственный вектор матрицы T_r , соответствующий ее первому собственному значению $d(1) = 0$. В этом векторе зашифрована такая информация:

$0.5774q_1 + 0.5774q_2 - 0.5774q_3 = 0$. Сокращая общий множитель, получим

$q_1 + q_2 - q_3 = 0$. Мы, будучи просто умными людьми, ранее без всякой программы догадались о том же самом, что $q_1 + q_2 = q_3$.

Вот мы дошли до первого пункта, где Грибов увидел важное обстоятельство. Оказывается, можно избежать мучений при поиске линейных зависимостей между естественными колебательными координатами. Не надо изучать труды теоретиков, которые посвятили целые главы монографий выводу рецептов поиска этих зависимостей в алгебраической форме (для разных типов молекулярных структур – разветвленных, циклических – рецепты получались разными). Можно ввести любые удобные исследователю колебательные координаты, затем построить соответствующую матрицу T_r , диагонализировать ее, найти номера нулевых собственных значений матрицы, посмотреть соответствующие им собственные векторы. Всё. Все линейные зависимости между колебательными координатами данной модели автоматически выявлены и описаны. Остается употребить эти зависимости, если это нужно в каком-то специальном исследовании. Или тут же забыть о них, как о страшном сне.

Идем дальше. Операция поиска собственных значений d и собственных векторов L_{τ} матрицы T_r на языке линейной алгебры означает, что мы перешли к новой системе обобщенных импульсов i , тем самым, привели квадратичную форму кинетической энергии модели к каноническому виду:

$$E_k = \frac{1}{2}(d_1 p_1^2 + d_2 p_2^2 + d_3 p_3^2).$$

Мы знаем, что такая форма описывает трехмерный эллипсоид, главные оси которого ориентированы вдоль осей p_1, p_2, p_3 . Но есть одна странность. Из-за того, что $d_1 = 0$, у нас получается несколько иное выражение для кинетической энергии, а именно

$$E_k = \frac{1}{2}(d_2 p_2^2 + d_3 p_3^2).$$

Но это уже не трехмерный эллипсоид, а нормальный двумерный эллипс! Становится ясно, что мы находимся не в трехмерном линейном пространстве молекулярной модели, а в двумерном.

Мы еще знаем из линейной алгебры, что переход от импульсов, сопряженных исходным колебательным координатам q , к нынешним импульсам p_1, p_2, p_3 совершается с помощью матрицы L_{tau} . Она-то и переводит казавшееся трехмерным пространство исходных импульсов в двумерное пространство. Мы также знаем из теории колебаний молекул, что та же матрица L_{tau} используется для перехода к новым колебательным координатам, сопряженным новым обобщенным импульсам. При этом исходная матрица силовых постоянных преобразуется в новую матрицу

$$U_{\text{tau}} = L'_{\text{tau}} U_q L_{\text{tau}}.$$

Тут мы дошли до второго пункта, где Грибов увидел еще одно важное обстоятельство. Оказывается, можно с помощью данного преобразования перейти не только к канонической квадратичной форме для потенциальной энергии, но и перейти из исходного линейного пространства q , казавшегося трехмерным, в двумерное пространство новых колебательных координат. Для этого надо удалить из матрицы L_{tau} первый столбец, собственный вектор, отвечающий нулевому собственному значению матрицы T_r .

В командном окне MatLab выполним команды, которые уберут из массивов ненужные элементы,

```
d(1) = [];
Ltau(:,1) = [];
d =
    0.2040
    0.2493
Ltau =

    0.4082    0.7071
    0.4082   -0.7071
    0.8165     0
```

После преобразования с такой матрицей L_{tau} новая матрица силовых постоянных U_{tau} станет двумерной, а две новые колебательные координаты станут независимыми. Но введем еще одну рационализацию, которая окажется полезной в дальнейшем. Мы хотели бы, чтобы собственные значения матрицы U_{tau} давали нам сразу квадраты частот колебаний. Для этого, по аналогии с одномерным осциллятором, новая матрица U_{tau} должна быть нагружена собственными значениями матрицы T_r .

Грибов предложил с этой целью преобразовать матрицу L_{tau} , умножив ее на диагональную матрицу $\text{tau} = \text{sqrt}(d)$.

Приготовим массив tau в рабочей области MatLab командой

```
tau = sqrt(d);
```

Затем умножим каждый столбец матрицы L_{tau} на соответствующий элемент массива tau :

```
Ltau(:,1) = Ltau(:,1)*tau(1);
Ltau(:,2) = Ltau(:,2)*tau(2);
```

И совершим преобразование

```
Utau = Ltau*Uq*Ltau;
```

```
Utau =
```

1.4960 0.0000
0.0000 5.5352

Нам повезло. Новая матрица силовых постоянных получилась не только двумерной и нагруженной собственными значениями матрицы Γ . Она получилась еще и диагональной. Тем самым, мы просто видим квадраты частот двух нормальных колебаний модели. Для того, чтобы получить частоты в привычных единицах см^{-1} , надо извлечь корень и умножить результат на 1000 см^{-1} . Получаем частоты 1223.1 и 2352.7.

Теперь получим формы колебаний в естественных колебательных координатах и в сопряженных им импульсах, Lq и Lp .

Представим себе, что нам не так повезло, и матрица U_{tau} получилась недиагональной. Тогда нам пришлось бы вычислить ее собственные значения и собственные векторы с помощью оператора

$$[Lu, d] = \text{eig}(U_{\text{tau}});$$

Дальше нам опять придется обратиться к умению Грибова понимать линейную алгебру и аналитическую геометрию.

Произвольную симметричную матрицу U_{tau} мы превращаем в диагональную с помощью матрицы Lu . Геометрически это означает поворот эллипса (в общем случае – многомерного эллипсоида) к его главным осям. Это есть переход к новым, нормальным колебательным координатам. Но мы уже один раз переходили к новым колебательным координатам с помощью матрицы L_{tau} . Полное преобразование пространства от конечных нормальных координат к исходным колебательным координатам дается произведением матриц:

$$Lq = L_{\text{tau}} * Lu;$$

$$Lq = \begin{pmatrix} 0.1844 & 0.3531 \\ 0.1844 & -0.3531 \\ 0.3688 & 0.0000 \end{pmatrix}$$

Как известно из теории колебаний молекул, эту информацию надо интерпретировать так.

Если значение нормальной координаты $Q_1 = 1$, а $Q_2 = 0$, то $q_1 = 0.1844$, $q_2 = 0.1844$, $q_3 = 0.3688$. Это симметричное колебание, в котором обе связи CO растянулись одинаково. При этом координата «распухания» модели тоже положительна. Да, модель распухает. Мы могли бы понять это, суммируя $q_1 + q_2$. Но не всегда хочется суммировать, приятнее видеть готовый результат. А для этого нам и нужна колебательная координата q_3 .

Если значение нормальной координаты $Q_1 = 0$, а $Q_2 = 1$, то $q_1 = 0.3531$, $q_2 = -0.3531$, $q_3 = 0$. Это антисимметричное колебание, в котором одна связь CO растянулась, а другая сжалась на столько же. При этом координата «распухания» модели равна нулю – размер молекулы в таком колебании не меняется. Видеть здесь ноль особенно приятно. Думать не надо.

Теперь подумаем об импульсах. Тоже с помощью Грибова. Мы поворачивали пространство колебательных координат дважды. А пространство импульсов – лишь один раз. Когда мы повернули пространство координат второй раз, эллипс в пространстве импульсов тоже как-то повернулся. Хвост вытащил – нос увяз. Не удастся одновременно привести обе квадратичные формы к каноническому виду. Вот тут Грибов и предлагает трюк – превратить эллипс импульсов в круг (в многомерном случае – эллипсоид в многомерную сферу). Тогда при втором повороте

пространства координат окружность импульсов повернется вокруг центра, но останется окружностью (многомерной сферой). И обе квадратичные формы примут канонический вид.

Тогда можно будет и от нормальных импульсов переходить к исходным импульсам, без чего нельзя, например, рассчитать интенсивности поглощения в ИК спектре модели.

В вычислительном алгоритме эта идея реализуется так.

Находим такое преобразование импульсов $L_{\tau 2}$, которое не только поворачивало бы эллипсоид импульсов к его главным осям, но и превращало его в сферу единичного радиуса, изменяя масштабы по осям импульсов.

```
Ltau2 = zeros(n,m);  
for i = 1:m  
    Ltau2(:,i) = Ltau(:,i)/(tau(i)^2);  
end % i  
Lp = Ltau2*Lu;
```

Так написано в нашем скрипте `vibrations.m` в разделе 6.2. Но теперь понятно, почему там так написано.

Итак, я опубликовал здесь всю анатомию рецепта Грибова, специально отметив, что было сделано теоретиками до Грибова, и что именно придумал Л.А. Грибов.