

# NATRC

---

Non-Adiabatic Transfer Rates Calculations

## 1. Описание

Программа выполняет расчет неадиабатических скоростей перехода между двумя состояниями в молекулах. Вычисления проводятся на основе модели Плотникова-Биксона-Джортнера. В качестве входных данных используются выходные данные GAMESS для расчетов типа ENERGY, NACME и HESS.

## 2. Системные требования

Linux OS 32/64-bit

gfortran  $\geq 4.8$

Intel MKL  $\geq 10.2.2$

## 3. Запуск расчета

### а. Компиляция и запуск

Компиляция программы производится следующим образом

```
cd NATRC/  
gfortran NATRC.f90 -Wl,--start-group  
mkl_lpath/libmkl_gf.a mkl_lpath/libmkl_sequential.a  
mkl_lpath/libmkl_core.a -Wl,--end-group -lpthread -lm -  
ldl -o NATRC
```

*Примеры, как может выглядеть путь к библиотеке mkl\_lpath:*

*/opt/intel/composerxe/mkl/10.2.2.025/lib/32/*

*/opt/intel/mkl/lib/intel64/*

*Примечание:*

*Файл libmkl\_gf.a может быть записан как libmkl\_gf\_lp64.a для x64-систем*

После компиляции в папке формируется файл NATRC, который и является программой для расчета скоростей неадиабатического перехода. Чтобы запустить программу, введите:

```
./NATRC
```

## в. Входные данные

Папка NATRC/ кроме основной программы должна содержать файлы с входными данными. Среди которых **input**-файл с входными параметрами, log-файл пасте-расчета молекулы *с начальной геометрией*, log-файл пасте-расчета *с конечной геометрией*, а также dat-файл, содержащий гессиан молекулы в одном из состояний. Название первого файла всегда остается неизменным, другие файлы могут иметь произвольные названия, которые записываются в input. Данные содержащиеся в них получаются в результате расчета на GAMESS.

## с. Описание файла input

Файл input содержит описание ключевых параметров, которые нужны для расчета. Все параметры описываются в формате:

```
parameter: value
```

На одной строке должно содержаться описание только одного параметра. Комментарий указывается с помощью восклицательного знака! Допускаются строки, содержащие только комментарии.

```
parameter: value !comment
```

В качестве параметра можно ставить лишь ограниченный набор ключевых слов. Через двоеточие указывают значение параметра. Оно может быть числом, либо текстовой строкой без кавычек. Большинство параметров - обязательные. Если их не указать, то программа выдаст ошибку.

## Ключевые слова

<b>number_states_nacme</b>	Число состояний, которые использовались при NACME расчете. Может быть только $\geq 2$ .
<b>initial_state</b>	Номер состояния, в котором молекула находится в начальный момент времени. Может быть только $\leq \text{number\_states\_nacme}$
<b>final_state</b>	Номер состояния, в которое молекула приходит. $< \text{number\_states\_nacme}$ и $< \text{initial\_state}$ .
<b>Eif</b>	Разность энергий между состояниями в равновесных геометриях. Может быть только положительной.
<b>Temperature</b>	Температура системы в Кельвинах. Необязательный параметр. Если не указана, то она определяется автоматически как 298.15
<b>initial_nacme_file</b>	Имя log-файла, полученного в результате насме расчета молекулы с начальной геометрией Должен содержать координаты атомов в молекуле, насме коэффициенты для перехода между $\text{number\_states\_nacme}$ состояниями.
<b>final_nacme_file</b>	Имя log-файла, полученного в результате насме расчета молекулы с конечной геометрией Должен содержать координаты атомов в молекуле.
<b>hess_file</b>	Имя dat-файла, полученного в результате расчета колебательных мод. Должен содержать гессиан, колебательные моды и собственные вектора этих мод.
<b>NumRotTrM</b>	Число поступательных и вращательных мод, которые исключаются из расчета. Необязательный параметр. Если не указан, то устанавливается значение 6.
<b>cutoff</b>	Параметр регулирует число мод, которые будут исключены из расчета. Чем он меньше, тем большее число мод включается в расчет. При 0.0 расчет идет для всех мод, при 1.0 ни для одной. (По умолчанию 0.05)
<b>deep</b>	Отсекаются все квантовые числа при которых получатся значения меньше максимального, умноженного на deep. Если $\text{deep}=1.0$ , то квантовые числа не будут варьироваться. Если $\text{deep}=0.0$ , то при расчете будут использованы все квантовые числа, которые при расчете дают скорость отличную от 0. (По умолчанию 0.001)

#### 4. Очень краткая теория

Скорость неадиабатического переноса описывается по формуле

$$k_{nr} = \frac{4}{\Gamma_f} \sum_n |V_{i0,fn}|^2 \quad (1)$$

Возмущение рассчитывается по формуле:

$$V_{i0,f\{n_1,n_2,\dots,n_{3N-6}\}} = - \sum_v \sum_{q=x,y,z} M_v^{-1} \left\langle i \left| \frac{\partial}{\partial R_{qv}} \right| f \right\rangle \times \left[ \sum_{j=1}^{3N-6} B_{vqj} \sqrt{\frac{1}{2n_j!}} \omega_j (n_j - y_j)^2 y_j^{n_j-1} \exp(-y_j) \prod_{\substack{k \neq j \\ k=1}}^{3N-6} \sqrt{\frac{\exp(-y_k) y_k^{n_k}}{n_k!}} \right] \quad (2)$$

Где  $M_v$  – масса  $v$ -го атома,  $R_{qv}$  – координата  $v$ -го атома соответствующая одной из  $x,y,z$  осей координат,  $\left\langle i \left| \frac{\partial}{\partial R_{qv}} \right| f \right\rangle$  – элемент матрицы неадиабатических коэффициентов NACME между состояниями  $i$  и  $f$  (по факту скорее вектор, чем матрица).  $B_{vqj}$  – матрица для преобразования нормальных координат в декартовы.  $\omega_j$  – частота колебания  $j$ -ой моды. Факторы Хуана-Риса  $y_j$  находится как

$$y_j = \frac{\omega_j \Delta Q_j^2}{2} \quad (3)$$

Где  $\Delta Q_j$  – изменение равновесных нормальных координат молекулы при переходе от начальной геометрии  $|i\rangle$  к конечной  $|f\rangle$ .

Уширение  $\Gamma_f$  находится по формуле:

$$\Gamma_f = 2\delta\nu\sqrt{2\ln 2} \quad (4)$$

$$(\delta\nu)^2 = \frac{1}{2} \sum_j y_j^2 v_j^2 (2n_j + 1) \\ v_j = \omega_j / 2\pi c$$

Среднее число осцилляторов с энергией  $\hbar\omega_j$  определяется распределением Бозе-Эйнштейна

$$n_j = \frac{1}{\exp\left(\frac{\hbar\omega_j}{kT}\right) - 1}$$

## 5. Как работает программа

### а. Описание переменных

**Вектор координат  $R$**  [Боровский радиус] (*В транспортированном виде*)

$$R^T = (x_1, y_1, z_1, x_2, y_2, z_2 \dots x_N, y_N, z_N)$$

*Real(4)::coord(2,3000)* Первый столбец начальные координаты, второй-конечные.

**Вектор масс  $m$**  [а.е.м.] (*В транспортированном виде*)

$$m^T = (m_1, m_1, m_1, m_2, m_2, m_2 \dots m_N, m_N, m_N)$$

*Real(4)::mass(3000)*

**Гессиан  $\mathcal{H}$**  [Хартри/бор<sup>2</sup>]

$$\mathcal{H} = \begin{pmatrix} \frac{\partial^2 E(R)}{\partial x_1^2} & \frac{\partial^2 E(R)}{\partial x_1 \partial y_1} & \dots & \frac{\partial^2 E(R)}{\partial x_1 \partial z_N} \\ \frac{\partial^2 E(R)}{\partial y_1 \partial x_1} & \frac{\partial^2 E(R)}{\partial y_1^2} & \dots & \frac{\partial^2 E(R)}{\partial y_1 \partial z_N} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial^2 E(R)}{\partial z_N \partial x_1} & \frac{\partial^2 E(R)}{\partial z_N \partial y_1} & \dots & \frac{\partial^2 E(R)}{\partial z_N^2} \end{pmatrix}$$

*Real(4)::hess(3000,3000)*

**Частоты колебания мод  $\Omega$**  [1/а.е.в.]

$$\Omega^T = (\omega_1, \omega_2, \omega_3 \dots \omega_{3N})$$

*Real(4)::mode(3000)*

**Факторы Хуана-Риса  $Y$**  (*В транспортированном виде*)

$$Y^T = (y_1, y_2, y_3 \dots y_{3N})$$

*Real(4)::HRFs(3000)*

**Неадиабатические коэффициенты  $K$**  (*В транспортированном виде*)

$$K = (k_1, k_2, k_3 \dots k_{3N})$$

$$k_1 = \left\langle i \left| \frac{\partial}{\partial x_1} \right| f \right\rangle$$

$$k_2 = \left\langle i \left| \frac{\partial}{\partial y_1} \right| f \right\rangle$$

$$k_3 = \left\langle i \left| \frac{\partial}{\partial z_1} \right| f \right\rangle$$

$$k_4 = \left\langle i \left| \frac{\partial}{\partial x_2} \right| f \right\rangle$$

$$k_{3N} = \left\langle i \left| \frac{\partial}{\partial z_N} \right| f \right\rangle$$

*Real(4)::paste(10,10,3000)* Первый параметр обозначает номер конечного состояния, второй начального, третий номер коэффициента

## в. Описание процедур и функций

Для удобства напротив некоторых формул стоят пометки **markerX**. В коде такие же пометки **!!!markerX**. По этим меткам можно быстро найти, какой из кусков кода соответствует определенной формуле.

**subroutine eignvalvec(mass, hess, number\_atoms)**

**На вход** процедуры подаются вектор масс **m**, гессиан **H**, и число атомов **N**

Сначала находится матрица **W**

$$W = M H M$$

M – диагональная матрица (3Nх3N)

$$M = \begin{pmatrix} 1/\sqrt{m_1} & 0 & \dots & 0 \\ 0 & 1/\sqrt{m_1} & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 1/\sqrt{m_N} \end{pmatrix}$$

Потом решается уравнение,

$$W H = W H$$

которое можно свести к системе:

$$W h_j = \omega_j^2 h_j$$

где  $h_j$  собственный вектор матрицы **W**, соответствующий  $j$ -ой моде

$$h_j^T = (h_{1,j}, h_{2,j} \dots h_{3N,j})$$

$$H = \begin{pmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,3N} \\ h_{2,1} & h_{2,2} & \dots & h_{2,3N} \\ \vdots & \vdots & \dots & \vdots \\ h_{3N,1} & h_{3N,2} & \dots & h_{3N,3N} \end{pmatrix}$$

**Mark1**

В матрице **H** каждый столбец соответствует собственному вектору  $h_j$ .

$$W = \begin{pmatrix} \omega_1^2 & 0 & \dots & 0 \\ 0 & \omega_2^2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \omega_{3N}^2 \end{pmatrix}$$

**Mark2**

Из последней матрицы легко получается вектор **Ω**.

**В результате работы процедуры eignvalvec** мы получаем матрицу собственных векторов мод **H** и вектор частот колебаний мод **Ω**.

**HuangRhys(Omega,A,Mass,Ri,Rf,number\_modes)**

На вход функции поступает матрица  $A = H^{-1}$ , вектор частот колебаний  $\Omega$ , векторы координат стартовой  $R_i$  и конечной геометрии  $R_f$ .

$$C_j = \sqrt{m_j}(r_{ij} - r_{fj}) \quad \text{Mark3}$$

Смещения в нормальных координатах:

$$Q = AC$$

$$\begin{pmatrix} Q_1 \\ Q_2 \\ \dots \\ Q_{3N} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1N} \\ A_{21} & A_{22} & \dots & A_{2N} \\ \vdots & \vdots & \dots & \vdots \\ A_{N1} & A_{N2} & \dots & A_{NN} \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \\ \dots \\ C_{3N} \end{pmatrix} \quad \text{Mark4}$$

$$y_j = \frac{\omega_j Q_j^2}{2} \quad \text{Mark5}$$

В результате работы процедуры HuangRhys мы получаем вектор факторов Хуана-Риса  $Y$

**Pertrubation(K, Y, Ω, n, m, A, NMRT, nm)**

В этой функции рассчитывается результат формулы (2) при заданных значениях переменных. Оригинальная формула содержит три вложенных друг в друга суммирования. Здесь эти операции были заменены на перемножение друг на друга матриц и векторов.

На вход подаются вектор колебательных частот  $\Omega$ , вектор содержащий неадиабатические координаты  $K$ , вектор с факторами Хуана-Риса  $Y$ , вектор содержащий набор квантовых чисел  $n$ , вектор с массами атомов  $m$ , матрица

$B = A^T = (H^{-1})^T$ , число колебательных мод, которые удаляются из расчета **NMRT**

$$P\_Mu_j = \sqrt{\frac{1}{n_j!} \exp(-y_j) y_j^{n_j}} \quad \text{Mark6}$$

$$FV_j = \frac{K_j}{m_j} \quad \text{Mark7}$$

$$PS_j = \left( \prod_{\substack{i=NMRT+1 \\ j \neq i}}^{3N} P_{Mu_i} \right) \sqrt{\frac{1}{2(n_j!)} (n_j - y_j)^2 y_j^{n_j-1} \exp(-y_j)} \quad \text{Mark8}$$

$$SV = FV^T \cdot \text{diag}(\sqrt{m}) \cdot B \quad \text{Mark9}$$

Формула для расчета SV была написана так, что в ней не учитывались поступательные и вращательные моды.

$$\begin{pmatrix} SV_1 \\ SV_2 \\ \vdots \\ SV_{3N} \end{pmatrix} = \begin{pmatrix} k_1 & k_2 & \dots & k_{3N} \\ m_1 & m_1 & \dots & m_N \end{pmatrix} \begin{pmatrix} \sqrt{m_1} & 0 & \dots & 0 \\ 0 & \sqrt{m_1} & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \sqrt{m_{3N}} \end{pmatrix} \begin{pmatrix} B_{1,NMRT+1} & B_{1,2} & \dots & B_{1,3N} \\ B_{2,NMRT+1} & B_{2,2} & \dots & B_{2,3N} \\ \vdots & \vdots & \dots & \vdots \\ B_{3N,NMRT+1} & B_{3N,2} & \dots & B_{3N,3N} \end{pmatrix}$$

$$\begin{pmatrix} SV_1 \\ SV_2 \\ \vdots \\ SV_{3N} \end{pmatrix} = \begin{pmatrix} k_1 & k_2 & \dots & k_{3N} \\ m_1 & m_1 & \dots & m_N \end{pmatrix} \begin{pmatrix} B_{1,NMRT+1}\sqrt{m_1} & B_{1,2}\sqrt{m_1} & \dots & B_{1,3N}\sqrt{m_1} \\ B_{2,NMRT+1}\sqrt{m_1} & B_{2,2}\sqrt{m_1} & \dots & B_{2,3N}\sqrt{m_1} \\ \vdots & \vdots & \dots & \vdots \\ B_{3N,NMRT+1}\sqrt{m_{3N}} & B_{3N,2}\sqrt{m_{3N}} & \dots & B_{3N,3N}\sqrt{m_{3N}} \end{pmatrix}$$

Итоговое значение возмущения находится по формуле

$$V = SV^T P S$$

**Mark10**

В результате работы функции **Pertrubation** мы получаем возмущение V

Для экономии времени формулы **Mark7** и **Mark9** рассчитываются в основном теле программы, в файле NATRC.f90

### с. Описание алгоритма суммирования

Фрагмент помечен скобками **Mark11start** и **Mark11end**

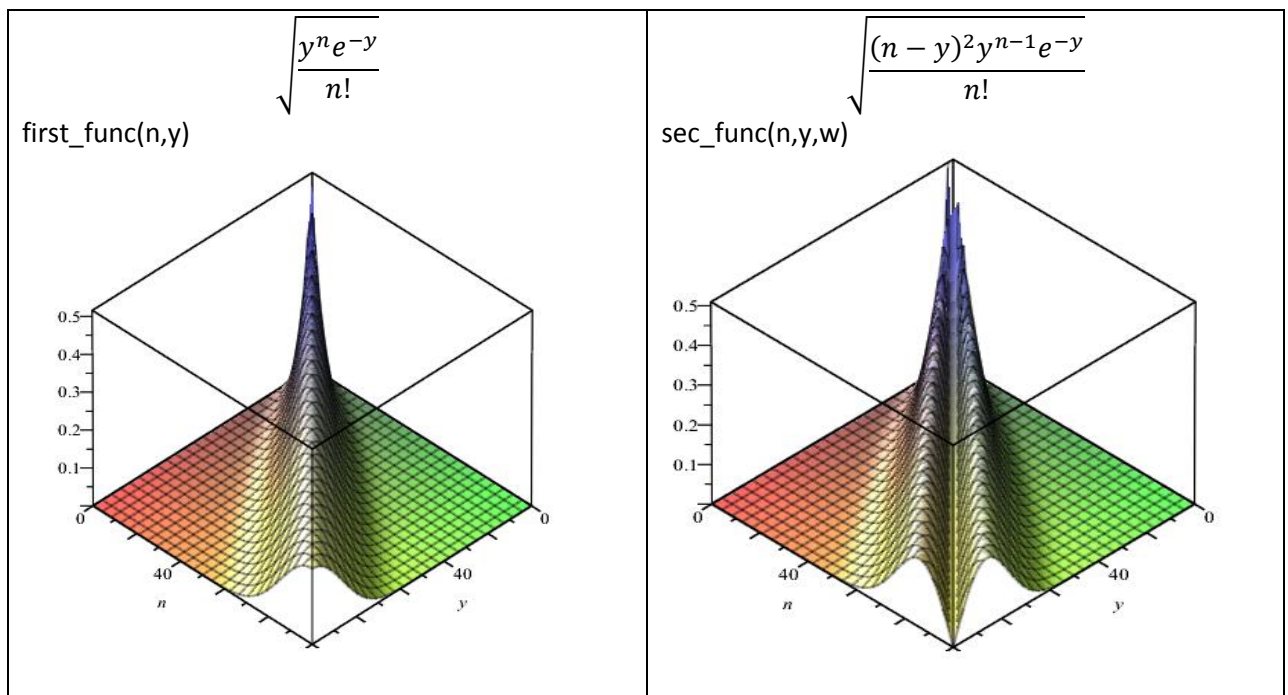
Чтобы найти скорость перехода, программа должна перебрать все наборы квантовых чисел, которые удовлетворяют закону сохранения энергии. А именно, суммарная энергия всех колебательных мод, умноженных на соответствующие им квантовые числа **E<sub>vib</sub>=Sum(n<sub>i</sub>ω<sub>i</sub>)**, должна быть равна энергии перехода **E<sub>if</sub>** между уровнями, с погрешностью 100 см<sup>-1</sup>.

Перебор происходит следующим образом. Вводятся наборы минимальных и максимальных значений квантовых чисел **n<sub>min</sub><sub>i</sub>** и **n<sub>max</sub><sub>i</sub>**. Считать за их пределами не имеет смысла, потому что результат на них пренебрежимо мал. Когда пределы выбраны, находится стартовое значение набора квантовых чисел, который определяется путем целочисленного деления энергии перехода на энергии мод. Сначала **E<sub>iv</sub>+100см<sup>-1</sup>** делится на самую большую колебательную энергию ω<sub>i</sub>. Целая часть результата деления и есть стартовое квантовое число. При этом оно не должно быть больше максимально возможного квантового числа. Потом мы берем остаток от деления и повторяем процедуру со следующей меньшей по энергии модой. Так происходит до тех пор, пока мы не найдем квантовые числа для всех мод. Найденный набор представляет стартовый набор квантовых чисел. Перебор квантовых чисел начинается с самой маленькой по энергии моды. Это квантовое число каждый раз уменьшается на единицу до тех пор, пока **E<sub>vib</sub>** не станет меньше **E<sub>if</sub>-100см<sup>-1</sup>**. После этого уменьшается на единицу не первое



квантовое число, а второе. Потом первое рассчитывается так, чтобы **Evib** было максимально близко к верхней границе. Далее процесс продолжается заново, уменьшается на единицу первое квантовое число. Также когда **Evib** выйдет за нижнюю границу, произойдет 2-го квантового числа с пересчетом 1-го. В тот момент, когда перерасчет не помогает **Evib** вернуться в границы **Eif-100см<sup>-1</sup> – Eiv+100см<sup>-1</sup>**, программа уменьшит на единицу 3-е число. Если после это удалось вернуться в границы, то процесс начинается с воздействия на 1-ое число. В противном случае, программа будет переходить к следующему квантовому числу до тех пор, пока условие не выполнится. Когда условие перестает выполняться, при уменьшении квантового числа с максимальной энергией, перебор заканчивается. В процессе перебора, для всех мод, попадающих в предел границы **Eif-100см<sup>-1</sup> – Eiv+100см<sup>-1</sup>**, рассчитывается возмущение V. Сумма всех найденных V используется для нахождения скорости.

#### d. Нахождение nmax\_i и nmin\_i



Как видно из рисунков, используемые в ПБДж функции, а следовательно и V нелинейно зависят от фактора Хуана-Риса  $y$ . Поэтому, его нельзя использовать как однозначный критерий вклада моды в потенциальную энергию. Поэтому я решил оценивать по максимальным значениям, которые может дать мода.

Допустим, у нас есть набор факторов Хуана-Риса к каждой моде:  $y_i$ . Очевидно, что каждому будет соответствовать только одно квантовое число  $n_i$  при которой  $\sqrt{\frac{y^n e^{-y}}{n!}}$  дает максимальное значение. Все в месте они дают максимальное значение A.

$$A = \prod_i \sqrt{\frac{y_i^{n_i} e^{-y_i}}{n_i!}}$$

При этом также очевидно, что для второй функции набор квантовых чисел выходит совершенно отличным от  $m$ .

Получается, что

$$\sqrt{\frac{y_j^{m_j-1}}{m_j!} e^{-y_j} (m_j - y_j)^2} \prod_{i \neq j} \sqrt{\frac{y_i^{n_i} e^{-y_i}}{n_i!}} = A \frac{\sqrt{\frac{y_j^{m_j-1}}{m_j!} e^{-y_j} (m_j - y_j)^2}}{\sqrt{\frac{y_j^{n_j} e^{-y_j}}{n_j!}}} =$$

$$Max_j = A \sqrt{\frac{y_j^{n_j-1} (n_j - y_j)^2}{y_j^{n_j}}} = A \sqrt{(m_j - y_j)^2 y_j^{m_j-n_j-1} \frac{n_j!}{m_j!}}$$

$\omega_1$	$Max_1$	$\mathbf{m}_1 n_2 n_3 n_4 \dots n_{3N}$
$\omega_2$	$Max_2$	$n_1 \mathbf{m}_2 n_3 n_4 \dots n_{3N}$
$\omega_3$	$Max_3$	$n_1 n_2 \mathbf{m}_3 n_4 \dots n_{3N}$
...	...	...
$\omega_{3N}$	$Max_{3N}$	$n_1 n_2 n_3 n_4 \dots \mathbf{m}_{3N}$

Таким образом  $Max_j$  - это максимальное значение, которым может обладать функция

$\sqrt{\frac{y_j^{k_j-1}}{k_j!} e^{-y_j} (k_j - y_j)^2} \prod_{i \neq j} \sqrt{\frac{y_i^{k_i} e^{-y_i}}{k_i!}}$ . При этом  $k = n_1 n_2 \dots \mathbf{m}_j \dots n_{3N-1} n_{3N}$  - это единственный набор квантовых чисел, при котором эта функция равна  $Max_j$ .

Логично предположить, что моды, обладающие очень маленькими максимумами по сравнению со всеми остальными модами, имеют так же и малый вклад в энергию возбуждения. После того как значащие моды найдены для них находится суммы энергий возбуждения для различных наборов квантовых чисел, расположенных в диапазоне от  $n_{min}$  до  $n_{max}$ .

$$n_{min} = n_{min1} n_{min2} n_{min3} \dots n_{min3N}$$

$$n_{max} = n_{max1} n_{max2} n_{max3} \dots n_{max3N}$$

Алгоритм расчета следующий:

Функция: `found_maxima(HRFs, nm, NMRT, cutoff, P_M, P_A, omega, nscme)`

- Берется значение  $y_i$  и округляется до целого в большую сторону, получившееся значение принимается в качестве  $n_{start1,i}$ . Потом  $y_j$  округляется в меньшую сторону, получившееся

значение выбирается как  $n_{start2,i}$ . После чего, для каждого рассчитывается значение

$\sqrt{\frac{y_i^{n_i} e^{-y_i}}{n_i!}}$ . То значение  $n_{start,i}$ , при котором получается наибольшее значение функции, **выбирается в качестве  $n_i$** .

- b) Около значение  $y_j$  берется диапазон квантовых чисел от  $m_{start,j}$  до  $m_{finish,j}$ . Эти числа индивидуально подбираются к разным  $j$ . Если  $j \leq 40$ , то  $m_{start,j} = \text{int}(y_j) - 10$ , а  $m_{start,j} = \text{int}(y_j) + 10$ . В противном случае они рассчитываются по формулам  $m_{start,j} = \text{int}(y_j) - \text{int}(i/4)$ , и  $m_{start,j} = \text{int}(y_j) + \text{int}(i/4)$ .

Для каждого  $m$ , принадлежащего этому диапазону, рассчитывается  $\sqrt{\frac{y_j^{m_j-1}}{m_j!} e^{-y_j} (m_j - y_j)^2}$ .

То, при котором получилось максимальное значение, **выбирается в качестве  $m_j$**

- c) Для каждого  $j$  рассчитывается отношение  $rel_j = \sqrt{\frac{y_j^{m_j-1}}{m_j!} e^{-y_j} (m_j - y_j)^2} / \sqrt{\frac{y_j^{n_j} e^{-y_j}}{n_j!}}$
- d) Потом рассчитывается  $cont0_j = \sum_i (\theta_i / Mass_i) B_{ij} rel_j$ , которые пересчитываются в  $cont_j = cont0_j / \sum_i cont0_i$ . Из получившихся выбираются только те  $cont_j$ , которые больше некоторого произвольно выбранного порога  $\epsilon 1$ . Моды, соответствующие этим  $cont_j$ , считаются значащими. Впоследствии возмущение рассчитывается только для них. Для всех незначащих мод квантовые числа не варьируются и считаются равными нулю.

Функция:  $found\_area(HRFs, \omega, NMRT, nm, cutoff)$

- e) Для значащих мод необходимо рассчитать  $n_{min}$  и  $n_{max}$ . Сначала методом аналогичным методы пункта находят положения максимумов функции  $\sqrt{\frac{y_j^{n_j-1}}{n_j!} e^{-y_j} (n_j - y_j)^2}$  справа и слева от  $y_j$ . Эти положения **мы обозначаем как  $n_{loc1,j}$  и  $n_{loc2,j}$**  для правого и левого максимума соответственно
- f) К  $n_{loc1,j}$  мы прибавляем +1 пока значение соответствующей ему функции не станет меньше чем значение правого максимума, умноженное  $\epsilon 2$ . То  $n$ , при котором это произошло мы **обозначаем как  $n_{max,j}$** .
- g) К  $n_{loc1,j}$  мы прибавляем 1 пока значение соответствующей ему функции не станет меньше чем значение *правого* максимума, умноженное  $\epsilon 2$ . То  $n$ , при котором это произошло, мы **обозначаем как  $n_{max,j}$** .
- h) От  $n_{loc2,j}$  мы отнимаем 1 пока значение соответствующей ему функции не станет меньше чем значение *левого* максимума, умноженное  $\epsilon 2$ . То  $n$ , при котором это произошло мы **обозначаем как  $n_{min,j}$** .
- i) В дальнейшем мы ищем энергию возмущения лишь для значащих мод, чьи наборы квантовых чисел находятся в диапазоне от  $n_{min}$  до  $n_{max}$ .